Intelligent Motion Video Guidance for Unmanned Air System Ground Target Surveillance

John Valasek∗, Kenton Kirkpatrick†, and James May‡

Advances in unmanned flight have led to the development of Unmanned Air Systems that are capable of carrying state-of-the-art video capturing systems for the intended purpose of surveillance and tracking. These vehicles have the capability to fly through a target area with a mounted camera and allow humans to operate both the UAS and the camera to attempt to survey any objects that are deemed targets. These systems have worked well when controlled by humans, but having them operate autonomously to reduce operator workload and manpower is even more challenging when the camera is fixed to the airframe instead of being mounted on a gimbal, so that the aircraft must be steered in order to steer the camera. The presence of winds must also be accounted for. This paper develops an algorithm for surveillance of ground targets by UAS with fixed pan and tilt cameras, in the presence of winds. This paper develops an algorithm for surveillance of ground targets by UAS. The specific RL algorithm used is Q-learning, and the objective of the approach is to bring any target located in an image captured by a camera into the center of the image using the learned control policy. The learning agent determines offline (initially) how to control the UAS and camera to get a target from any point in the image to the center and hold it there. A feature of this approach is that the learning agent will continue to learn and refine and update the previously offline learned control policy, during actual operation. Results presented in the paper demonstrate that the approach has merit for autonomous surveillance of ground targets.

I. Introduction

One way to introduce the concept of autonomy to the Unmanned Air System (UAS) motion video tracking problem is to determine a control policy that is capable of controlling the UAS autonomously along a certain trajectory, while having the camera controlled by a human. Another way is to do the opposite, and have the UAS flown manually while the camera gimbals to capture and track identified targets. Both of these methods have been explored before and have merit, but having both the UAS and the camera operated autonomously could provide greater flight and tracking efficiency. Having a system that is capable of controlling a UAS and camera system to keep a selected target visible in the camera screen would free the human supervisor to focus on selecting viable targets and analyzing the images received.

The biggest challenge stems from the need to determine an optimal control policy for keeping the target in the middle of the image, using a fixed pan and tilt camera in the presence of winds. Conventional control techniques require determining an appropriate cost function and then finding the weights that make the control optimal. Although finding the optimal control is often straightforward, determining the cost function that best describes the problem is not straightforward. For this research, Reinforcement Learning (RL) is utilized for the determination of the optimal control policy that will both gimbal the camera and steer the UAS to provide target tracking.

Reinforcement Learning (RL) is a subset of machine learning techniques that have been implemented in similar control policy learning scenarios with success. It does not require the declaration of a cost function

∗Professor and Director, Vehicle Systems & Control Laboratory, Aerospace Engineering Department. Associate Fellow AIAA, valasek@tamu.edu
†Graduate Research Assistant, Vehicle Systems & Control Laboratory, Aerospace Engineering Department. Student Member AIAA, kentonkirk@gmail.com
‡Graduate Research Assistant, Vehicle Systems & Control Laboratory, Aerospace Engineering Department. Student Member AIAA, jim.may.jr@gmail.com
because it learns the optimal control policy based on physical interaction with the environment rather than mathematical approximations of the problem. RL algorithms break the problem down into a set of states and actions for attempting to reach a goal, and receive rewards from the environment for meeting those goals. Given a particular state of the system, feasible actions are chosen, and the governing control policy is updated based on what kind of reward was given for the results of the action. Over time, the learning agent is able to converge to the optimal control policy for achieving the desired goal from each state.

The specific RL algorithm to be used is Q-learning\(^5\) modified with an Adaptive Action Grid (AAG). The AAG method was developed by Lampton and Valasek\(^2,3\) as a means to provide greater accuracy in reaching the goal state (i.e., the target), while also decreasing the size (dimensions) of the state-space to be considered. This dramatically decreases the total number of states in the system, so that the learning time becomes more feasible and the storage requirements more tractable. Consider a typical optical or infrared (IR) image which contains background features plus a target located in a Region of Interest (ROI). It is assumed here that a human operator would identify the ROI and the target in the image, although the algorithm could conceivably identify the ROI and target by itself, based upon ROI and target characteristics supplied by the human operator. The AAG software agent then discretizes the entire state-space (i.e. image) with coarse grid spacing, followed by a finer discretization of just the ROI, which is the area of the image in the immediate vicinity of the target. By not discretizing the entire state-space with a fine resolution, there are fewer state-action pairs to both learn and store in memory. However, more precision is needed to reach the goal (target), so multiple levels of finer discretization are used in the ROI as the learning agent gets closer to the goal.

This paper develops an algorithm for surveillance of ground targets by UAS. The specific RL algorithm used is Q-learning, and the objective of the approach is to bring any target located in an image captured by a camera into the center of the image using the learned control policy described above. The learning agent will determine offline (initially) how to control the UAS and camera to get a target from any point in the image to the center and hold it there. A feature of this approach is that the learning agent will continue to learn and refine and update the previously offline learned control policy during actual operation.

