Security-Aware Synthesis of Human-UAV Protocols*

Mahmoud Elfar, Haibei Zhu, M. L. Cummings and Miroslav Pajic

Abstract— **In this work, we synthesize collaboration protocols for human-unmanned aerial vehicle (H-UAV) command and control systems, where the human operator aids in securing the UAV by intermittently performing geolocation tasks to confirm its reported location. We first present a stochastic game-based model for the system that accounts for both the operator and an adversary capable of launching stealthy false-data injection attacks, causing the UAV to deviate from its path. We also describe a synthesis challenge due to the UAV's hidden-information constraint. Next, we perform human experiments using a developed RESCHU-SA testbed to recognize the geolocation strategies that operators adopt. Furthermore, we deploy machine learning techniques on the collected experimental data to predict the correctness of a geolocation task at a given location based on its geographical features. By representing the model as a delayed-action game and formalizing the system objectives, we utilize off-the-shelf model checkers to synthesize protocols for the human-UAV coalition that satisfy these objectives. Finally, we demonstrate the usefulness of the H-UAV protocol synthesis through a case study where the protocols are experimentally analyzed and further evaluated by human operators.**

I. Introduction

Contrary to what the terminology may suggest, autonomous systems mostly involve human presence; from actively engaging with the system to merely monitoring the system status or intervening whenever necessary [1]. A typical example is the human-unmanned aerial vehicle (H-UAV) command and control system, where various deployed applications depend on having human operators responsible for supervising a fleet of UAVs during a mission. The operator performs various supervisory tasks including, for example, updating mission goals, monitoring agent status, and adjusting flight plans [2]. The operator can also be assigned primary tasks that are mission relevant, such as imagery tasks.

With the human presence, considering human factors becomes an essential part of the modeling and design of those systems, for which several attempts have been introduced in literature. The reliance on experimental data has been proposed to model human-autonomy interactions in various applications such as autonomous cars [3], [4], industrial [5], [6] and social robotics [7]. Due to their ability to model reactive systems, Markovian formalisms, e.g., markov decisions processes (MDPs) and stochastic games, were exploited for the theoretical exploration of human-robot interactions [8].

On the other hand, UAV navigational systems have been recently proven to be vulnerable to cyber and physical attacks, such as false-data injection attacks that target GPS receivers [9], [10], raising security concerns in this domain. A number of studies have focused on attack detection via sensor redundancy (e.g., [11–14]). Yet a class of these attacks can remain stealthy by introducing non-aggressive and incremental deviations [15–18]. Although humans are likely to surpass autonomy in such situations of high uncertainty [19], no research has addressed the human role in ensuring the security of H-UAV systems; such as, whether we can improve the overall security guarantees by harnessing the human power of inductive reasoning and the ability to provide context and additional information to the system in real-time.

In this paper, we focus on synthesis of protocols for H-UAV systems where the operator can intermittently perform geolocation tasks to aid in detection of possible attacks. First, the system dynamics, operator geolocation task, and the adversarial behavior are modeled using stochastic games. By developing RESCHU-SA testbed, experiments were conducted to understand operators strategies during geolocation tasks. Next, we use machine learning to predict correctness of a geolocation task at a given location. Note that in security problems, the system (i.e., UAV) is not aware of the information related to attacker's actions, which presents a significant synthesis challenge. Thus, we construct the model as a delayed-action game, which allows for the use of off-the-shelf tools (PRISM-games) to synthesize security-aware H-UAV protocols; such protocols provide UAV path plans that increase chances of attack detection. Moreover, the protocols specify time instances at which the operator is advised to perform a geolocation task, maximizing its correctness. The formal synthesis of the advisory system guarantees a limit to the workload level to avoid performance deterioration without compromising the system security. Finally, we present a case study where the synthesized protocols are analyzed and subjectively evaluated by human operators.

The rest of this paper is organized as follows. Section II provides a background on stochastic games and H-UAV control systems before formulating the problem statement. Section III presents the system modeling using stochastic and delayed-action games, while Section IV describes the experiments used to realize the model parameters. The protocol synthesis framework is illustrated in Section V. Section VI provides a case study where the synthesized protocols are analyzed and evaluated. Finally, Section VII concludes the paper and provides a discussion and future directions.

II. Background and Problem Statement

We start with the related background on stochastic games and strategy synthesis, followed by the problem statement.

A. Stochastic Games

Stochastic multiplayer games (SMGs) can model reactive systems with both stochastic and nondeterministic transitions, where the latter are resolved by more than one player. *Stochasticity* arises when the system evolution cannot be precisely predicted, yet a probabilistic profile can be assumed.

The authors are with the Department of Electrical & Computer Engineering, Duke University, USA. {mahmoud.elfar, haibei.zhu, mary.cummings, miroslav.pajic}@duke.edu.

*This work was supported by the NSF CNS-1652544 grant, and by the ONR under agreements number N00014-17-1-2012 and N00014-17-1-2504.

Conversely, *nondeterminism* abstracts players' decisions, either to incorporate a family of behaviors or to reason about a winning strategy. A *turn-based* game is played such that, at each state, only one player at most can make decisions. Turn-based SMGs have proven to be useful for modeling reactive systems [20].

Definition 1 (Turn-Based SMGs)**.** *A turn-based SMG over players* $\Gamma = \{\mathrm{I}, \mathrm{II}, \bigcirc\}$ *is a tuple* $\mathcal{G} = \langle S, (S_{\mathrm{I}}, S_{\mathrm{II}}, S_{\bigcirc}), A, \varsigma, \delta\rangle$*, where* S *is a finite set of states, partitioned into players' and stochastic states* S_{I}*,* S_{II} *and* $S_{\bigcirc}$*;* A *is a finite set of actions;* $\varsigma \in Dist(S_{\mathrm{II}})$ *is an initial distribution over* S_{II}*;* $\delta : S \times S \to [0,1]$ *is a transition function s.t.* $\delta(s,s') \in \{0,1\}$*,* $\forall s \in S_{\mathrm{I}} \cup S_{\mathrm{II}}$ *and* $s' \in S$*, and* $\delta(s,s') \in [0,1]$*,* $\forall s \in S_{\bigcirc}$ *and* $s' \in S_{\mathrm{I}} \cup S_{\mathrm{II}}$*, where* $\sum_{s' \in S} \delta(s,s') = 1$ *holds.*

In contrast to SMGs where the game state is visible to all players, in [21] we introduced *delayed-action games* (DAGs) that partially obscure the game state from one player by substituting the hidden information (i.e., the truth) regarding some states with the player's belief for these hidden states.

