2013 IEEE/RSJ International Conference on
Intelligent Robots and Systems (IROS)
November 3-7, 2013. Tokyo, Japan

Optimal Path Planning of a Target-Following Fixed-Wing UAV Using Sequential Decision Processes*

Stanley S. Baek†, Hyukseong Kwon†, Josiah A. Yoder†, and Daniel Pack‡

***Abstract*— In this work, we consider the optimal path of a fixed-wing unmanned aerial vehicle (UAV) tracking a mobile surface target. One of the limitations of fixed-wing UAVs in tracking mobile targets is the lack of hovering capability when the target moves much slower than the minimum UAV speed, requiring the UAV maintain an orbit about the target. In this paper, we propose a method to find the optimal policy for fixed-wing UAVs to minimize the location uncertainty of a mobile target. Using a grid-based Markov Decision Process (MDP), we use an off-line policy iteration algorithm to find an optimal UAV path in a coarse discretized state space, followed by an on-line policy iteration algorithm that applies a finer grid MDP to the region of interest to find the final optimal UAV trajectory. We validate the proposed algorithm using computer simulations. Comparing the simulation results with other methods, we show that the proposed method has up to 13% decrease in error uncertainty than ones resulted using conventional methods.**

I. Introduction

Over the last few decades, engineers and researchers have made remarkable progress toward the development of robust unmanned aerial vehicles (UAVs) in both civil and military applications including intelligence collection, surveillance, reconnaissance, and environmental monitoring, to name a few. For such applications, ground or water surface target tracking is a fundamental and challenging task. Since the sensing capabilities of UAVs are constrained by the UAV dynamics and the onboard sensor capabilities, intelligent path planning for UAVs to maximize target localization accuracy is an important area of research.

Unlike rotary-wing UAVs, capable of hovering over a mobile surface target, fixed-wing UAVs must maintain a minimum speed to produce enough lift force and plan ahead when tracking targets that move slower than UAVs. The optimal path of a fixed-wing UAV following a target would be one of the paths shown in Fig. 1, depending on target speeds. If the target speed, V^t, is greater than the minimum UAV speed, $V^u_{\min}$, and less than the maximum UAV speed, $V^u_{\max}$, the simplest way to follow the target is to match the UAV speed with the target speed as shown in Fig. 1a. For

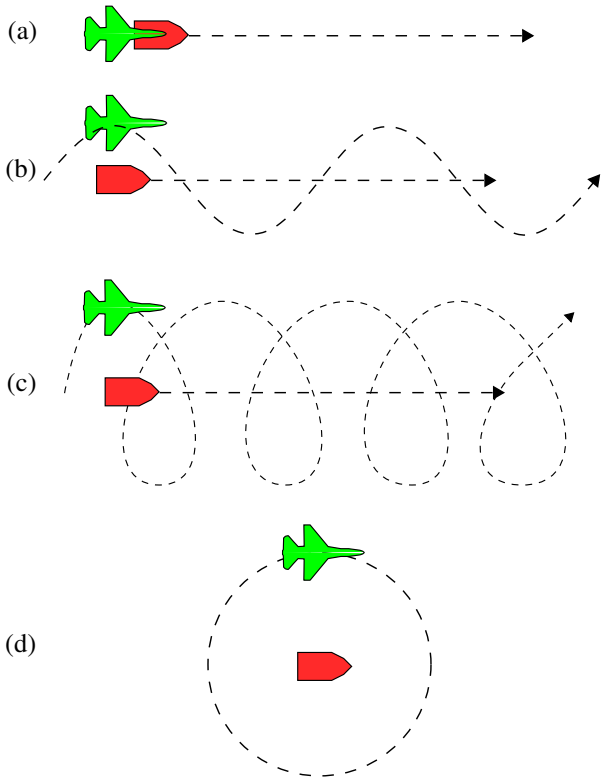


Fig. 1: The paths of a fixed-wing UAV following a target traveling at the speed of (a) $V^u_{\min} \leq V^t \leq V^u_{\max}$, (b) $V^t < V^u_{\min}$, (c) $V^t << V^u_{\min}$ (d) $V^t = 0$, where V^t is the target speed and V^u is the UAV speed.

$V^t < V^u_{\min}$, the UAV can no longer maintain a straight path to follow the target. Instead, the UAV path must be a sinusoidal path as shown in Fig. 1b. As V^t becomes much less than $V^u_{\min}$, a suitable path will be an orbital motion about the target. As an extreme case where the target is not moving, the optimal path will be a circular path about the target. In this paper, we propose a method to find in real-time the optimal path of a fixed-wing UAV following a target at various velocities, and we demonstrate the optimal path is indeed very close to one of the aforementioned paths.

A number of engineers have introduced a variety of path planning algorithms for target tracking. Beard *et al.* proposed a system architecture for assigning and intercepting targets using cooperative UAV systems [3]. In their method,

*Distribution A. Appeared for Public Release: Distribution Unlimited. The views expressed in this article are those of authors and not necessarily those of the U.S. Air Force Academy, the U.S. Air Force, the Department of Defense, or the U.S. Government.

†S. Baek, H. Kwon, and J. Yoder are with the Academy Center for UAS Research, Department of Electrical and Computer Engineering, United States Air Force Academy, USAFA, CO 80840, USA.