II. Algorithm Development

Reinforcement Learning is a process of learning through interaction in which a program uses previous knowledge of the results of its actions in each situation to make an informed decision when it later returns to the same situation. It is a method that has been used for many diverse problems ranging from board games to behavior-based robotics. The purpose of the learning agent used in RL is to maximize the long-term cumulative reward, not just the immediate reward.\(^5\) In this work the goal is to remain in the image, with a preference for being far from the edges. The reward structure must be set up to effect this desire. The agent uses the knowledge gained by reward maximization to update a control policy that is a function of the states and actions. This control policy is essentially a large matrix that is composed of every possible state for the rows, and every possible action for the columns. The three most commonly used classes of RL algorithms are Dynamic Programming, Monte Carlo, and Temporal Difference.\(^5\) The majority of Dynamic Programming methods require an environmental model, making the use of them impractical in problems with complex models. Monte Carlo only allows learning to occur at the end of each episode, causing problems that have long episodes to have a slow learning rate. Temporal Difference methods have the advantage of being able to learn at every time step without requiring the input of an environmental model. The most commonly used method of Temporal Difference is known as Q-learning, with the most common variation being Watkins Q-learning.\(^6\) Q-learning is an on-policy form of Temporal Difference that utilizes an action-value function update rule based on the equation:\(^5\)

\[
Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_{t+1} + \gamma \max_a Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)]
\]

where \(s_t\) is the current state, \(a_t\) is the current action, \(s_{t+1}\) is the next state, \(a_{t+1}\) is the next action, \(Q\) is the action-value function (used in the control policy), and the \(k\) subscript signifies the current policy. The parameter \(\alpha\) is a parameter that is used to “penalize” the RL algorithm when it repeats itself within each episode. The parameter \(\gamma\) is the future policy discount factor. \(\alpha\) and \(\gamma\) are design parameters that are kept constant for this work. To utilize RL for this problem the proper representation of the environment as the parameters of the Watkins Q-learning algorithm must be made. States, goals, rewards, and actions must be designed. The states of the RL agent, \(s\), are defined for this problem to be those states of the system.
that either give information regarding the target’s position or those of the UAS that can be controlled for tracking. This yields a state-space consisting of 3 variables: target x-position in the image, target y-position in the image, and UAS bank angle.

\[ s = [X \ Y \ \phi]^T \] (2)

The goal for an RL agent is defined using the reward structure. The overall goal is to have the target remain in the image frame once commanded, so a proper set of reward requirements must be constructed. Since leaving the image frame is considered the worst result, it retains the worst reward. In this case, a value of \( r = -20 \) is given for a target leaving the image frame. It is also desired to remain away from the edges of the image, so a reward of \( r = -5 \) is given for hitting the image boundary, while a positive reward of \( r = +20 \) is awarded for reaching the center of the image. This encourages the RL agent to move the UAS such that the target stays as far from any edge of the image frame as is possible. All other possible states yield a neutral reward of \( r = 0 \). Since the goal is generally defined for the RL agent as the state that yields the highest reward, the goal is defined as all states where the target image position is at the center.

\[ g = [0 \ 0 \ \phi]^T \] (3)

The agent for this problem is limited in its control of the states because the target global position is independent of any action the UAS takes. The only part of the environment that the UAS can control is itself. Based on the assumptions for this problem, the only way the UAS can control the position of the target in the image frame is to change its bank angle. Therefore, the action-space for this problem is defined to be increments in the UAS bank angle, where for this problem \( \Delta \phi = 2 \) degrees.

\[ a = [-2 \ 0 \ 2]^T \] (4)

This formulation of the RL problem is therefore

\[ s = [X_i \ Y_i \ \phi]^T \] (5)

\[ a = [-\Delta \phi \ 0 \Delta \phi \ +\Delta \phi]^T \] (6)

\[ g = [X_{ic} \ Y_{ic} \ \phi]^T \] (7)

### III. Simulation Results

#### III.A. Disturbance Free

All learning takes place offline, and after a sufficient number of learning episodes have been completed the control policy performance and robustness is evaluated via Monte Carlo simulation. Test cases consisting of a fixed target tracking scenario and a moving target tracking scenario are presented. For each case, target position in the image frame and time histories of the UAS states are displayed. Monte Carlo simulations are presented for a chosen timespan of 100 seconds. This represents the typical amount of time for the controller to position the aircraft in a stable tracking configuration. The RL agent was allowed to run for 1,000,000 episodes, with Monte Carlo snapshots taken at a few places beginning at 500,000 episodes. The Monte Carlo randomization places the initial position of the target in one of the four quadrants of the image frame, and at a random position within each quadrant. The controller must then steer the UAS so that ideally, the target is driven to the center of the image frame. One representative case is provided for each of the four quadrants. Images positions are given in pixels and aircraft bank angles are given in degrees. Aircraft inertial positions are in meters.