Definition 2 (Delayed Action Game)**.** *A delayed-action game (DAG) is a tuple* $\hat{\mathcal{G}} = (\mathcal{G}, V, \Gamma)$ *where* $\mathcal{G}$ *is a turn-based SMG over a set of game variables* $V = V_{\mathcal{T}} \cup V_{\mathcal{B}}$ *and players* $\Gamma = \{\mathrm{I}, \mathrm{II}, \bigcirc\}$*, such that* $S \subseteq \mathsf{Eval}\,(V) \times \mathcal{P}(\mathsf{Eval}\,(V_{\mathcal{T}})) \times \Gamma$*. In addition,* $V_{\mathcal{T}}$ *is the set of variables holding the true values of the game, known to player* I*; and* $V_{\mathcal{B}}$ *is the set of variables holding player* II *beliefs about the values of the game.*

In Definition 2, for any variable $\mathtt{var}$ from a set V, $\mathsf{Eval}\,(\mathtt{var})$ denotes the set of evaluations that assign values to $\mathtt{var}$. Also, for a set A, $\mathcal{P}(A)$ refers to the power set 2^A.

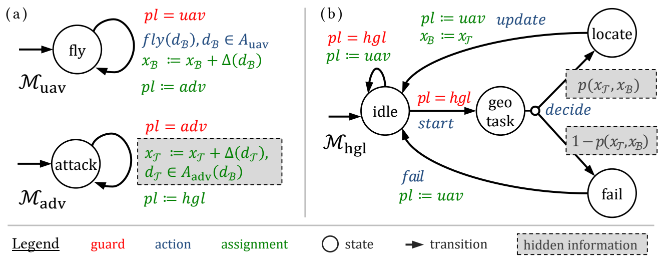
Intuitively, a player's *strategy* is how she resolves nondeterminism throughout the game, while a *protocol* is a set of strategies adopted by a coalition of players. The synthesis problem seeks a winning strategy (protocol); that is, it resolves choices for a player (coalition) such that some objectives are satisfied. Synthesis *objectives* can be specified using temporal logic such as ATL and rPATL [22].

B. *Human-UAV Supervisory Systems*

This work is motivated by H-UAV supervisory systems, where the UAV is supervised by a human operator [1]. A typical mission is for the UAVs to reach a number of targets to perform imagery tasks. While an autonomous planner automatically assigns UAVs to target locations, the operator can override these assignments or the path plan if necessary. Once a target is reached, the operator is notified to assist with the imagery task by analyzing the live camera feed [2].

UAVs are prone to adversarial attacks, such as GPS spoofing, that can drive the UAV away from its planned path [9], [10], [23]. Several techniques have been proposed to aid with detecting such attack by relying on redundant sensors (e.g., [11–14]). Nevertheless, when a sufficient number of sensors is compromised, a smart attacker can remain stealthy by injecting non-aggressive and incremental deviations in sensor measurements, which can still force the UAV into any undesired state through the actions of controller [15–18].

On the other hand, a *geolocation task* ultimately aims at localizing the UAV through side channel information. For example, we showed that landmarks and other geo-features from a UAV's camera feed can be used to estimate the UAV's location in real time by a human-operator [24]. Thus, in this work, we address the problem of synthesizing protocols for the Human-UAV supervisory system that employs the human operator for attack detection. The synthesized protocols shall provide the UAV path plan, as well as the time instances at which the operator shall be advised to perform a geolocation task, while ensuring that a given set of performance objectives (e.g., time, safety, workload) is satisfied.



(a)
fly
$\mathcal{M}_{\mathrm{uav}}$
$pl = uav$
$fly(d_{\mathcal{B}}), d_{\mathcal{B}} \in A_{\mathrm{uav}}$
$x_{\mathcal{B}} := x_{\mathcal{B}} + \Delta(d_{\mathcal{B}})$
$pl := adv$
attack
$\mathcal{M}_{\mathrm{adv}}$
$pl = adv$
$x_{\mathcal{T}} := x_{\mathcal{T}} + \Delta(d_{\mathcal{T}}),$
$d_{\mathcal{T}} \in A_{\mathrm{adv}}(d_{\mathcal{B}})$
$pl := hgl$
(b)
$pl = hgl$
$pl := uav$
$pl := uav$ update
$x_{\mathcal{B}} := x_{\mathcal{T}}$
locate
idle
$pl = hgl$
start
geo task
decide
$p(x_{\mathcal{T}}, x_{\mathcal{B}})$
$1 - p(x_{\mathcal{T}}, x_{\mathcal{B}})$
$\mathcal{M}_{\mathrm{hgl}}$
fail
$pl := uav$
fail
Legend
guard
action
assignment
state
transition
hidden information

Fig. 1. SMG-based model components: (a) UAV ($\mathcal{M}_{\mathrm{uav}}$) and adversary ($\mathcal{M}_{\mathrm{adv}}$), and (b) human geolocation ($\mathcal{M}_{\mathrm{hgl}}$).

III. System Modeling

In this section, we describe a system model, before showing how it can form a DAG used for the protocol synthesis.

A. *SMG-Based Model*

Due to the stealthy nature of cyber attacks considered in this study, one should differentiate between the UAV *belief* about its location, which may not be accurate, and the ground *truth*, which is assumed to be known to the adversary. Fig. 1(a) shows a standard UAV model $\mathcal{M}_{\mathrm{uav}}$ (e.g., as in [25]) where the UAV movement is discretized into action set $A_{\mathrm{uav}} = \{\mathsf{N}, \mathsf{S}, \mathsf{E}, \mathsf{W}, \mathsf{NE}, \mathsf{NW}, \mathsf{SE}, \mathsf{SW}\}$. Note that the UAV's actions affect its belief $x_{\mathcal{B}}$. Conversely, $\mathcal{M}_{\mathrm{adv}}$ shows how the adversary's stealthy actions are constrained by the UAV movements, influencing the ground truth $x_{\mathcal{T}}$ to avoid detection, only slowly increasing deviations from the planned trajectory can be achieved by the attacker; e.g., through GPS spoofing and actions of the low-level controller, small errors are added to the desired commands in each step. If, for example, the UAV is heading N, then the adversary available actions are $A_{\mathrm{adv}}(\mathsf{N}) = \{\mathsf{NE}, \mathsf{N}, \mathsf{NW}\}$. Note that more aggressive attacks can be detected as in e.g., [11], [12].

