Abstract—The optimal control of a “blind” UA V searching for a target moving on a road network and heading at a known speed toward a set of goal vertices is considered. To aid the UA V , some roads in the network have been instrumented with Unattended Ground Sensors (UGSs) that detect the target’s passage. When the UA V arrives at an instrumented node, the UGS therein informs the UA V if and when the target visited the node. In addition, the UA V can choose to wait/loiter for an arbitrary time at any UGS location/node. At time 0, the target passes by an entry node on his way towards one of the exit nodes. The UA V also arrives at this entry node after some delay and is thus informed about the presence of the target/target in the network, whereupon the chase is on - the UA V is tasked with capturing the target. Because the UA V is blind, capture entails the UA V and target being collocated at an UGS location. If this happens, the UGS is triggered and this information is instantaneously relayed to the UA V, thereby enabling capture. On the other hand, if the target reaches one of the exit nodes without being captured, he is deemed to have escaped. For a given initial delay, we compute the pursuit policy, if it exists, that achieves capture in minimum time, under worst-case target actions.

I. INTRODUCTION

We are concerned with capturing a ground target moving on a road network. The operational scenario is as follows. The access road network to a restricted (protected) zone is instrumented with Unattended Ground Sensors (UGSs), placed at critical locations. As the target, referred to as the “target”, passes by an UGS, the UGS is triggered. A triggered UGS turns, say, from green to red and records the target’s time of passage. The UGSs are placed on certain edges of the graph. We assume that the speed of the target, the layout of the road network and the placement of the UGSs is known to the UA V. When the UA V arrives at an UGS location, the information stored by the UGS is uploaded to the UA V, namely, the green/red status of the UGS and, if the UGS is red, the time elapsed (delay) since the target’s passage. The target can be captured in one of two ways: either the target and UA V synchronously arrive at an UGS location, or the UA V is already loitering/waiting at an UGS location when the target arrives there. In both cases, the UGS is triggered, instantaneously informs the UA V, and the target is captured. The decision problem for the UA V is to select which UGS to visit next, including possibly staying at the current UGS location (and if so, for how long?) awaiting the arrival of the target. The decisions are made by the UA V at discrete time instants, immediately after arriving at and interrogating an UGS. Without loss of generality, we assume the target is traveling on the road network at unit speed. The UA V, on the other hand, can fly between any two UGS locations at the constant speed, V. In addition, the UA V can also wait/loiter at an UGS location for an arbitrary amount of time.

Fig. 1: Road Network, UGSs Graph, and Four Possible target Paths

In Fig. 1 an illustrative road network is shown. The roads are shown in red (arrows indicate direction of travel) and the numbered UGSs are shown as blue circles. Let there be m UGSs on the network, indexed by \( j = 1, \ldots, m \). Since information is only available (and capture only possible) at the UGS locations, we focus on the embedded graph, \( G(U, E) \), that has the UGS locations as vertices, i.e., \( U = \{1, \ldots, m\} \). We make the critical assumption that \( G \) is a directed acyclic graph. In a related work, the authors have proposed an alternate methodology for graphs with cycles [1]. To visualize the setup, see Fig. 1, where the corresponding graph, \( G \), is shown in the top right. Here, node 1 is the entry node into the network. A directed edge, \( e_{i,j} \in E \) between two nodes on the graph has a weight that equals the distance along the road network from node \( i \) to node \( j \). For each \( j \in U \), let \( C(j) \subset U \) indicate the set of child nodes that the target can get to from \( j \). Let \( G = \{ j : j \in U \text{ and } C(j) = \emptyset \} \) indicate the set of exit/goal nodes that the target is heading towards. In Fig. 1, nodes 5, 6 and 7 are the exit nodes. Furthermore, for each \( c \in C(j) \), let...
the distance along the road network between the parent and child node be indicated by $T(j,c)$. Since the target travels at unit speed, this is also the time taken by the target to go from node $j$ to its child node $c$. The UAV’s travel time from node $i$ to node $j$ is given by a scaled distance metric, $d_{V}(i,j)$. Here, it represents the Euclidean distance between the nodes divided by the UAV’s speed $V$ and it satisfies:

$$d_{V}(i,j) \leq d_{V}(i,s) + d_{V}(s,j),$$ (1)

for any $i,j,s \in \mathcal{U}$ and $d_{V}(j,j) = 0, \forall j \in \mathcal{U}$. We assume that the UAV’s travel time between any UGS and its child node is strictly less than the target’s travel time between the two nodes, i.e.,

$$d_{V}(j,c) < T(j,c), \forall c \in C(j), \forall j \in \mathcal{U}.$$ (2)