The results are taken from one single test case of the Monte Carlo runs, in which the target initial conditions place it in the image frame in quadrant 1. Figure 1 shows a 3-dimensional view of the aircraft moving in inertial airspace tracking the target. As can be seen in Figure 1, the aircraft approaches a circular orbit to track the stationary target on the ground. In Figure 2, the position of the target at each timestep
is displayed. The target begins in quadrant 1 of the image and moves toward the goal of the center of the image. However, it is unable to remain there given the aircraft’s current state and it is lost from the center. As the aircraft banks left, the target moves up in the image frame, while a right bank moves the target down. The forward motion of the aircraft causes difficulty in tracking the target in the x-direction, and this is reflected in the target almost being lost off the left side of the image. The reinforcement that the agent has received leads it to settle in a state of keeping the target in quadrant 2. Time histories of the target position in the image frame, aircraft bank angle, and commanded change in aircraft bank angle are shown in Figure 3. It is seen that the controller keeps the target in the image frame throughout the simulation timespan.

Figure 1. Simulation 3-D View: Stationary Target

Figure 2. Image Time History: Stationary Target
With a successful orbit of a moving target learned, the RL agent was then presented the task of learning to follow a target that moves. This target moves in a straight line at 60 mph, the same speed as the cruise of the aircraft. Under this condition, the aircraft attempts to follow alongside the target as it travels forward. In Figure 4, it can be seen that the aircraft begins following alongside the target well, and begins to deviate away from it as time moves forward. This is due to the stationary camera requiring tracking to be handled by banking the aircraft. It can be seen in Figure 5 that the target is maintained in the image frame throughout the duration of this simulation. Like the stationary case, the target passes the prescribed goal of centering in the image and settles in quadrant 2. The time histories shown in Figure 6 reveal that to maintain this tracking requires frequent bank angle commands, unlike the stationary case.
III.B. Wind

Accounting for wind in this problem has been done using a variety of methods,\textsuperscript{7,8} and accounting for wind disturbances using the present method requires altering the learning process in the state-space to handle the additional state information. Wind can be handled by the learning agent, but it requires knowledge of the wind speed and direction. This modification can be done by adding two new states to the learning state-space. This modified state-space is shown in Equation 8, where $v_w$ and $\psi_w$ are the wind speed and wind heading angle, respectively.
\[ \mathbf{s} = \begin{bmatrix} X & Y & \phi & v_w & \psi_w \end{bmatrix}^T \]  

(8)

With wind added to the simulation, new learning was experienced with random wind speeds and directions initialized at the beginning of each episode. After 1,000,000 learning episodes, Monte Carlo results were used with the learned Q-matrix. The following figures show an example from these Monte Carlo results for a stationary target with wind disturbances. The wind vector for this particular simulation is 13 mph at a heading angle of 45 degrees. As can be seen in Figure 7, the aircraft approaches the circular orbit as from before, but due to the wind disturbance it is not nearly as smooth of a circular orbit as in Figure 1. Figure 8 shows that throughout the duration of the simulation, the target does remain in the image. By comparing Figure 9 to Figure 3, it can be seen that many more bank angle commands are required to maintain tracking when there is wind, as is expected.

**Figure 7. Simulation 3-D View: Stationary Target with Wind**

**IV. Conclusions and Future Work**

Based on the results presented in this paper, it is concluded that:

1. Algorithm learning convergence for the static target case with fixed camera is rapid, due to the low number of state-action pairs. The number of learning episodes required to converge to a “good” solution varies with the particular case/scenario being learned, but all results have shown a clear point of diminishing returns about which running additional learning scenarios provides only a marginal improvement in performance.

2. For all of the stationary target cases evaluated, the RL controller keeps the target in the image frame throughout the simulation timespan. Although the target does not stay in the reinforcement learning positive goal area (the origin), the broader tracking goal of keeping the target in the image frame itself for a useful period of time is generally met.

3. Camera installation and orientation, and the initial position of the target relative to the initial position of the aircraft have a strong influence on the results. In each example, the controller attempts to drive the target into the second quadrant. This is due to the geometry of the scenario and the location of the camera in the aircraft. Having the target in the second quadrant allows the aircraft to turn ahead of
the target, in order to keep it in the image frame in the future. Consequently, if the camera is oriented to point out the right side of the aircraft, the aircraft is steered to keep the target in the first quadrant.

4. Preliminary results for the case of a moving target show that this method has promise for learning to track a moving target, and merits further investigation.

5. Preliminary results for wind results shows that using wind measurements in the state-space is a promising method for accounting for wind in the learning process, and merits further investigation alongside the moving target scenarios.
This research will be expanded in future work in several ways. One expansion that is to be explored is the use of a gimbaled camera rather than a fixed-base camera mounted on the UAS. This will allow for greater ease of tracking, but will require a reimagining of the Reinforcement Learning problem. Another extension that will be explored is the introduction of more action choices by varying the UAS altitude and/or cruise speed. This will allow for learning to follow a moving target that is traveling at various speeds and along a variety of trajectories, but will greatly increase the learning state-action pairs. For each expansion explored, the inclusion of wind considerations as done in this paper will allow for accurate appraisal of the UAS ability to track ground targets in actuality. With learned control policies for each research scenario, the final research to be conducted in this line will be to evaluate each policy through flight testing on an actual UAS.

Acknowledgment

This research is funded by the Raytheon Company, Intelligence and Information Systems. The Technical Monitor is Mr. Michael R. Moan. This support is gratefully acknowledged by the authors.

References