‡D. Pack is with the Department of Electrical and Computer Engineering, University of Texas, San Antonio, TX 78249, USA.

coordination and negotiation between UAVs were made, considering estimated time for a UAV to reach a target (arrival time at target) and available threats from scattered obstacles. Tisdale *et al*. reported practical UAV control strategies for fixed-wing aircraft using adaptable receding-horizon control [11]. In their paper, an increasing horizon planner was proposed to find appropriate optimization strategies for different situations. He *et al*. suggested a quardrotor path planning approach based on a belief network for tracking ground targets in an urban-like (road networked) environment [6]. The belief of each target's pose was modeled with a multi-modal Gaussian distribution.

Compared to estimating all feasible and continuous UAV paths to acquire the optimal UAV path, it is often more efficient to discretize a continuous state space into a state space with a fixed number of discrete states. Additionally, the nonholonomic motion of fixed-wing UAVs reduces the number of available choices for the next UAV states. Lavis *et al*. proposed a search and tracking approach for moving targets by dynamically changing the size of the search space to contain only the high-probability target locations. This improves the computational efficiency and allows targets to be tracked beyond the original search area [7]. Their approach used recursive Bayesian estimation, maintaining two different probabilistic density functions for the update and prediction stages. Another method developed by Yu *et al*. suggests a cooperative target tracking approach using dynamic occupancy grids of the estimated target locations [13]. To model occupancy grids for probabilistic path planning, they used a Bayesian filter and a second-order Markov chain.

Among discretized approaches, the Markov Decision Process (MDP) [4] is a powerful method for finding optimal solutions. Akselrod *et al*. applied a decentralized MDP approach to solve a multiple target tracking assignment problem with multiple UAVs [1]. In their modified decentralized MDP method, the transitions and observations of each agent are considered independent to obtain the polynomial algorithm. Yeow *et al*. proposed a hierarchical MDP for multiple target tracking in wireless sensor networks [12]. By using the clustered sensor network, an energy efficient and distributed multiple target tracking network can be constructed. Al-Sabban *et al*. [2] demonstrated a path planning algorithm for a UAV using MDP to minimize energy consumption by exploiting wind-energy distributed in grid cells.

In this paper, we propose an optimal UAV path planning technique while maintaining appropriate target detectability with sequential decision processes. In order to acquire the optimal path, an MDP with grid-based discretization is applied to estimate costs for each UAV path decision. The main contributions of this work are 1) a computationally efficient state space reduction technique, 2) an MDP design for the optimal path planning of a fixed-wing UAV tracking a mobile target, and 3) a real-time algorithm to compute the optimal policy.

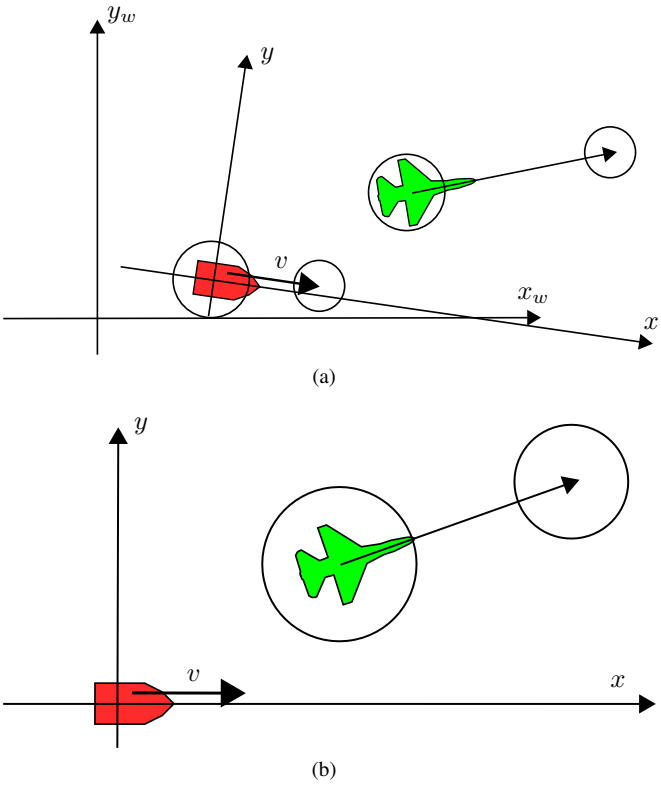


Fig. 2: A model of a UAV following a surface target in (a) a world coordinate frame and (b) a target-fixed coordinate frame

II. Dynamic Model

The path planning approach used in this paper is built on the assumption of a target traveling at a constant speed and heading. However, because we continuously refine our estimates of the target's velocity, our approach allows us to track any moving target. Although it is simple, the "almost constant velocity" method has been shown to be quite effective in many applications, and our experiments demonstrate that the approach works well at tracking targets which change their speed and heading.

The coordinate system that we use in our planning algorithms is fixed to a moving target. To build an intuitive understanding of how our approach can handle uncertainty in both target and UAV motion, in this section, we briefly derive the uncertainty of a UAV in the coordinate frame of a moving target based upon the uncertainties in a fixed reference frame.

Suppose in a world-fixed Cartesian reference frame (x_w, y_w), shown in Fig. 2a, the target is located at $\mathbf{x}_w^t \in \mathbb{R}^2$ and moving with velocity $\mathbf{v}_w^t \in \mathbb{R}^2$.[1] Suppose further that we are able to produce estimates of the UAV location and velocity, respectively, $\hat{\mathbf{x}}_w^u \sim \mathcal{N}(\mathbf{x}_w^u, P_{w,x}^u)$ and $\hat{\mathbf{v}}_w^u \sim \mathcal{N}(\mathbf{v}_w^u, P_{w,v}^u)$. Similarly, we have estimates of the targets location $\hat{\mathbf{x}}_w^t \sim \mathcal{N}(\mathbf{x}_w^t, P_{w,x}^t)$ and velocity $\hat{\mathbf{v}}_w^t \sim \mathcal{N}(\mathbf{v}_w^t, P_{w,v}^t)$.

