Synthesis of Human-in-the-Loop Control Protocols for Autonomous Systems

Lu Feng, *Member, IEEE*, Clemens Wiltsche, Laura Humphrey, *Member, IEEE*, and Ufuk Topcu, *Member, IEEE*

***Abstract*—We propose an approach to synthesize control protocols for autonomous systems that account for uncertainties and imperfections in interactions with human operators. As an illustrative example, we consider a scenario involving road network surveillance by an unmanned aerial vehicle (UAV) that is controlled remotely by a human operator but also has a certain degree of autonomy. Depending on the type (i.e., probabilistic and/or nondeterministic) of knowledge about the uncertainties and imperfections in the human–automation interactions, we use abstractions based on Markov decision processes and augment these models to stochastic two-player games. Our approach enables the synthesis of operator-dependent optimal mission plans for the UAV, highlighting the effects of operator characteristics (e.g., workload, proficiency, and fatigue) on UAV mission performance. It can also provide informative feedback (e.g., Pareto curves showing the trade-offs between multiple mission objectives), potentially assisting the operator in decision-making. We demonstrate the applicability of our approach via a detailed UAV mission planning case study.**

Note to Practitioners—**This paper is motivated by the problem of synthesizing operator-dependent optimal mission plans for unmanned aerial vehicles (UAVs) that take into account a variety of operator performance characteristics (e.g., workload, proficiency, and fatigue). While the proposed approach is demonstrated in the domain of human–automation UAV mission planning, in general it can be applied towards the design of control protocols (i.e., behaviors taken in response to external inputs) for any autonomous system that interacts with human operators. The benefit of this approach is that it directly considers human factors in the synthesis of autonomous control protocols, while existing approaches either synthesize control protocols without considering human factors or treat human–automation interactions as a post-design concern. In order to take human factors into consideration, the proposed approach uses abstractions that represent the effects of human performance characteristics on human-automation interactions. These abstractions are then integrated with formal models of the autonomous system under consideration, and reactive synthesis techniques are applied to generate autonomous control protocols given a specification for overall performance. To demonstrate this concept, we: 1) develop an example scenario based on a UAV road surveillance mission; 2) propose a hypothetical model for human operator behavior and performance that takes into account characteristics such as workload, proficiency, and fatigue; and 3) synthesize UAV control protocols and perform numerical experiments using tools for probabilistic model checking and two-player stochastic games. Based on the example scenario and hypothetical human model, experimental results suggest that this approach may be feasible for synthesizing control protocols for autonomous systems that account for uncertainties and imperfections in interactions with human operators.**

***Index Terms*—Control protocol synthesis, human factors, human–automation interaction, probabilistic models and specifications, unmanned aerial vehicles (UAVs).**

I. Introduction

AUTONOMOUS systems are becoming increasingly prevalent in today's society. For example, prototypes of driverless cars have been developed and tested on public roads, and unmanned aerial vehicles (UAVs) have been used for crop dusting and weather monitoring. Despite the name, however, autonomous systems usually do not act in isolation; rather, they often perform their intended functions at the behest of a human operator who acts as either a supervisor or a collaborator, depending on the onboard autonomy's designed purpose and level of sophistication. Numerous highly publicized incidents and accidents involving autonomous systems have demonstrated the dangers of failing to account for operator interactions in the system design. As an example, in the Global Hawk incident [2], a UAV unexpectedly accelerated to a ground speed of 155 knots (under the control of automated mission planning software) when the operator commanded it to taxi at a speed of 6 knots; a lack of proper coordination between the operator and automation caused the UAV to eventually run off the runway and crash. Given the potential for such incidents, regulators have recognized the importance of accounting for human factors when designing safety-critical autonomous systems. For instance, the National Highway Traffic Safety Administration advised that the amount of time and state of situational awareness needed by an operator to safely retake manual control of a car from an automated state should be considered in the design of driver–vehicle interfaces as well as in the development of operator training and certification

Manuscript received December 16, 2015; accepted February 01, 2016. Date of publication March 08, 2016; date of current version April 05, 2016. This paper was recommended for publication by Editor J. Wen upon evaluation of the reviewers' comments. The work was supported by the James S. McDonnell Foundation Postdoctoral Fellowship , ERC Advanced under Grant AdG-246967 VERIWARE, OSD , AFRL , AFOSR under Grant FA9440-12-1-0302, AFOSR under Grant 13RQ03COR, ONR under Grant N000141310778 and Grant N00014-15-IP-00052, and NSF under Grant 1446479.