As in Fig. 1(b), the human geolocation model $\mathcal{M}_{\mathrm{hgl}}$ can initiate a geolocation task via the action *start*. The outcome of the task, however, can be successful or not with probabilities $p(x_{\mathcal{T}}, x_{\mathcal{B}})$ and $1 - p(x_{\mathcal{T}}, x_{\mathcal{B}})$, respectively, depending primarily on both $x_{\mathcal{B}}$ and $x_{\mathcal{T}}$[1] — detailed probability modeling is provided in Section IV. While the presented model is a standard SMG, since $x_{\mathcal{T}}$ is unknown to the UAV — and thus so is $p(x_{\mathcal{T}}, x_{\mathcal{B}})$ — the model cannot be used to reason about strategies. Moreover, if SMG semantics are used (i.e., the truth is implicitly known to the UAV), a synthesized strategy becomes a function of the adversarial specific actions — which are unknown — thus, rendering the strategy useless.

[1]Other factors affecting p, such as operator skills and the current workload, are not considered in this study. Also, the model assumes that task repetition has no impact on p. See the discussion in Section VII.

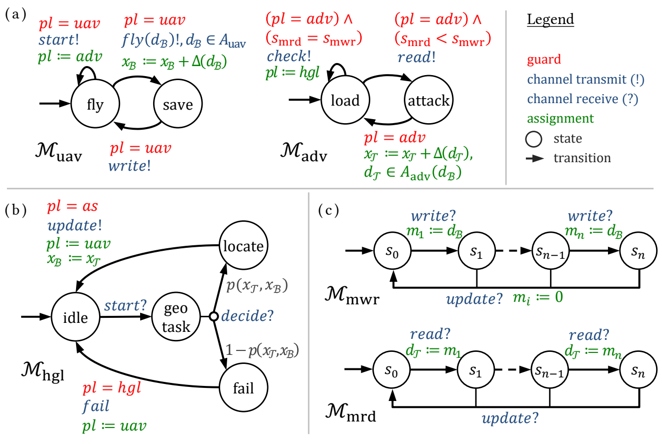


Fig. 2. DAG-based model with the UAV ($\mathcal{M}_\text{uav}$); adversary ($\mathcal{M}_\text{adv}$); human geolocation ($\mathcal{M}_\text{hgl}$); and DAG memory read ($\mathcal{M}_\text{mrd}$), write ($\mathcal{M}_\text{mwr}$).

Algorithm 1: DAG construction procedure

```
Input: Initial SMG model: $\hat{\varsigma}_0, \mathcal{M}_\text{adv}, \mathcal{M}_\text{uav}, \mathcal{M}_\text{hgl}, \mathcal{M}_\text{mwr}, \mathcal{M}_\text{mrd}$
Result: $\hat{\mathcal{G}}$: DAG construct
while end criterion not met do
    while $a_\text{uav} \neq$ geolocation task do                    /* UAV turn */
        $\mathcal{M}_\text{uav}.x_\mathcal{B} \leftarrow effect\ (a_\text{uav}, x_\mathcal{B})$ UAV moves, updating belief
        $\mathcal{M}_\text{mwr}.write(a_\text{uav}, {+}{+}wr)$ write action to memory
    while $wr \leqslant rd$ do                                       /* ADV turn */
        $\mathcal{M}_\text{mrd}.read(a_\text{uav}, {+}{+}rd)$ read UAV action from memory
        $\mathcal{M}_\text{adv}.x_\mathcal{T} \leftarrow effect\ (\beta(a_\text{uav}), x_\mathcal{T})$
    if draw $x \sim Brn(f(x_\mathcal{T}, x_\mathcal{B}))$ then               /* Stochastic turn */
        $\mathcal{M}_\text{uav}.x_\mathcal{B} \leftarrow \mathcal{M}_\text{adv}.x_\mathcal{T}$ update belief to match truth
        $reset(wr, rd)$ reset memory
    else $reset(rd)$ forget (hide) ADV actions
```

B. DAG-Based Model

To overcome these challenges, a DAG variation of the SMG model can be used (Fig. 2). The basic idea is that, during a mission, the UAV is unaware of differences between its belief and the ground truth until a geolocation task is correctly done. Thus, although we model missions as the players taking turns (SMG model), the same behavior can be captured if the UAV makes its moves, updating only the belief state, until a geolocation occurs; followed by the adversary's corresponding actions, updating the ground truth. With such a DAG model, the UAV takes a number of actions ahead of the adversary; that is, the adversary's actions are hidden from the UAV as they have not occurred yet in the model. Since the UAV decisions are made without knowing the adversary's specific actions, a UAV synthesized strategy becomes independent of those actions. Hence, the DAG model provides a different representation of the system without altering its behavior, as we show in [21], and thus can be used to synthesize strategies for the original SMG model. This mechanism is realized by the auxiliary components $\mathcal{M}_\text{mwr}$ and $\mathcal{M}_\text{mrd}$ that model write and read operations, respectively, on a FIFO memory stack $(m_i)_{i=0}^{n-1} \in A_\text{uav}^n$ as shown in Fig. 2(c). By synchronizing with $write$, $\mathcal{M}_\text{mwr}$ saves the last UAV action in memory location m_i, executing $s_i \xrightarrow{write} s_{i+1}$, $i \in \{0, ..., n{-}1\}$. Similarly, $\mathcal{M}_\text{mrd}$ synchronizes with the action $read$ to read the saved UAV action from memory location m_j, executing $s_j \xrightarrow{read} s_{j+1}$, $j \in \{0, ..., n{-}1\}$.