[1] We assume the motion of the UAV is constrained in 2D space, i.e., the altitude of the UAV is constant. This assumption is valid since we plan to extend our method to multiple UAVs tracking multiple targets, and we plan to have each UAV maintain its altitude to avoid collisions.

We now define a second coordinate system (x, y), whose origin is fixed to the target shown in Fig. 2b. Since we will use this coordinate system throughout the remainder of the paper, we use no prefix for simplicity. We set the transformation from world-fixed coordinate frame to target-fixed coordinates to be

$$\begin{aligned} \hat{\mathbf{x}}^u &= \hat{\mathbf{R}}\hat{\mathbf{x}}^u_w - \hat{\mathbf{x}}^t_w, \\ \hat{\mathbf{v}}^u &= \hat{\mathbf{R}}\,\hat{\mathbf{v}}^u_w, \end{aligned}$$

where matrix $\hat{\mathbf{R}}$ rotates $\hat{\mathbf{v}}^t_w$ to lie along the x-axis of the target coordinate frame. Although the position and heading of the target are known perfectly in the target coordinate frame, $\hat{\mathbf{R}}$ and $\hat{\mathbf{x}}^t_w$ propagate the uncertainty of the target position and heading into $\hat{\mathbf{x}}^u$ and $\hat{\mathbf{v}}^u$. In general, $\hat{\mathbf{x}}^u$ and $\hat{\mathbf{v}}^u$ will not be normally distributed, but they will be unimodal. Therefore, we can approximate them by a Gaussian distribution as illustrated by the larger circles in Fig. 2.

We use a coordinated-flight model of fixed-wing UAV motion. In this model, the UAV flies with a constant velocity v and can roll up to its maximum bank angle ϕ, giving it a minimum turning radius

$$R = \frac{v^2}{g \tan \phi},$$

where g is the gravitational acceleration. Based upon these parameters, in the time period T, the UAV will change its heading by a maximum of

$$\theta_{\max} = \frac{vT}{R} = \frac{Tg \tan \phi}{v},$$

where $\theta_{\max}$ is measured in radians. We model the UAV to assume a fixed roll over each time-step, a common assumption for high-level coordinated-turn planning models.[2]

III. Grid-Based Path Planning

We use a Markov Decision Process (MDP) to find the optimal path of a UAV based on the current states of the UAV and the target. The path planning problem is to select a sequence of waypoints for the UAV that minimizes the uncertainty of target localization. With an onboard sensor, we assume that the location uncertainty is proportional to the distance from the UAV to the target.

We use a grid-based discretization for a discrete state space MDP. The main goal of discretization is to reduce the number of states for the MDP while the discretized state space adequately represents the continuous state space. The system of a UAV and a target in a 2 dimensional space has 8 degrees of freedom — the positions and velocities of both UAV and target. For a discretized system, we transform the (x_w, y_w) vector space into the (x, y) vector space as discussed in the previous section. Also, we assume the target is always located at the origin for each time step and moving along the x axis, and the UAV is located at a vertex of the grid in the discrete state space as illustrated in Fig. 3. The UAV's heading angle is defined with respect to the target velocity vector. As the granularity of the grid becomes finer, we have a better approximation of continuous state space system, but with a large number of states, it is computationally expensive. This is the main reason that an online path planning using MDP has not been considered in most UAV target tracking algorithms. In the following subsections we propose a new algorithm using MDP to overcome aforementioned exponentially growing computation issue.

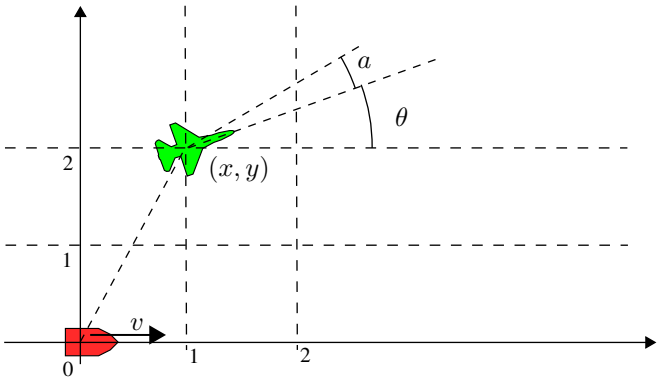


Fig. 3: Discretized MDP state space in a grid.