Without loss of generality, we assume that the target (upon entering the network) first visits node 1 at time 0 and $1 \notin \mathcal{G}$. Let there be $n \geq 1$ possible target paths denoted by $P_{1}, \ldots, P_{n}$ emanating from node 1 and terminating at an exit node. For the example problem, see the enumeration of the 4 possible target paths shown on the bottom right of Fig. 1. We represent a target path $P_{k}$, $1 \leq k \leq n$, by the following notation: $P_{k} = (1 \rightarrow s_{k}^{1} \rightarrow \ldots \rightarrow s_{k}^{l_{k}})$, where $s_{k}^{i}$ is the $i^{th}$ UGS along path $k$ and $s_{k}^{l_{k}} \in \mathcal{G}$. Here, $l_{k}$ is the number of UGSs along path $k$. For example, in Fig. 1 $P_{1} = (1 \rightarrow 3 \rightarrow 5)$, so $s_{1}^{1} = 3, s_{1}^{2} = 5$ and $l_{1} = 3$.

A. Properties of the target’s Path

Let $T_{k}(j)$ be the time of arrival of the target to the $j^{th}$ UGS along path $k$.

$$T_{k}(j) = T(1,s_{k}^{j}) + \sum_{r=2}^{j-1} T(s_{k}^{r},s_{k}^{r+1}), \quad j = 2, \ldots, l_{k}. \quad (3)$$

So, the length of each path is given by $|P_{k}| = T_{k}(l_{k})$. If the target were to pick the shortest path to an exit node, then he would choose, $k = \arg\min_{k=1}^{n} |P_{k}|$. Since $G$ is a directed acyclic graph, the target cannot visit any particular UGS more than once. However, it is possible that the target can reach an UGS, $U \in \mathcal{U}$ via different paths. So, for UGS/node $U_{j}$ in the graph, $j = 1, \ldots, m$, we associate the set, $\mathcal{L}_{j} = \{\mathcal{L}_{j}(1), \ldots, \mathcal{L}_{j}(n)\}$, where $\mathcal{L}_{j}(k)$ is the time at which the target would visit node $j$ while traveling along path $k$. Here, time is measured relative to time 0, when the target visits node 1. If node $j$ does not appear in some path $k \in \{1, \ldots, n\}$, then we set the corresponding time, $\mathcal{L}_{j}(k) = \infty$. We assume without loss of generality, that $\forall j, \exists k$ such that $\mathcal{L}_{j}(k) < \infty$. This condition implies that every UGS appears, at least, in one of the paths. Clearly, if this were not the case, such an UGS can be removed from consideration. By definition, we have $\mathcal{L}_{1} = \{0, \ldots, 0\}$, since node 1 is visited by the target at time 0 and along every possible target path emanating from UGS 1. We also define the set, $\mathcal{P}_{j}$, $j = 1, \ldots, m$, to be the set of paths that contain node $j$. By definition, $\mathcal{P}_{j} = \{k : \mathcal{L}_{j}(k) < \infty, k = 1, \ldots, n\}$ and $\mathcal{P}_{j} \neq \emptyset, \forall j$. We define the initial uncertainty in target path information available to the purser to be $I_{0}$. Since the target could have taken any one of $n$ paths, $I_{0} = \{1, \ldots, n\}$. Note that this definition of target position uncertainty appears to be unusual in that for small initial delays, the UAV will know where the target is on the road that contains node 1 (e.g., see left plot in Fig. 1) and so, there is no uncertainty in his position/state; but we still say his path is uncertain in that $I_{0} = \{1,2,3,4\}$ - see bottom right plot in Fig. 1. This is an important point: because the tacitly assumed information pattern is s.t. the target has no situational awareness, one could argue that the target might as well decide on his “strategy”, namely, what path will he take, at $t = 0$ - in other words, the target operates in open-loop. So we stipulate that at each point in time, and based on the evidence collected so far, the information of the UAV is the currently feasible set of possible paths, one of which the target, having made his choice at time 0, is currently traveling on. This definition of path uncertainty, meaning, the uncertainty about which of the $n$ paths the target is actually traveling on, results in a significant simplification of the underlying coupled estimation and control problem. Hereafter, we shall use the words uncertainty and information interchangeably with reference to the set of complete paths that the target is possibly traveling on.