The DAG is formed by composing both the model and auxiliary components using the procedure summarized in Algorithm 1. The composed DAG allows the UAV ($\mathcal{M}_\text{uav}$) to play a finite number of actions before attempting a geolocation task, updating its belief $x_\mathcal{B}$. Based on these actions, the adversary ($\mathcal{M}_\text{adv}$) plays her delayed actions, updating the ground truth $x_\mathcal{T}$. Hence, the probability of performing a correct geolocation task is given by some function $f(x_\mathcal{T}, x_\mathcal{B})$ — the realization of the function $f(x_\mathcal{T}, x_\mathcal{B})$ is discussed in Section IV. If the geolocation task is unsuccessful, the adversarial actions are discarded and the UAV can continue to move or attempt another task. Otherwise, the UAV belief is updated to match the ground truth, at which point the game is repeated with a new starting location. In the rest of the paper, we will refer to each of these repetitions as a *subgame* $\hat{\mathcal{G}}_i$, where i is the subgame index, and to the set of all subgames of interest as the *supergame* $\hat{\mathcal{G}}$. Intuitively, each subgame explores the possible adversarial effect for each UAV sequence of actions, while the supergame examines the collective behavior of those subgames.

IV. Human Geolocation Model

As the presented DAG-based model is characterized by the probabilities $f(x_\mathcal{T}, x_\mathcal{B})$, in this section we present the experimental platform and evaluations used to obtain this function.

A. Experimental Platform and Design

The aim of this experiment was twofold. First, we wanted to validate the hypothesis that UAV operators can successfully perform geolocation tasks. Second, we wanted to understand what strategies the operators adopt to perform the geolocation tasks and their relevant factors.

To test our hypothesis, we developed Security-Aware RESCHU extension (RESCHU-SA),[2] a virtual platform for studying the security aspects of human-UAV supervisory systems prone to cyber attacks. Fig. 3 shows RESCHU-SA operator interface (OI) with the map area displaying flight plans, target locations, and threat zones. While the map shows a UAV's reported location, which may be compromised, the camera feed displays the live video stream from the selected UAV, from the UAV's true location. Through the OI, the operator can supervise a fleet of UAVs on a timed mission where a number of target locations should be visited by the UAVs, and a visual task should be completed by the operator through the UAV camera feed once each location is reached.

Each experiment consisted of two missions with high and low workloads. During each mission, a number of geolocation requests are randomly introduced, where the operator can respond by activating the camera feed to perform a geolocation task and further report whether the reported location is correct (see the attached video). After the two missions are done, the operator is interviewed and a retrospective verbal protocol analysis is performed to elicit more insights on their behavior and decisions made during the experiment.

[2]Research Environment for Supervisory Control of Heterogeneous Unmanned Vehicles (RESCHU) was originally developed to evaluate Human-UV supervisory systems [26], [27]. The initial platform does not support attack modeling or real-time video feed from any location from a map, based on UAVs' trajectories that the user can change during experiments.

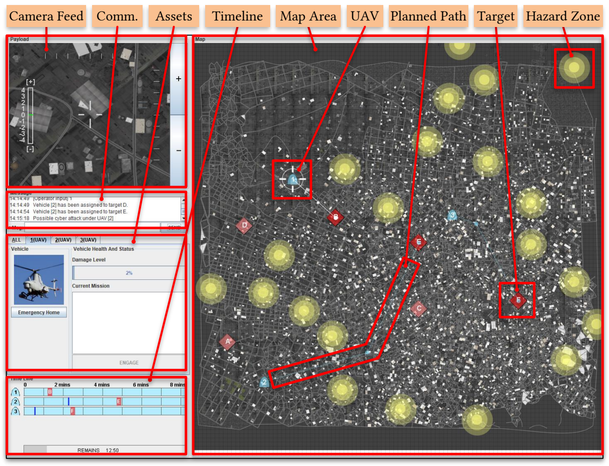


Fig. 3. RESCHU-SA operator interface (OI) elements. A UAV placement on the map is based on its *potentially compromised* GPS reported location.

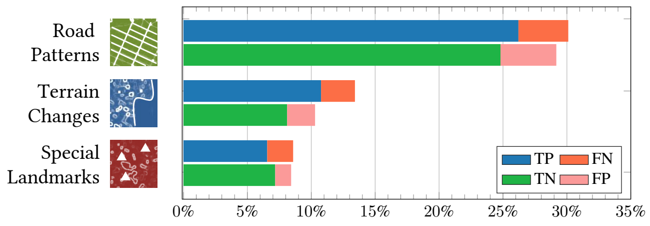


Fig. 4. Human geolocation strategies observed during the experiments. A true positive (TP) refers to a geolocation task confirming discrepancies while a GPS spoofing attack is active.

B. Human Geolocation Strategies

With 36 participants and a total of 641 geolocation attempts, the results of this experiment are summarized in Fig. 4 – due to space limitations, the full results can be found in [28]. Participants were found to adopt 3 main strategies for geolocationing: comparing road patterns (59.3%), observing terrains (23.7%), and examining landmarks (17.0%). Note that the road pattern-based strategy was used more than half of the time, with the lowest error rate (15.0%) compared to strategies based on terrains (20.4%) and landmarks (19.3%).

Since the results suggest that the operator's ability to compare the map with the camera feed depends heavily on the discrepancies between the geographic features of both scenes, capturing the map's geographic information system (GIS) in the model becomes critical to the synthesis problem. To this end, we represent a GIS layer as a labeling function that maps a location on the grid to a set of labels (i.e., atomic propositions) capturing the features of interest for that location. Table I lists the GIS layers relevant to the geolocation strategies, as obtained from our experiments, while Fig. 5 shows a 4-map example segments and their associated labels.

TABLE I

LIST OF THE GIS LAYERS AND THEIR SEMANTICS.