A. Markov Decision Process

The infinite horizon MDP used in this work is defined as $\mathcal{M} = \{\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}\}$ representing a random and sequential decision process. The parameters of MDP are defined as:

- $\mathcal{S}$ is the finite set of possible states representing the motion of the UAV with respect to the target. A state $s \in \mathcal{S}$ is denoted by $s = (x, y, \theta, v)$, where $q := (x, y) \in \mathbb{Z}^2$ is the grid position of the UAV with respect to the current position of the target, $\theta \in \{2\pi/m, 2 \cdot 2\pi/m, ..., 2\pi\}$ is one of the m-discretized heading angles of the UAV with respect to the velocity vector of the target, and v is the target speed. For each time step, we estimate the speed of a target using the previous and current locations of the target, and the target is assumed to move at the estimated velocity. Based on this assumption, we determine the best policy for each state to attain the infinite horizon optimal path.
- $\mathcal{A}$ is the finite set of actions available from each state. An action, $a \in \mathcal{A}$, is defined by the change of UAV heading from the current heading of the UAV. We assume that the UAV can move in a finite number of directions, e.g., $a \in \{-2 \cdot 2\pi/m, -2\pi/m, 0, 2\pi/m, 2 \cdot 2\pi/m\}$, where m is the number of discretized heading angles as defined earlier and $2 \cdot 2\pi/m$ is the maximum turning angle, $\theta_{\max}$, at the minimum UAV speed.
- $\mathcal{T} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ is the state transition model for the discretized state space, and it is the probability of the next state s', given action a in state s, $P(s'|s, a)$. The state transition of continuous state space in the $(\bar{x}, \bar{y})$ vector space is defined by

[2]In reality, a UAV's minimum speed is dependent upon what roll (bank) angle it uses. For the sake of simplicity, we assume that we use the minimum speed required to maintain the maximum bank angle selected for the control algorithm.

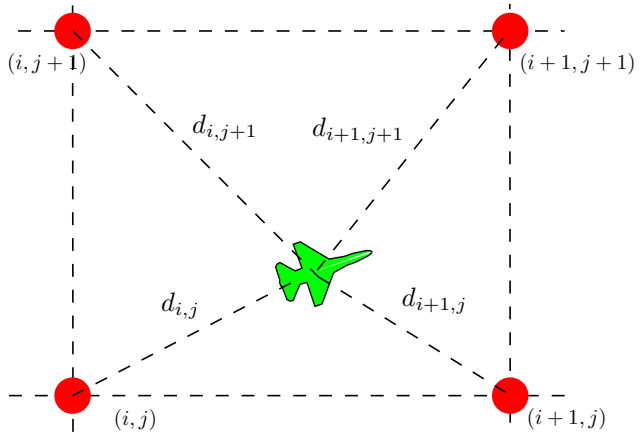


Fig. 4: State transition of a discrete state space MDP in a grid. The probability distribution of the discretized states at $k+1$ is inversely proportional to the distance from vertices and the expected location of the UAV at $k+1$.

$$\begin{aligned} \theta_{t+1} &= \theta_t + a_t + w_\theta \\ x_{t+1} &= x_t + \Delta_T u_t \cos\phi - \Delta_T v_t + w_x \\ y_{t+1} &= y_t + \Delta_T u_t \sin\phi + w_y \\ v_{t+1} &= v_t + w_v \end{aligned}$$

where u_t is the speed of the UAV at time t, v_t is the speed of the target, $a_t \in \mathcal{A}$ is the input steering angle, Δ_T is the time duration for a single time step, $\phi = \theta_t + a_t$ is the expected heading angle at $t+1$, and w's are the zero mean Gaussian random noises. For the discrete state transition, we first take the expectation of the next continuous state and find the probability distribution based on the distance between the nearest four vertices and the expected location of the UAV as shown in Fig. 4. The probability of the next grid location, $q' \in \mathbb{Z}^2$, is given by the inverse of the distance between q' and the expected location of the UAV divided by the sum of the inverse of distances between the nearest four vertices and the estimated UAV location, i.e.,

$$P(q'_{i+m,j+n}|\zeta_{i,j}) = \frac{d^{-1}_{i+m,j+n}}{\sum_{k,l\in\{0,1\}} d^{-1}_{i+k,j+l}} \tag{1}$$

for $m, n \in \{0, 1\}$, where $d_{i,j}$ is the distance[3] between the expected UAV location, $(\mathbb{E}[x_{t+1}], \mathbb{E}[y_{t+1}])$, and the vertex at (i, j) and

$$\zeta_{i,j} = (\lfloor \mathbb{E}[x_{t+1}]/g_w \rfloor, \lfloor \mathbb{E}[y_{t+1}]/g_h \rfloor) \tag{2}$$

where g_w and g_h are, respectively, the grid cell width and height and $\lfloor c \rfloor$ is the largest integer not greater than c. The probability distribution of the next heading angle θ' is defined by

$$\begin{aligned} P(\theta' = \theta + a|s, a) &= p \\ P(\theta' = \theta + a + \frac{2\pi}{m}|s, a) &= \frac{1-p}{2} \\ P(\theta' = \theta + a - \frac{2\pi}{m}|s, a) &= \frac{1-p}{2} \end{aligned}$$

where $p \in [0, 1]$ is the probability of the next heading angle without disturbances. In other words, the resulting orientation of the UAV will be the same as the intended orientation with probability of p. Since we consider the target is moving at a constant speed for each iteration of MDP, the state transition of target speed is

$$P(v_{k+1} = v_k|s, a) = 1 \tag{3}$$

- $\mathcal{R} : \mathcal{S} \to \mathbb{R}$ is the expected reward for each state transition. In general, the reward of an MDP is a function of the current state, the action applied to the current state, and the next state. For this work, however, the reward is just a function of the current state, i.e., the proximity of the UAV to the target, (x_k, y_k), defined by

$$r(s) = C_{\max} - (x_k^2 + y_k^2)^{\frac{1}{2}}, \tag{4}$$

where $C_{\max}$ is the maximum range of the gimbaled camera onboard the UAV. Equation (4) implies a negative reward for the target not in the UAV's sensor range. The maximum distance to the target, $C_{\max}$ is determined by the camera specification. Since we assume that we use a gimbaled camera, we can change the orientation from the sensor to the target so that the camera sees the target at all times. However, allowable field of view (FOV) and pixel size of the target to recognize the ground target should be an important factor to determine the value of $C_{\max}$. Let us assume the allowable FOV angle to be α_f, the resolution (width or height) of image plane as P, the expected target size on the ground to be t_{act}, and the minimum size of the target in the image plane as t_{pix}, then we can have the following relation:

$$\frac{\text{Target Pixel Size}}{\text{Image Size}} = \frac{\text{Target Size}}{\text{Ground Coverage}} \tag{5}$$