B. Evolution of System State

Even though the UAV and target motion evolve in continuous time, decisions are made (by the UAV only) at discrete time steps. The UAV makes these decisions immediately after reaching an UGS location at time $t$ and obtaining the measurement $y$ therein: $y = -1$ for “green”, or $y = d$ for “red” + delay $d$. Let the UAV position at decision time $t$ be specified by the UGS index, $p(t) \in \{1, \ldots, m\}$. The decision variable, $u$ indicates the UGS location $u \in \{1, \ldots, m\}$ that the UAV should visit next.

The control action $u$ is dependent on the current time, UAV position and most recent information state: $u = F(t, I(t), p(t))$, where the mapping $F$ is to be determined by an optimality principle - see (10) in the sequel. So, the UAV’s position and UAV decision time evolve according to:

$$t^{+} = \begin{cases} 
  t + d_{V}(p,u), & u \neq p \\
  \min_{k \in \mathcal{I}} L_{p}(k), & u = p
\end{cases}$$ (4)

$$p^{+}(t^{+}) = u.$$ So, if the UAV decides to stay put at the current location, the next decision epoch is the earliest possible time at which new information becomes available at the current UGS $p(t)$. The crucial point here is that although “to wait or not” is a decision to be made by the UAV, the waiting time itself is purely determined by the target arrival times and UAV observations. This comes about because of the assumptions: 1) constant target speed and 2) acyclic graph. We denote by $y(t)$ the measurement the UAV made at node $p(t)$. The observation could either be a red UGS $p(t)$ with delay $d \geq 0$ i.e., $y(t) = d$, or a green UGS $p(t)$; whereupon the observation is denoted by $y(t) = -1$. Note that the UAV may choose $u = p(t)$ only if the observation $y(t) = -1$. If the UAV observes a red UGS, it confirms that the target did
pass through UGS \( p(t) \) and there is no value in the UAV staying at \( p(t) \) any longer. Indeed, it would be detrimental to the search effort (in terms of time to capture).

Suppose the target path uncertainty information available to the UAV at \( p(t) \) is \( \mathcal{I}(t) \). We calculate the information/path uncertainty set at time \( t^+ \) for the two possible observations at \( u \) as follows:

Red \((y^+ = d \geq 0)\): The UAV will observe a red UGS with delay \( d \geq 0 \) where \( d \in \{s|s = \tau^+ - L_u(k), s \geq 0, k \in \mathcal{P}_u \cap \mathcal{I}(t)\} \). This implies that the target was at the location of UGS \( u \) at time \( t^+ - d \). Therefore, the information at time \( t^+ \) will be:

\[
\mathcal{I}(t^+|u, d) = \{k : k \in \mathcal{I}(t), L_u(k) = t^+ - d\}. \tag{5}
\]

So we only retain those paths from \( \mathcal{I}(t) \) that are consistent with the target passing through \( u \) at time \( t^+ - d \).

Green \((y^+ = -1)\): The UAV will observe a green UGS at time \( t^+ \). This implies that the target has not visited \( u \) thus far. Therefore, the information update is given by:

\[
\mathcal{I}(t^+|u, -1) = \{k : k \in \mathcal{I}(t), L_u(k) > t^+\}. \tag{6}
\]

So we only retain those paths from \( \mathcal{I}(t) \) that are consistent with the target passing through \( u \) at a time greater than \( t^+ \).

II. OPTIMIZATION PROBLEM STATEMENT

Given the definitions earlier, we are now in a position to formally state the solution to the minimum time pursuit problem. To facilitate understanding, this section starts off by describing a sufficient condition by which capture is guaranteed. The target passes by node 1 at time 0. The UAV arrives for the 1st time at node 1 at time \( t_0 > 0 \) and is tasked with capturing the target. One can easily imagine (see Fig. 1) that when \( t_0 \) is small, capture should be possible, given the speed advantage (2). On the other hand, if \( t_0 \) is large, the target will likely escape, no matter what the UAV does. For a given \( t_0 \), we are interested in computing the pursuit policy, if it exists, that leads to capture in minimum time under worst-case target actions.