L. Feng is with the Department of Computer and Information Science, University of Pennsylvania, Philadelphia, PA, 19104 USA (e-mail: lufeng@cis.upenn.edu).

C. Wiltsche is with the Department of Computer Science, University of Oxford, Oxford OX1 3QD, U.K. (e-mail: clemens.wiltsche@cs.ox.ac.uk).

L. Humphrey is with the Control Science Center of Excellence, Air Force Research Laboratory, Dayton, OH 45433 USA (e-mail: laura.humphrey@us.af.mil).

U. Topcu was with the Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, PA 19104 USA. He is now with the Department of Aerospace Engineering and Engineering Mechanics, The University of Texas at Austin, Austin, TX 78712-1221 USA (e-mail: utopcu@seas.upenn.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TASE.2016.2530623

requirements [3]. The Federal Aviation Administration raised similar considerations for UAVs: for the purpose of collision avoidance, human interfaces must display information about the state of the UAV and nearby air traffic, taking into account visual processing capabilities of the operator [4]. Indeed, vast amounts of data and literature from human factors research are becoming available, which can provide guidance for the design of autonomous systems (see Section II).

Reactive synthesis offers a promising paradigm of approaches to design correct-by-construction control protocols for autonomous systems. Given a model (e.g., transition system) and a property specification (often expressed in temporal logics) for an autonomous system, synthesis approaches can automatically generate a protocol (or strategy) for controlling the system that satisfies the property. Over the past decades, various reactive synthesis techniques have been developed for the design of different types of autonomous systems. A review of such techniques can be found in [5], while some recent advances in synthesis for probabilistic systems are presented in [6] and [7]. These techniques have been applied to real-world case studies such as UAV mission planning [8] and autonomous urban driving [6].

With respect to correct-by-construction design of autonomous systems that interact with human operators, a key challenge for reactive synthesis is obtaining appropriate models of human operator behavior and performance. Work in this area is still limited, and available models are not necessarily well suited to reactive synthesis. In this paper, we present a detailed case study focused on synthesizing human-in-the-loop control protocols for UAV mission planning, in which we develop a hypothetical model of operator behavior and performance amenable to reactive synthesis based on high-level trends induced from human factors literature. Specifically, we develop this model in the context of a simple scenario involving road network surveillance by a UAV. We first build abstractions for human-automation interactions in this scenario based on Markov decision processes (MDPs), a widely used model for discrete-time stochastic control processes. A (fully probabilistic) operator model is developed, taking into account a rich set of human performance characteristics (e.g., proficiency, workload, and fatigue). The human-automation interaction is then modeled as a product MDP from the composition of the operator model and a UAV model. Given a mission objective, we can then synthesize an optimal UAV piloting plan that satisfies it via finding a strategy in the MDP. If models for individual operators are available, we may even synthesize individualized optimal UAV piloting plans. Moreover, if there are multiple mission objectives, we can draw Pareto curves to help operators understand trade-offs. We can also demonstrate the impact of operator characteristics on UAV mission performance.

It may be beneficial to add nondeterminism in the operator model, e.g., for modeling human dynamic re-tasking of UAVs to address previously unforeseen circumstances. To distinguish the two different sources of nondeterminism due to the operator and the UAV, we augment the MDP models to stochastic two-player games. The goal is then to synthesize a winning strategy for the UAV (Player 1) against all strategies (including the worst-case) of the operator (Player 2). This separate role consideration is also useful in modeling design decisions about function allocation, (i.e., deciding which tasks should be assigned to the operator and which tasks should be assigned to the autonomy). As with MDPs, we can similarly synthesize individualized UAV strategies and analyze mission objective tradeoffs with games. In addition, we may guide refinement of the admissible operating region and provide informative feedback to operators for achieving better mission performance.