GIS Layer	Labels Set	Labeling Function
Terrains	$\Sigma_{tr} = \{rural, urban, water, \ldots\}$	$L_{tr} : \mathsf{Eval}(x) \rightarrow \mathcal{P}(\Sigma_{tr})$
Roads	$\Sigma_{rd} = \{rect, star, dense, \ldots\}$	$L_{rd} : \mathsf{Eval}(x) \rightarrow \mathcal{P}(\Sigma_{rd})$
Landmarks	$\Sigma_{lm} = \{building, tower, block, \ldots\}$	$L_{lm} : \mathsf{Eval}(x) \rightarrow \mathcal{P}(\Sigma_{lm})$

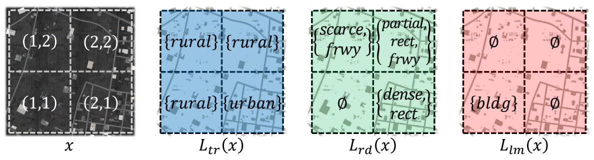


Fig. 5. An example of map segments and their associated GIS labels.

C. Human Geolocation Task (GT) Predictor

The abstract goal of a geolocation task is to determine whether the UAV's reported belief $x_{\mathcal{B}} = x_{\text{map}}$ (see Fig. 1(b)) matches the ground truth $x_{\mathcal{T}} = x_{\text{cam}}$ observed via the camera feed.[3] The operator's ability to correctly perform the task is affected by the GIS features of both x_{cam} and x_{map}. For instance, if $x_{\text{cam}} = x_{\text{map}}$ (i.e., they refer to the same location), the presence of distinctive features is likely to increase the chances of a correct decision. Conversely, if $x_{\text{cam}} \neq x_{\text{map}}$, then the probability of a correct decision relies heavily on the relative distinctiveness between both locations. Thus, we define a GT predictor as follows.

Definition 3. *GT predictor is a tuple $\mathcal{K} = (\Sigma, \mathcal{H}, f)$ where Σ is the set of GIS layers, $\mathcal{H} = \{r_i \mid r_i \colon X^2 \rightarrow \mathbb{R}, i = 1, \ldots, n\}$ is a set of n numerical measures of similarity between two locations in X, and $f \colon \mathcal{P}(\Sigma)^2 \times \mathbb{R}^n \rightarrow [0,1]$ is a prediction function of the GT correctness.*

Given two sets of labels and a tuple of numeric measures of similarity between two given locations, the function f predicts the correctness of a GT. In this case-study, we used $\Sigma = \cup_{j \in \{tr,rd,lm\}} \Sigma_j$; as a similarity measure we used the maximum normalized 2D cross-correlation coefficient $r_{rd} \in [0,1]$ between the road patterns of two locations- $\mathcal{H} = \{r_{rd}\}$.

We derive the prediction function f (and hence $\mathcal{K}$) using machine learning-based techniques. Basically, a predictive model is trained on a database of image pairs, where the training inputs are the GIS labels and similarity measures of each pair; the training output is the corresponding human estimation of the similarity of each pair; and the model outcome is $\hat{f} \colon \mathcal{P}(\Sigma)^2 \times \mathbb{R}^n \rightarrow [0,1]$. A survey was administrated to human participants to collect their opinion of how similar two locations are. During the survey, each participant was shown a total of 100 pairs of locations. For each pair, the participant was asked to select their estimation on a scale from 1 (very similar) to 5 (very dissimilar), as shown in Fig. 6. Next, labels from Σ were manually assigned to the selected locations. The function $\hat{f}$ was realized using a bagged-trees ensemble learner (RMSE = 0.635, RS = 0.650). Finally, for two locations x_{cam} and x_{map} we have that[4]

$$f\left(x_{\text{cam}}, x_{\text{map}}\right) = \begin{cases} \hat{f}\left(x_{\text{cam}}, x_{\text{map}}\right) & x_{\text{cam}} = x_{\text{map}} \\ 1 - \hat{f}\left(x_{\text{cam}}, x_{\text{map}}\right) & \text{otherwise.} \end{cases}$$

That is, the measure of correctness is the human tendency to consider two images similar or dissimilar if they represent the same or different locations, respectively.

[3] In this work, we only address the determination of whether two locations are the same, rather than how the actual coordinates can be found.

[4] For notational simplicity, we employ $f\left(x_{\text{cam}}, x_{\text{map}}\right)$ to refer to $f\left(L\left(x_{\text{cam}}\right), L\left(x_{\text{map}}\right), \mathcal{H}\left(x_{\text{cam}}, x_{\text{map}}\right)\right)$, where $L(x)$ is the suitable layer-labeling function, capturing the GIS layers at location x (as in Table I).

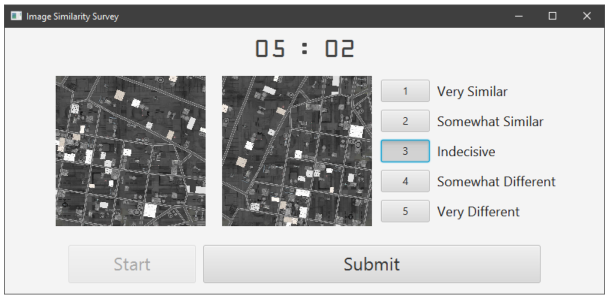


Fig. 6. Survey interface where image similarity is rated on a 5-point scale.

V. Protocol Synthesis

A. Synthesis Objectives

The primary synthesis objective is to find protocols for the H-UAV coalition based on the following requirements.

(a) *Reach the target location.* As one subgame may not yield a feasible flight plan to directly reach the target, checkpoints can be set as intermediate targets to render the objective feasible. By assigning the label *reach* to the set of states with acceptable checkpoint locations, the objective can then be formalized as $\mathrm{Pr}_{\max}\left[\mathsf{F}\ \mathit{reach}\right] \geqslant p_{\min}$ for some bound $p_{\min}$, and has to hold for all encountered subgames.

(b) *Avoid hazard zones.* To reach the target, the UAV must avoid all known hazard zones, where the UAV is likely to endure damage. If the corresponding states are assigned the label *hazard*, the objective can formalized as $\mathrm{Pr}_{\max}\left[\mathsf{G}\ \neg \mathit{hazard}\right] \geqslant p_{\min}$ for some bound $p_{\min}$. We assume here that encountering a hazard zone results in losing the asset, hence the global operator. Consequently, this objective has to hold for all encountered subgames.