If we express the above relation approximately with variables,

$$\frac{t_{pix}}{P} = \frac{t_{act}}{C_{max} \cdot 2\tan(\alpha_f/2)} \tag{6}$$

Then C_{max} is estimated as

$$C_{max} = \frac{P}{2\tan(\alpha_f/2)} \cdot \frac{t_{act}}{t_{pix}} \tag{7}$$

B. Policy Iteration

In MDP, the *value* of a state $s \in \mathcal{S}$ under a policy $\pi(s) \in \mathcal{A}$, denoted by $V^\pi(s)$, is the expected return when starting

[3]To avoid singularity when the distance is zero, we added a small value $\epsilon \approx 0.0001$ to the distance.

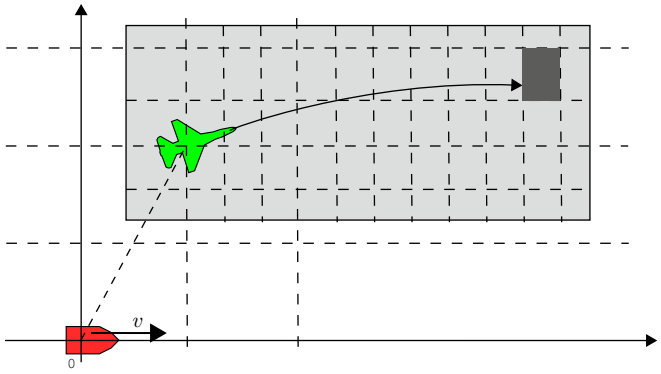


Fig. 5: Fine-grained grid for an MDP covering the area (depicted in gray) near the optimal path obtained from the lookup table generated by the policy iteration algorithm using a coarse-grained grid.

in s and following π thereafter, as defined in [10]

$$\begin{aligned} V^{\pi}(s) &= \mathbb{E}_{\pi}\left[r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + ... \middle| s_t = s\right] \\ &= \mathbb{E}_{\pi}\left[r_{t+1} + \gamma r V^{\pi}(s_{t+1}) \middle| s_t = s\right] \\ &= \sum_a \xi(s,a) \sum_{s'} T(s,a,s')[r(s) + \gamma V^{\pi}(s')] \quad (8) \end{aligned}$$

where s' is the next state followed by the current state s when action a is applied, $\xi(s,a)$ is the probability of taking action a in state s under policy π, and the discount factor $0 \leq \gamma \leq 1$ modulates the effect of future rewards on present decisions. Equation (8) is called the state-value function for policy π, and this policy can be evaluated iteratively by

$$V^{\pi}_{k+1}(s) \leftarrow \sum_{s'} T(s, \pi(s), s')[r(s) + \gamma V^{\pi}_k(s')], \quad (9)$$

where V_k is the state-value function at the k-th iteration. Each *policy evaluation* defined by (9) is started with the value function for the previous policy. Using this updated state-value function V_k, we obtain a better policy using *policy improvement* given by

$$\pi'(s) = \arg\max_a \sum_{s'} T(s,a,s')[r(s) + \gamma V^{\pi}_k(s')], \quad (10)$$

where $\pi'(s)$ is the new *greedy* policy. Starting from $V_0(s) = 0$ and arbitrarily$\pi(s)$ $\forall s \in \mathcal{S}$, we repeat (9) followed by (10) until $\pi(s)$ converges to the optimal policy π^* [5]. The pseudocode for the policy iteration algorithm can be found in [9], [10].

Using the policy iteration, we find the optimal policy for all discrete states. One of the problems in this algorithm, however, is that it is computationally very expensive. As a consequence, it is impractical to be used in real time by a small UAV. The method we developed, however, uses the algorithm offline for a number of fixed target velocities and uses the computed optimal policy as a lookup table in real-time. This lookup table method works well for a constant target speed, but sometimes it is susceptible to varying target speed with observation noise.

TABLE I: Specifications of offline and online policy iterations

	Offline	Online
Grid size	700 m × 700 m	Area of interest
Grid cell size	12 m × 12 m	3 m × 3 m
Max Sensor Range ($C_{\max}$)	300 m	300 m
# of orientation angles	64	64
# of possible input angles	5	5
Minimum turning radius	60 m	60 m
Target speeds	$\{0, 1, ..., 12\}$ m/s	$[0, V_{\max}]$
discount (γ)	0.95	0.95