The remainder of this paper is organized as follows. First, we review related work in Section II. We then describe the motivating example in Section III and introduce formal specifications and models in Section IV. We present our modeling approach and experimental results for MDPs and stochastic two-player games in Sections V and VI, respectively. Finally, we critique our work and discuss directions for future research in Section VII.

II. Related Work

Human–automation interaction can generally be defined as the process of a human operator and an autonomous system working together to accomplish a goal [9]. The main focus of this area is then understanding and shaping interactions such that goals in a given domain can be achieved in a manner that is effective, efficient, and safe. In this paper, we will investigate how, given requirements for human–automation interaction, a correct-by-construction protocol for the automation's behavior can be generated using reactive synthesis approaches, assuming a model of the human operator's behavior and performance is available. Other approaches for analyzing human–automation interaction and guiding design of the automation have also been proposed, though developing appropriate models of human behavior and performance remains a challenge.

In fact, a great deal of past research in human factors has focused on quantifying how human behavior and performance on different tasks is affected by a variety of human characteristics, including long-term attributes such as age, experience, task proficiency, spatial reasoning ability, and attentional control [10], as well as short-term attributes such as stress, workload, proficiency, and fatigue. For instance, data from [11] demonstrate that, on a wide variety of tasks, human error often increases with higher levels of stress and decreases with higher levels of proficiency; moreover, for vigilance tasks that require detecting simple infrequent signals over prolonged periods of time without rest, missed detections tend to increase over time. Similar trends can be found in other studies. For instance, one study of vigilance tasks found declining response rates after as little as 3 min of task performance, with response rates eventually plateauing at 70%–80% of initial rates [12]. Differences in task performance can also vary between operators. For instance, a meta-analysis of 53 studies concluded that introverts have better overall performance than extraverts on visual detection tasks [13]. Operator performance on visual identification and classification tasks can also vary significantly in response times and accuracy, e.g., due to differences in age or experience [14].

Recent work has made progress in developing methods to guide analysis and design of human-automation systems, drawing inspiration from such results. For instance, consider supervisory control, in which a single human operator oversees and intermittently interacts with multiple autonomous systems. An important factor to consider in this domain is operator workload. At a high level, one question is how many autonomous systems a single operator could hypothetically supervise at maximum workload. In [15], a measure called fan-out is proposed to predict an upper bound on this number based on neglect time, i.e., the amount of time a single system can be neglected before it needs operator attention, and interaction time, i.e., the amount of time required for the operator to interact with the system until it no longer requires attention. This measure was later modified in [16] to include wait time, i.e., the amount of time it takes the operator to respond to a request for interaction from the system. Assuming that the number of autonomous systems falls below the operator's upper bound and there is some flexibility in the timing of the supervisory tasks, it is then possible to try to balance or schedule human-automation interactions to "optimize" workload in accordance with the Yerkes–Dodson law, which states that human performance increases with mental arousal up to a point, then decreases thereafter [17]. For instance, in [18], a queuing theory-based discrete event simulation is used to predict measures of mission performance based on operator attention allocation strategies, interaction times, and workload, the level of autonomy and collaboration of the supervised systems, and environmental events that trigger human–automation interactions. The authors note that this type of model could be used to determine, e.g., what level of autonomy is needed for a given number of systems to maintain an optimal workload of $\sim 70\%$, as measured through utilization time or "percent busy time" of the operator. Related research models the human operator as a dynamical queue such that operator workload depends on recent utilization history [19]; such models can then be used to derive a task scheduling policy that an automated task manager could employ to maximize operator task throughput. Similarly, [20] extends the concepts in [15] and [16] to develop a model that predicts at runtime whether an operator will interact with an autonomous system before it reaches a critical state based on time remaining and the operator's current allocation of attention, as assessed by eye fixations; this allows the automation to determine when it should issue an alert or take a corrective action that does not require operator interaction.