These objectives are refined to elicit rPATL queries. Specifically, for a subgame $\hat{\mathcal{G}}_i$, the following query is used

$$\phi_{\mathrm{syn}}(k) \coloneqq \langle\!\langle \mathrm{uav} \rangle\!\rangle \mathrm{Pr}_{\max=?}\left[\neg \mathit{hazard}\ \mathsf{U}^{\leqslant k}\ (\mathit{locate} \wedge \mathit{reach})\right],$$

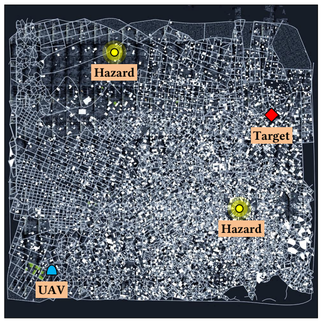
i.e., find a strategy that maximizes the probability of not encountering a hazard until a checkpoint is reached (*reach*) and a geolocation task is successful (*locate*) within a horizon k.The procedure described in Algorithm 2 is then used to solve the synthesis problem. Starting with the initial subgame, the procedure checks for each horizon whether ϕ_{syn} is satisfied (hence a strategy exists) if the geolocation task is performed at the last stage. These checks are terminated when reaching the maximum search horizon or failing to satisfy the objectives.[5] Next, subgames are pruned to discard strategies that fail to satisfy local bounds. The remaining strategies are used to populate the set of end locations, where each end-location represents an initial location to a reachable subgame that needs to be explored. The procedure is repeated for all subgames with initial state in the end locations set to obtain a strategy π_i for each reachable subgame $\hat{\mathcal{G}}_i$. Thus, for a q number of reachable subgames, the supergame is reduced to an MDP $\hat{\mathcal{G}}^{\{\pi_i\}_{i=1}^{q}}$ (whose states are the reachable subgames) which is checked against the query

$$\hat{\mathcal{G}}^{\{\pi_i\}_{i=0}^{q}}:\ \phi_{\mathrm{ana}}(n) \coloneqq \langle\!\langle \mathrm{adv} \rangle\!\rangle \mathrm{Pr}_{\min=?}\left[\mathsf{F}^{\leqslant n}\ \mathit{target}\right]$$

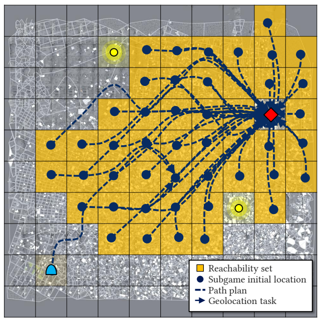
[5]If no strategy exists for a stage i, the same holds for all horizons of size $j > i$.

Algorithm 2: Protocol synthesis procedure

```
Input: Initial location $x_0$, synthesis query $\phi_{\mathrm{syn}}$, max horizon $h_{\max}$
Output: H-UAV protocols $\Pi = \{(\pi_{\mathrm{uav}}, \pi_h)\}$
$X \leftarrow \{x_0\}$ initialize set of initial locations (subgames)
foreach unexplored initial location $x_i \in X$ do
    $\hat{s}_0 \leftarrow (\mathrm{UAV}, x_i, \epsilon)$ set subgame initial state
    $stop \leftarrow \bot$, $h \leftarrow 1$ reset stopping flag and horizon
    while $h \leqslant h_{\max} \wedge \neg stop$ do
        $(\pi_{\mathrm{uav}}, \varphi) \leftarrow synth\left(\hat{\mathcal{G}}^{\pi_h}_{\hat{s}_0}, \phi_{\mathrm{syn}}\right)$ find a winning strategy
        if $\pi_{\mathrm{uav}}$ exists then
            $\Pi \leftarrow \Pi \cup (\pi_{\mathrm{uav}}, \pi_h, \varphi)$ add to the protocol
            $X \leftarrow X \cup reach(\pi_{\mathrm{uav}})$ update reachability set
            $h \leftarrow h + 1$ explore next horizon
        else $stop \leftarrow \top$
    $prune(\Pi)$
```



(a) Mission setup.



(b) Synthesized protocols.

Fig. 7. The mission setup used for experimental evaluation and the corresponding protocols. For clarity, the map colors are altered and the protocols are partially displayed.

find the minimum probability of the UAV eventually reaching the target within a maximum number of geolocation tasks n.

B. Synthesis Procedure

The protocol synthesis procedure is summarized in Algorithm 2. Starting from the initial location x_0 and horizon $h = 1$, the first subgame is constructed and explored. For each horizon, the synthesizer searches for a corresponding optimal strategy and set of reachable locations, until either the maximum search horizon $h_{\max}$ is reached or no feasible strategy can be found. The same process is repeated using unexplored locations in the reachability set. Note here that the set Π can be pruned (e.g., by discarding strategies that violates auxiliary requirements) to reduce computation time.

VI. Experimental Results

Fig. 7(a) shows the environment setup used for evaluation. The map was discretized into a $10{\times}10$ grid, where crossing the map boundaries is penalized for both $x_{\mathcal{B}}$ and $x_{\mathcal{T}}$. Also, for the UAV to ever arrive to the designated target, the adversary is prohibited from launching attacks for at least the first horizon. The UAV mission is to reach the target without encountering any of the hazard zones. The model shown in Fig. 2 was implemented using the PRISM-games [22] model checker on an Intel Core i7 4.0 GHz CPU with 16GB RAM.

From the synthesis procedure (Algorithm 2), the protocols to complete the mission were obtained as shown in Fig. 7(b). For the first subgame, Fig. 8(a) shows how the horizon at which the geolocation task is performed impacts the probability of correctness that can be guaranteed, regardless of the

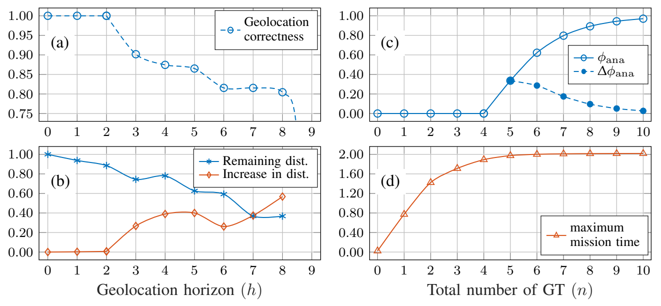


Fig. 8. The analysis results for the synthesized protocol. Times and distances are relative to those of the shortest path.

adversarial actions, while Fig. 8(b) shows both the remaining distance to target and the increase in traveled distance, relative to the shortest path. The trends show that, as time passes without performing a geolocation task, the probability of its correctness decreases, while the remaining distance mostly decreases as the UAV approach the target. Interestingly, the strategy where the geolocation task is scheduled at horizon $h = 6$ provides a local minimum for the increase in traveled distance while maintaining a probability of correctness close to $h = 7$ and $h = 8$. This probability drops at $h \geqslant 9$ as guaranteeing that the UAV never encounter a hazard zone before the scheduled geolocation task becomes infeasible.