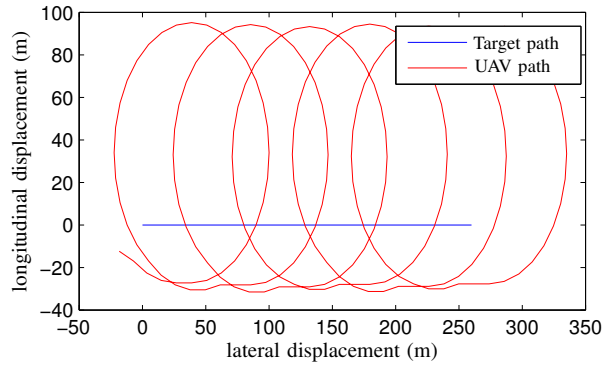
In addition to the lookup table method, we have developed an algorithm that uses the lookup table results to improve the optimal trajectory. For a given state, we create a fine-grained grid for an MDP covering the area near the optimal path obtained from the lookup table as shown in Fig. 5. The number of states required is now much less than the offline MDP because we only need to cover a small region (an example is depicted in gray in Fig. 5). For this new MDP, we set a large reward for the goal states (depicted in dark gray in Fig. 5) and process online policy iteration. Since the initial policy is already near optimal, the policy π converges to the optimal policy π^* much faster than the offline algorithm.

IV. Results and Analysis

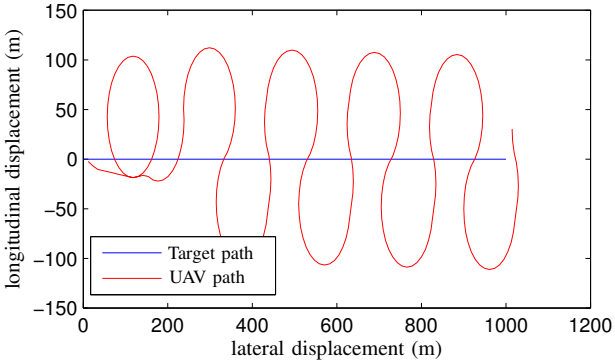
We processed the offline policy iteration for target speeds of 0, 1, ..., 12 m/s using the specifications listed on Table I. For each target speed, the process took approximately 40 hours.[4] Figure 6 shows UAV paths following targets with constant speeds. For a target moving much slower than the UAV (the speed ratio of the UAV to the target α is less than approximately 0.25), the optimal path of the UAV is a spiral motion as shown in Fig. 6a. As an extreme case, the UAV path becomes a circular orbit about the target as the target speed approaches zero. On the other hand, the optimal UAV path following a target slightly slower than the UAV ($\alpha >\sim 0.5$) is a sinusoidal path as shown in Fig. 6c. Fig 6b shows the optimal path for the UAV following the target when $\sim 0.25 < \alpha <\sim 0.5$. As a rather simple case, if a target moves at a speed greater than the minimum UAV speed but less than the maximum UAV speed, the optimal path would be linear. As we expected, these results agree with the predicted paths shown in Fig. 1.

For the online simulations as specified on Table I, we first estimated the target velocity and selected the optimal policy and path from the offline results based on the rounded target speed. Then, we generated a new MDP for the estimated target speed with the cell size of 3 m × 3 m. Although we had a finer granularity, the total number of states was orders of magnitude smaller than the offline MDP. The average time of computation was approximately 1 second for each time step.

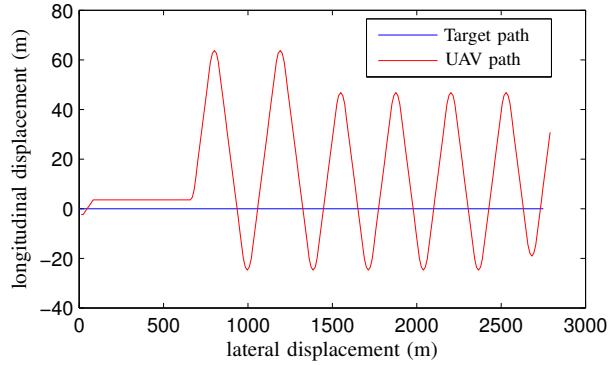
[4]We used Matlab on an Intel i7 microprocessor to run the algorithm.



(a)



(b)



(c)

Fig. 6: The optimal paths of 12 m/s UAV tracking a target moving at the speed of (a) 1.3 m/s, (b) 4.0 m/s, and (c) 11.0 m/s.

We have generated a number of target paths with their target velocities varying along the paths. The UAV with online policy iteration algorithm follows the target based on the noisy observations of the target locations with the noise variance of 5 meters in both x and y directions.

A cycloid path of a target shown in Fig. 7 has target velocities from 0 to 11.0 m/s and it changes its direction by 180° once each period. At the beginning of target tracking, the UAV tries to make a spiral path for the slow motion of the target. As the UAV finds the target speed increasing, the UAV

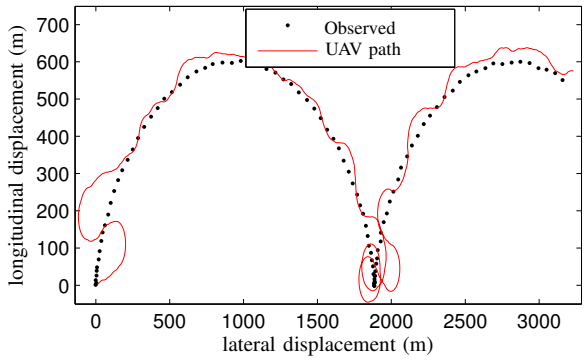


Fig. 7: UAV following a target on a cycloid path. The target velocity varies from 0 to 11 m/s. The sampling time of the path is 1 second, but the intervals between dots are 5 seconds for high visibility.