Let \( \mathcal{M}(1|\mathcal{I}(t_0), t_0) \) be the worst-case minimum time to capture the target from node 1 under the uncertainty \( \mathcal{I}(t_0) \). The target path information available to the UAV at node 1 at time \( t_0 \) is given by \( \mathcal{I}(t_0) = \{1, \ldots, n\} \). In a similar fashion, we define \( \mathcal{M}(p(t)|\mathcal{I}(t), t) \) to be the worst-case minimum time to capture the target from \( p(t) \) under the uncertainty \( \mathcal{I}(t) \). Also, let \( \mu(p(t)|\mathcal{I}(t), t) \in \{1, \ldots, m\} \) be the corresponding UGS index to which the UAV should head towards next, to enable capture. It is possible that the UAV cannot guarantee capture of the target from \( p(t) \) armed with information \( \mathcal{I}(t) \). In this case, we set \( \mathcal{M}(p(t)|\mathcal{I}(t), t) = \infty \).

**Condition for Guaranteed Capture:** A sufficient condition for guaranteed capture is that at some time decision time \( t \),

\[
\exists j \in \{1, \ldots, m\} \text{ s.t. } \mathcal{I}(t) \subseteq \mathcal{P}_j, \text{ and } t + d_{\mathcal{V}}(p(t), j) \leq L_j(k), \forall k \in \mathcal{I}(t). \tag{7}
\]

The above condition implies that the information available to the UAV is that all possible paths taken by the target must go through node \( j \) and furthermore, the UAV can travel to \( j \) before the target can get there, regardless of the path it chooses. If the above condition is met, the corresponding minimum time to capture (cost to go) is given by:

\[
\mathcal{M}(p(t)|\mathcal{I}(t), t) = \min_{u} \max_{k \in \mathcal{I}(t)} L_u(k) - t, \tag{8}
\]

\[
\text{s.t. } t + d_{\mathcal{V}}(p(t), u) \leq L_u(k), \forall k \in \mathcal{I}(t).
\]

The control action that leads to capture is given by: \( u(t) = u^*, \) where, \( u^* \) is the minimizer in (8). So, the UAV proceeds to \( u^* \) and waits until the target gets there. The waiting time is given by:

\[
w = \max_{k \in \mathcal{I}(t)} L_{u^*}(k) - d_{\mathcal{V}}(p(t), u^*) - t, \tag{9}
\]

Since the target path is still uncertain, the UAV has to wait for the longest time (worst-case) corresponding to the latest time that the target will get to \( u^* \). We note that the above condition is also necessary for guaranteed capture in a single move by the UAV. In other words, if the UAV wishes to move to a new location \( u \neq p(t) \) and capture the target at \( u \) no matter what the target does, then (7) must be satisfied.

A. Min-Max Optimization

Suppose at time \( t \), the UAV is at UGS index \( p(t) \) with path information \( \mathcal{I}(t) \) and decides to visit \( u \) next. Upon reaching \( u \), the information will change to: \( \mathcal{I}(t^+|u, y) \), where \( y \) is the observation that the UAV will make at \( u \). Recall that \( \mathcal{I}(t^+|u, y) \) is updated according to (5) and (6) for the red and green UGS observations respectively. So, the minimum time to capture from \( p(t) \) satisfies the recursion:

\[
\mathcal{M}(p(t)|\mathcal{I}(t), t) = \min_{u \in \mathcal{U}} \left\{ \begin{array}{l}
d_{\mathcal{V}}(p(t), u) + w(t, u, \mathcal{I}(t)) \\
+ \max_{y} \mathcal{M}(u|\mathcal{I}(t^+|u, y), t^+) \end{array} \right\}. \tag{10}
\]

where the next decision epoch \( t^+ \) is defined according to (4). This is so because, before visiting \( u \), the UAV cannot know whether the observation will be a red or green UGS. Hence, to guarantee capture, it has to assume the worst-case scenario that will result in the larger of two possible capture times at \( u \). To compute the capture time from \( p(t) \), we add the travel time from \( p(t) \) to \( u \) or the waiting time if \( u = p(t) \). Finally, we take the min over all possible nodes to get the least possible capture time from \( p(t) \). If the UAV chooses \( u = p(t) \), the waiting time is given by:

\[
w(t, u, \mathcal{I}(t)) = \min_{k \in \mathcal{I}(t)} L_u(k) - t. \tag{11}
\]

**Boundary Conditions:** The two boundary conditions corresponding to capture and escape, for the recursion (10) are given below.

- **BC-1** If the state \( p(t), \mathcal{I}(t) \) satisfies the necessary condition for guaranteed capture in a single move (7), then the corresponding capture time is given by (8).