For the supergame, Fig. 8(c) shows the impact of the geolocation task budget on the probability of a successful mission. The graph conveys that more than 4 geolocation tasks are required to guarantee a non-zero minimum probability of reaching the target against the worst-case attack. Fig. 8(d) shows the total expected mission time given a budget for the geolocation tasks, relative to the shortest-path time.

For subjective evaluation, an experiment was conducted where a number of participants were shown triplets of images (I_a, I_b, I_c) as shown in Fig. 9(left). In each triplet, I_a depicts a target location to which a shortest-path π_b and safest-path π_c plans are generated, where $\mathbb{I}_b$ and $\mathbb{I}_c$ are the sets of reachable locations if the UAV is under attack, respectively. The two images I_b and I_c are randomly withdrawn such that $I_b \in \mathbb{I}_b \setminus \mathbb{I}_c$ and $I_c \in \mathbb{I}_c \setminus \mathbb{I}_b$. Participants were asked to indicate which image is less likely to be confused with I_a (and hence safer). Fig. 9(right) shows that, while group A selections indicated no significant difference, group B indicated that locations from the safest-path plan are less likely to be confused with locations from the shortest-path plan. In retrospective interviews, group A explained that they picked the upper right image most of the time as the tight time constraints gave them no time to inspect the lower right image, while group B found the time to be adequate to inspect both. This may explain why I_b and I_c had similar chances of being selected by group A since the way they ordered was random. As these results highlight the effect of individual imagery skills on the geolocation task, further investigation will be done to thoroughly confirm this observation.

The performance results obtained for this case study are listed in Table II. Note that, for the same grid size, more complex maps require more time for model checking, while the state space size remains unaffected. Although the number of states is $\mathcal{O}\left((|A_{\text{uav}}||A_{\text{adv}}|)^h\right)$, the growth rate is typically reduced by the presence of hazard zones as the game stops branching at such states. Following the construction of the subgames, exploring the supergame itself consumes significantly less resources for model construction and analysis.

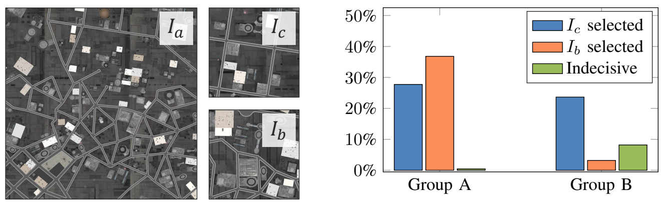


Fig. 9. (left) An example of the image triplets used for subjective evaluation, and (right) the evaluation results.

TABLE II
PERFORMANCE RESULTS FOR USING QUERIES ϕ_{syn} AND ϕ_{ana}.

Subgame $\hat{\mathcal{G}}_0$		Model Size			Time (sec)		
Map	h	States	Transitions	Choices	Model	ϕ_{syn}	ϕ_{ana}
10×10	3	948	1,196	1,112	2.716	0.012	–
	4	5,976	8,128	7,392	13.026	0.082	–
	5	28,345	39,964	37,243	67.208	0.380	–
	6	119,078	172,640	165,780	379.36	3.100	–
	7	490,021	722,575	709,033	1549.176	10.739	–
	8	2,165,888	3,228,811	3,203,108	5497.115	44.467	–
Supergame $\hat{\mathcal{G}}$		12,704	18,171	15,396	28.567	–	4.049

VII. Discussion and Conclusion

In this paper, we have presented an approach to synthesize collaboration protocols for human-UAV command and control systems. The approach utilizes delayed-action games to model the system such that the ground truth is hidden from the UAV, rendering the model useful for synthesis using off-the-shelf tools. As the geolocation task is used to confirm the UAV location, the model includes a measure of geolocation task correctness. Moreover, experimental results have shown that the operators mostly adopted three geolocation strategies. By extracting the GIS features relevant to these strategies, machine learning techniques were deployed to predict the aforementioned measure of correctness. Based on the developed model and a formal representation of system objectives, the DAG synthesis procedure uses PRISM-games model checker to synthesize and analyze the collaboration protocols. Finally, experimental analyses and subjective feedback were used to evaluate the synthesized protocols.

Though DAGs exploit parallel computation to reduce total processing time, the time to explore subgame horizons exponentially grows. Nevertheless, the DAGs as a formalism can benefit from approximate model checking techniques to explore larger horizons and higher orders of map discretization, where accuracy is traded-off with speed [29].

Another improvement to this study would be to consider other factors affecting operator performance, such as perceived workload, operator skill level, and proper quantification of task correctness. Albeit challenging, this may open the door for individualized protocol synthesis. When it comes to advisory systems, however, one must carefully examine factors affecting human trust in autonomy — a quality that, if absent, may render collaborative protocols ineffective. Hence, reinforcing trust through explanatory communication of those protocols is an avenue for future work.

References

[1] M. L. Cummings, S. Bruni, S. Mercier, and P. Mitchell, "Automation architecture for single operator, multiple uav command and control," Massachusetts Inst Of Tech Cambridge, Tech. Rep., 2007.

[2] C. E. Nehme, J. W. Crandall, and M. L. Cummings, "An operator function taxonomy for unmanned aerial vehicle missions," in *12th international command and control research and technology symposium*, 2007.

[3] D. Sadigh, K. Driggs-Campbell, A. Puggelli, W. Li, V. Shia, R. Bajcsy, A. L. Sangiovanni-Vincentelli, S. S. Sastry, and S. A. Seshia, "Data-driven probabilistic modeling and verification of human driver behavior," in *Formal Verification and Modeling in Human-Machine Systems: Papers from the AAAI Spring Symposium*, 2014.