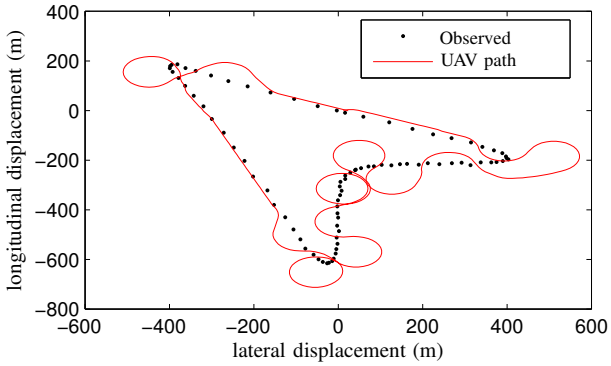


Fig. 8: UAV following a target on a cycloid path. The target velocity varies from 1.1 to 13.0 m/s.

path becomes oscillatory until the target speed reaches the minimum UAV speed. As the target slows down its speed and changes its heading direction at the end of one full cycle of the cycloid path, the UAV makes a couple of spiral motions until it makes another oscillatory path to follow the target. The average root mean square error (RMSE) of distance between the UAV and the target is 87.68 meters.

An *arrowhead* path shown in Fig. 8 is also interesting. The path has straight lines, sharp turns, and a slight turn with the velocity varying from 1.1 to 13.0 m/s. The target starts from the origin and moves toward the positive x direction at the beginning. As the target rapidly slows down to make a sharp turn, the UAV happens to fly away from the target, but it soon catches up with the target. The average RMSE between the UAV and the target is 72.15 meters.

The zigzag path shown in Fig. 9 is not a realistic path since there are 90° turns with constant velocity. The zigzag path, however, is a good example to test overshoot behaviors. The average RMSE between the UAV and the target is 64.77 meters for the zigzag path.

We compared the results with those generated using other methods. Rafi *et al.* [8] proposed a simple technique for

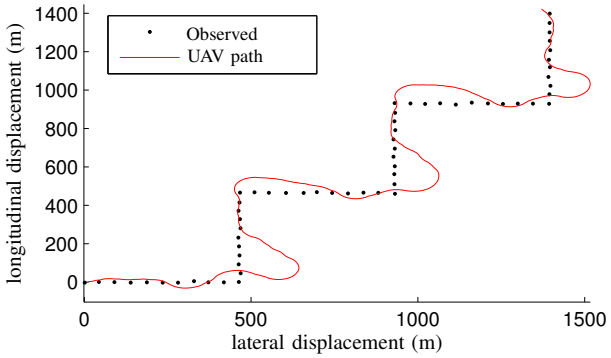


Fig. 9: UAV following a target on a zigzag path with 90° turns. The target velocity is 10.5 m/s.

TABLE II: Root Mean Square Errors (meters)

	Lookup table[a]	Online[b]	Rafi [c]	Heuristic
Cycloid	103.23	87.68	110.89	100.95
Arrowhead	111.21	72.15	76.38	77.13
Zigzag	94.75	64.77	82.44	107.69
Computational Time[d]	∼1 μs	∼1 sec	∼35 μs	∼35 μs

[a]Generated by the offline policy iteration
[b]Online policy iteration based on the policy from the offline policy iteration
[c]Rafi *et al*. [8]
[d]Using Matlab on an Intel i7 microprocessor

short-term path planning to minimize the long-term distance from the target. Rather than approaching the target directly, UAVs approach a tangent line to a minimum-radius orbit around the predicted target location. When already inside the orbit, they fly straight ahead.

We also developed a heuristic method that a UAV tries to keep the distance of the minimum orbital radius from the target. We also assumed that the UAV had five choices of yaw angle deviation at each time as discussed before. Among the UAV motion choices, the UAV selected the motion that made the distance between the estimated UAV location and the predicted target location closest to the desired (minimum) orbital radius. The predicted target location at time $t+1$ were determined by $\hat{\mathbf{x}}^t_{k+1} = \mathbf{x}^t_k + \Delta_t(\mathbf{x}^t_k - \mathbf{x}^t_{k-1})$.

Fig. 10 shows the simulation results of the Rafi and the heuristic methods. The results in Table II show that our optimal solution is superior to the two methods. For the cycloid path, our approach has performed with approximately 13% decrease in error. Although the computational time[5] of offline policy iteration lookup table method is small, that of online policy iteration method is much greater than the other methods. We are currently working toward both reducing RMSE for offline policy iteration method and decreasing the computational time for online policy iteration method.

[5]Since it is difficult to compare the computational costs with big O notations, we provide with the computational time for reader to get a sense of computational costs.