- **BC-2** If the current time \( t > T_{\mathcal{E}}(s_{k,t}) \) for some \( k \in \mathcal{I}(t) \), then the target would escape (in the worst-case) and we set \( \mathcal{M}(p(t)|\mathcal{I}(t), t) = \infty \).

We have to use the recursive equation (10) to compute
Note that by definition, the recursion and the associated tree search is combinatorial in nature and hence, any reduction in the search space would be useful. In this context, we introduce a control constraint, $u \in B(\mathcal{I}(t_0)) \subset \mathcal{U}$ in (10) that will enable us to compute $\mathcal{M}(1|\mathcal{I}(t_0), t_0)$, without loss of optimality, in an efficient manner. In the next section, we enumerate this approach.

**B. Optimal Control Constraint**

Recall that the necessary condition for guaranteed capture in one move (7) requires that all paths in the uncertainty set go through some UGS, $j$. In addition, the UAV needs to arrive at the UGS $j$ before the target gets there, so as to intercept it. It stands to reason that in the beginning of the search effort, the UAV should reduce the uncertainty and it does so by reducing the size/cardinality of the uncertainty set. So the choice of control $u$ in (10) should be such that, by going to it, the UAV can reduce the uncertainty set $\mathcal{I}(t)$ to $\mathcal{I}(t^+)$ such that $|\mathcal{I}(t^+|u, y)| < |\mathcal{I}(t)|$. Here we use $|$ to denote the size of the uncertainty i.e., the number of paths in the set. In other words, the UAV will visit $u$ only if there is a possibility that information is, or will become, available at $u$ on whether or not the target took a path through $u$. Else, there is no value in visiting $u$ and one may as well ignore it. Indeed, we can re-write the recursive equation (10) in the following manner:

$$
\mathcal{M}(p(t)|\mathcal{I}(t), t) = \min_{u \in B(\mathcal{I}(t))} \left\{ d_V(p(t), u) + w(t, u, \mathcal{I}(t)) \right\} + \max_y \mathcal{M}(\mathcal{I}(t^+|u, y), t^+),
$$

(12)

where the waiting time for $u = p(t)$ is given by (11). Let $\mathcal{I}'(u) = \mathcal{I}(t) \cap \mathcal{P}_u$ and $\mathcal{I}^g(u) = \mathcal{I}(t) \setminus \mathcal{I}'(u)$. The three distinct possibilities at $u$ are:

1) $\mathcal{I}'(u) = \mathcal{I}(t)$ which implies that capture is possible at $u$.

2) $\mathcal{I}'(u) \subset \mathcal{I}(t)$ which implies that the uncertainty is reduced at $u$ for both red and green observations.

3) $\mathcal{I}'(u) = \emptyset$ which implies that a green UGS is the only possible observation at $u$.

We define the restriction as follows: $B(\mathcal{I}(t)) = \{ u : \mathcal{I}'(u) \subseteq \mathcal{I}(t), u \in \mathcal{U}\}$. The restriction above implies that the UAV will visit $u$ only if one of two things happen. Either capture is possible at $u$ or the uncertainty is reduced at $u$ for either observation (red or green). The third possibility 3) implies that the only possible observation at $u$ is a green UGS with no reduction in the uncertainty! Clearly, in this case, there is no information to be gained by visiting $u$ and hence it can be removed from consideration.

Furthermore, from the triangle inequality constraint (1), it follows that the only reason to visit $u$ under possibility 1) is to immediately capture the target at $u$. As before, there is no value in visiting $u$ otherwise, since there is no additional information available at $u$. So, we have the following result.

**Lemma 1:** The optimal control $u$ to (12) is such that $\mathcal{I}'(u) = \mathcal{I}(t)$ iff $(p(t), \mathcal{I}(t))$ satisfy (7) and $u$ is the minimizer to (8).

In closing, to compute $\mathcal{M}(1|\mathcal{I}(t_0), t_0)$ for any initial delay $t_0 > 0$, we employ the recursion (12). The recursion progresses until one of the boundary conditions, BC-1 or BC-2 is met. Note that the optimal pursuit strategy is constrained to enforce a reduction in entropy at every move! Indeed the entropy i.e., the cardinality of the uncertainty set will reduce, at least by 1, for every move (including waiting) made by the UAV. As a result, if capture is guaranteed, the game will terminate in no more than $n$ steps/moves!