[4] W. Li, D. Sadigh, S. S. Sastry, and S. A. Seshia, "Synthesis for human-in-the-loop control systems," in *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*. Springer, 2014, pp. 470–484.

[5] C. Pérez-D'Arpino and J. A. Shah, "Fast target prediction of human reaching motion for cooperative human-robot manipulation tasks using time series classification," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE, 2015, pp. 6175–6182.

[6] V. V. Unhelkar, P. A. Lasota, Q. Tyroller, R.-D. Buhai, L. Marceau, B. Deml, and J. A. Shah, "Human-aware robotic assistant for collaborative assembly: Integrating human motion prediction with planning in time," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 2394–2401, 2018.

[7] H. Admoni and B. Scassellati, "Data-driven model of nonverbal behavior for socially assistive human-robot interactions," in *Proceedings of the 16th international conference on multimodal interaction*. ACM, 2014, pp. 196–199.

[8] L. Feng, C. Wiltsche, L. Humphrey, and U. Topcu, "Synthesis of human-in-the-loop control protocols for autonomous systems," *IEEE Transactions on Automation Science and Engineering*, vol. 13, no. 2, pp. 450–462, 2016.

[9] T. E. Humphreys, B. M. Ledvina, M. L. Psiaki, B. W. O'Hanlon, and P. M. Kintner, "Assessing the spoofing threat: Development of a portable gps civilian spoofer," in *Radionavigation Laboratory Conference Proceedings*, 2008.

[10] A. J. Kerns, D. P. Shepard, J. A. Bhatti, and T. E. Humphreys, "Unmanned aircraft capture and control via gps spoofing," *Journal of Field Robotics*, vol. 31, no. 4, pp. 617–636, 2014.

[11] M. Pajic, I. Lee, and G. J. Pappas, "Attack-resilient state estimation for noisy dynamical systems," *IEEE Transactions on Control of Network Systems*, vol. 4, no. 1, pp. 82–92, March 2017.

[12] M. Pajic, J. Weimer, N. Bezzo, O. Sokolsky, G. J. Pappas, and I. Lee, "Design and implementation of attack-resilient cyberphysical systems: With a focus on attack-resilient state estimators," *IEEE Control Systems*, vol. 37, no. 2, pp. 66–81, April 2017.

[13] R. Ivanov, M. Pajic, and I. Lee, "Attack-resilient sensor fusion for safety-critical cyber-physical systems," *ACM Transactions on Embedded Computing Systems*, vol. 15, no. 1, pp. 21:1–21:24, Feb. 2016. [Online]. Available: http://doi.acm.org/10.1145/2847418

[14] J. Park, R. Ivanov, J. Weimer, M. Pajic, S. H. Son, and I. Lee, "Security of cyber-physical systems in the presence of transient sensor faults," *ACM Transactions on Cyber-Physical Systems*, vol. 1, no. 3, pp. 15:1–15:23, May 2017.

[15] C. Kwon, W. Liu, and I. Hwang, "Analysis and design of stealthy cyber attacks on unmanned aerial systems," *Journal of Aerospace Information Systems*, vol. 11, no. 8, pp. 525–539, 2014. [Online]. Available: https://doi.org/10.2514/1.I010201

[16] Y. Mo and B. Sinopoli, "False data injection attacks in control systems," in *First Workshop on Secure Control Systems*, 2010.

[17] Y. Mo, R. Chabukswar, and B. Sinopoli, "Detecting integrity attacks on scada systems," *Control Systems Technology, IEEE Transactions on*, vol. 22, no. 4, pp. 1396–1407, 2014.

[18] I. Jovanov and M. Pajic, "Relaxing integrity requirements for resilient control systems," *CoRR*, vol. abs/1707.02950, 2017. [Online]. Available: https://arxiv.org/abs/1707.02950

[19] M. L. Cummings, "Man versus machine or man+ machine?" *IEEE Intelligent Systems*, vol. 29, no. 5, pp. 62–69, 2014.

[20] T. Chen, V. Forejt, M. Kwiatkowska, D. Parker, and A. Simaitis, "Automatic verification of competitive stochastic systems," *Formal Methods in System Design*, vol. 43, no. 1, pp. 61–92, 2013.

[21] M. Elfar, Y. Wang, and M. Pajic, "Security-aware synthesis using delayed-action games," Duke University, Tech. Rep., 2019. [Online]. Available: https://cpsl.pratt.duke.edu/files/images/elfar2019dag.pdf

[22] T. Chen, V. Forejt, M. Kwiatkowska, D. Parker, and A. Simaitis, "Prism-games: A model checker for stochastic multi-player games," in *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*. Springer, 2013, pp. 185–191.

[23] C. Kwon, S. Yantek, and I. Hwang, "Real-time safety assessment of unmanned aircraft systems against stealthy cyber attacks," *Journal of Aerospace Information Systems*, vol. 13, no. 1, pp. 27–45, 2015.

[24] H. Zhu, M. Elfar, M. Pajic, Z. Wang, and M. L. Cummings, "Human augmentation of uav cyber-attack detection," in *Human-Computer Interaction International*, 2018.

[25] S. Thrun, W. Burgard, and D. Fox, *Probabilistic robotics*. MIT press, 2005.

[26] C. E. Nehme, "Modeling human supervisory control in heterogeneous unmanned vehicle systems," Massachusetts Institute of Technology, Tech. Rep., 2009.

[27] B. Donmez, C. Nehme, and M. L. Cummings, "Modeling workload impact in multiple unmanned vehicle supervisory control," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 40, no. 6, pp. 1180–1190, 2010.

[28] H. Zhu, M. L. Cummings, M. Elfar, Z. Wang, and M. Pajic, "Operator strategy model development in uav hacking detection," *IEEE Transactions on Human-Machine Systems*, 2018. [Online]. Available: https://cpsl.pratt.duke.edu/files/images/zhu2018operator.pdf

[29] Y. Kantaros and M. M. Zavlanos, "Sampling-based optimal control synthesis for multi-robot systems under global temporal tasks," *IEEE Transactions on Automatic Control*, 2018.