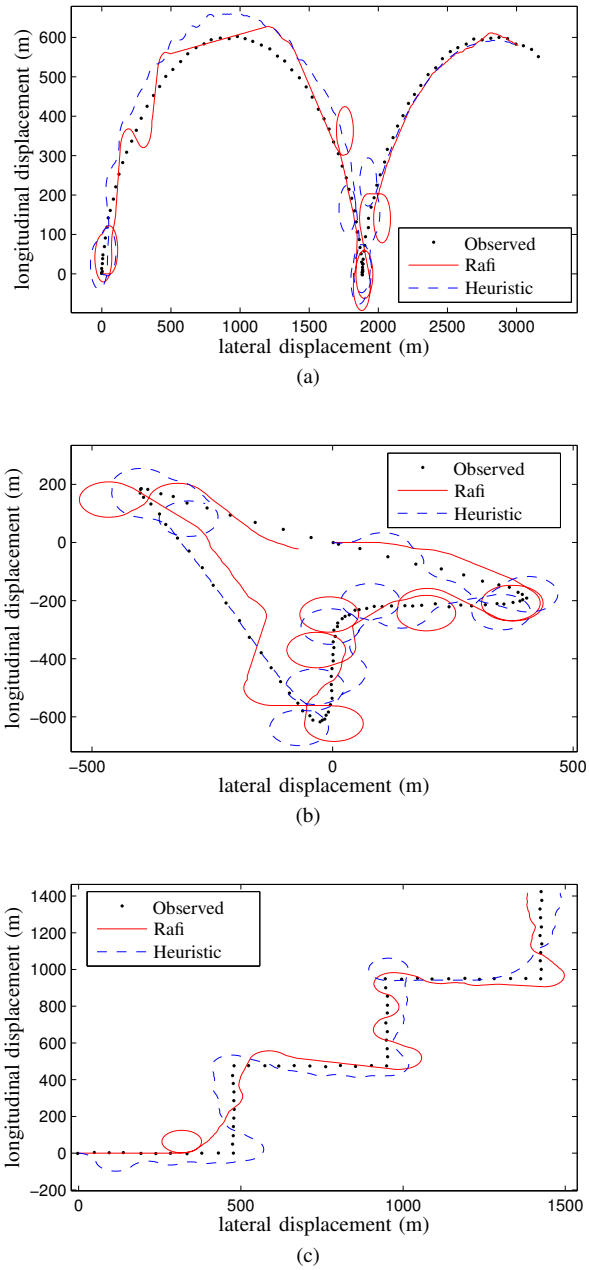


Fig. 10: The paths generated by the Rafi's method and the heuristic method for (a) a cycloid path and (b) an arrowhead path (c) a zigzag path.

V. Conclusion and Future Work

In this paper, we have shown a new optimal UAV path planning method to track a mobile target. Our method uses the theory of Markov Decision Processes to select an optimal sequence of UAV waypoints to follow the surface target while minimizing target localization uncertainties. Through policy iterations, we compute the globally optimal policy offline, and using an online policy iteration algorithm, we find the optimal UAV path in real-time to find the optimal path. The simulation results show how our method automatically

changes between various tracking regimes such as sinusoidal slaloming or spiral orbiting as the target velocity changes.

In the future, we plan to develop more accurate models for target motion and use them for intelligent UAV path planning. To this end, we are currently working on state reduction methods to decrease the number of discretized states as well as function approximation techniques to reduce the computation costs. Along with these improvements in our algorithm, we plan to verify the algorithm using hardware-in-the-loop flight simulations and to perform flight tests using UAVs currently available at our facility.

REFERENCES

[1] D. Akselrod, C. V. Goldman, A. Sinha, and T. Kirubarajan, "Collaborative sensor management for multitarget tracking using decentralized markov decision processes," in *Proceedings of SPIE 6236, Signal and Data Processing of Small Targets*, vol. 6236, May 2006, pp. 623 610–1.

[2] W. H. Al-Sabban, L. F. Gonzalez, and R. N. Smith, "Wind-energy based path planning for electric unmanned aerial vehicles using markov decision processes," in *Proceedings of the IEEE International Conference on Robotics and Automation*, May 2013, accepted.

[3] R. W. Beard, T. W. McLain, M. A. Goodrich, and E. P. Anderson, "Coordinated target assignment and intercept for unmanned air vehicles," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 6, pp. 911–922, August 2002.

[4] R. Bellman, "A Markovian decision process," *Journal of Mathematics and Mechanics*, vol. 6, no. 4, pp. 679–684, 1957.

[5] D. P. Bertsekas and J. N. Tsitsiklis, *Neuro-Dynamic Programming*, 1st ed. Belmont, MA: Athena Scientific, 1996.

[6] R. He, A. Bachrach, and N. Roy, "Efficient planning under uncertainty for a target-tracking micro-aerial vehicle," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Anchorage, Alaska, 2010.

[7] B. Lavis, T. Furukawa, and H. F. Durrant-Whyte, "Dynamic space reconfiguration for bayesian search and tracking with moving targets," *Autonomous Robots*, vol. 24, no. 4, pp. 387–399, 2008.

[8] F. Rafi, S. Khan, K. Shafiq, and M. Shah, "Autonomous target following by unmanned aerial vehicles," in *Defense and Security Symposium, Proc. SPIE*, May 2006, pp. 623 010–623 010–8.

[9] S. Russel and P. Norvig, *Artificial Intelligence: A Modern Approach*, 3rd ed. Upper Saddle River, New Jersey: Prentice Hall, 2010.

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

[11] J. Tisdale, Z. Kim, and J. Hedrick, "Autonomous UAV path planning and estimation," *IEEE Robotics and Automation Magazine*, vol. 16, no. 2, pp. 35–42, 2009.

[12] W. L. Yeow, C. K. Tham, and W. C. Wong, "Energy efficient multiple target tracking in wireless sensor networks," *IEEE Transactions on Vehicular Technology*, vol. 56, no. 2, pp. 918–928, 2007.

[13] H. Yu, R. W. Beard, K. Meier, and M. Argyle, "Probabilistic path planning for cooperative target tracking using aerial and ground vehicles," in *American Control Conference*, 2011, pp. 4673–4678.