---

**III. EXAMPLE PROBLEM**

We shall illustrate the recursion (12) on the example problem illustrated in Fig. 1. To make things concrete, we re-draw the example road network in Fig. 2a, with a grid in the background, to highlight the $(x, y)$ coordinates of nodes and the distances along edges. In Fig. 2b, we show the four different target paths (ordered from left to right) along with the target’s time of arrival $E_j(k)$ (in parentheses) at nodes along each path. Indeed, $P_1 = (1 \to 3 \to 5)$, $P_2 = (1 \to 3 \to 4 \to 6)$, $P_3 = (1 \to 3 \to 4 \to 7)$ and $P_4 = (1 \to 2 \to 7)$. We assume that the UAV travels
between any two nodes at the constant speed, $V = 1.62$. For a initial delay, $t_0 = 4$ at node 1, the optimal control policy and cost to go are shown in Fig. 3. So, the minimum time to capture from node 1, $\mathcal{M}(1|\{1, 2, 3, 4\}, t_0 = 4) \approx 13.537$. Also shown is the pursuit policy, with the UGS to go to next shown below for both red and green observations (left to right respectively). For instance, the optimal UAV action from node 1 is to go to node 2. If node 2 is red, the UAV goes to node 7 and waits for $w \approx 2.205$ time units. If node 2 is green instead, the UAV goes to node 5 and so on, until finally capture occurs at one of the exit nodes. Also shown is the waiting time (if any) at the capture nodes. Note that for initial delay, $t_0 = 4$, the UAV can do no better than capture the target at one of the exit nodes. If we reduce the initial delay to $t_0 = 2$, we see that the performance is much better as shown in Fig. 4. Indeed, the target is either captured at node 3 or in the worst-case at node 7. The corresponding min-max time to capture is given by $\mathcal{M}(1|\{1, 2, 3, 4\}, t_0 = 2) \approx 12.657$.

A. Reducing the Computational Burden

One can achieve additional savings if the search paths that lead to escape can be identified sooner. Indeed, let $\mathcal{D}(p(t)|\mathcal{I}(t))$ be the latest time the UAV can arrive at/leave node $p(t)$ and guarantee capture, armed with the path information $\mathcal{I}(t)$. In a related work [2], the authors have elaborated on a bottoms-up approach to compute $\mathcal{D}(j|\mathcal{I})$ for a general road network with the same assumptions as was made here. Indeed, for the example problem, $\mathcal{D}(j|\mathcal{I})$ can be computed $\forall j \in \mathcal{U}$ and $\forall \mathcal{I} \in \mathcal{Z}$. With this information in hand, one can re-define the boundary condition for escape as follows:

**BC-2** If the current time and uncertainty set satisfy $t > \mathcal{D}(p(t)|\mathcal{I}(t))$, then the target would escape (in the worst-case) since the current time has exceeded the latest arrival time at $p(t)$ with a capture guarantee. And so, we set $\mathcal{M}(p(t)|\mathcal{I}(t), t) = \infty$.

B. Partial Information, Dynamic Game, and Dual Control

We are calculating the minimum time to capture pursuit strategy at node 1, if it exists, which guarantees the target’s capture before the latter reaches one of the the goal nodes, $j \in \mathcal{G}$. This is a deterministic pursuit-evasion game on a directed acyclic finite graph where the target’s strategy is open-loop control and the UAV has partial information. Such a game was previously considered in [3], [4], where the highly structured graph considered therein, was a Manhattan grid. Due to the UAV’s information pattern, which is restricted to partial observations of the physical state of the dynamic game, we are running into the difficulties brought about by the dual control effect [5], where the current information state determines the UAV’s optimal control while at the same time the information that will become available to the UAV will be in part determined by his current control.

IV. CONCLUSIONS

The optimal control of a UAV with limited sensing capability tasked with intercepting a blind target on a road network instrumented with UGS is considered. The UAV is interrogating the UGS, some of which were triggered by the moving target, and as such has access to partial observations only of the physical system’s state. A deterministic pursuit-evasion game on a directed acyclic finite graph where the blind target’s strategy is open-loop control and the UAV has partial information, is solved. Due to the UAV’s information pattern, which is restricted to partial observations of the physical state of the deterministic game at hand, the difficulties brought about by partial information in a dynamic game setting and the attendant dual control effect, could not be avoided; whence the computational complexity of the solution algorithm.

REFERENCES