Whereas the aforementioned models and methods are geared mostly towards questions related to supervisory control, more general models of human behavior and performance have been developed using cognitive architectures. These types of architectures attempt to codify aspects of cognitive agents that are constant across different application domains, including: 1) short-term and long-term memories that store content about beliefs, goals, and knowledge; 2) representation of elements that are contained in these memories and their organization into larger-scale mental structures; and 3) functional processes that operate on these structures, including performance mechanisms that utilize them and learning mechanisms that alter them [21]. As an example, one such model has been developed that predicts the effects of sleep deprivation, sleep restriction, and circadian rhythms on performance in sustained attention tasks by making the functional processes that activate in response to the task stimulus dependent on these factors [22]. Another such model predicts whether a human storyteller who is interrupted in the middle of repeating a memorized story would benefit from an automated suggestion on where to resume the story by encoding previously retold portions of the story as memory elements whose activation levels decay over time and include noise [23]. Models developed in cognitive architectures have also been used to develop "intelligent tutors" that can offer suggestions to human students when they encounter problems by comparing a student's actions to procedural knowledge that encodes possible solution steps as well as typical student missteps [24].

There has also been interest in using formal methods such as model checking and theorem proving for verification of systems that involve human-automation interaction [25]. In this domain, formal models of the software interface the human uses to interact with the automation have been used to represent potential human behavior, which can then be used to check certain types of human-automation interaction properties. For instance, [26] describes visibility properties, e.g., whether events have appropriate user feedback; reachability properties, e.g., whether effects of actions can be undone; and reliability properties, e.g., questions relating to the safety and consistency of automation and interface behaviors. Going beyond the interface, other research in this area has attempted to develop state machine representations of user mental models, i.e., user beliefs about how the automation behaves. Model checking approaches can then be used to search for deviations between these mental models and actual models of automation behavior, e.g., to identify mismatches between the two that can lead to potentially dangerous automation surprises [27]. To date, mental models have been developed based on user interviews, user questionnaires, observations of user behavior, and reviews of system training materials [28], [29]; additionally, [25] notes that human factors methods such as cognitive task analysis [30] could also be used to derive such models.

More recent research has focused on synthesizing designs for human–automation systems, rather than manually developing then verifying them. For instance, [31] considers reactive synthesis of an "auto-controller" for a semi-autonomous car that attempts to meet a set of temporal logic specifications but can fall back on a human driver if it determines it soon may no longer be able to do so. The auto-controller then drives the car in the nominal case, and another synthesized "advisory controller" indicates to the human when he or she must take control of the car. Synthesis of the auto-controller and advisory controller take into account certain human factors concerns, including the amount of time needed for the human driver to respond to an alert from the advisory controller. The approach is similar in concept to one proposed here, though our approach relies on more detailed human models and synthesizes automated controllers that perform joint tasks with human operators rather than controllers that switch between purely human and purely autonomous control.

In this paper, we leverage guidance from this relatively broad body of literature to develop hypothetical models of human behavior and performance that capture the effects of relevant human characteristics, including proficiency, workload, and fatigue. Using such models, we demonstrate how formal spec-

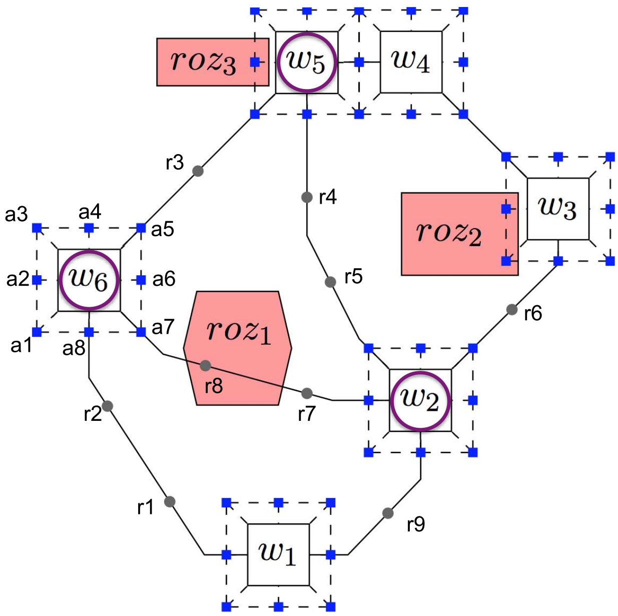

Fig. 1. Road network for UAV ISR missions (adapted from [8]).

ification and synthesis techniques can facilitate the design of control protocols for systems that interact with human operators. We emphasize that ongoing advances in human factors and quantitative modeling will enhance the applicability of the methods we propose in this paper.

III. Motivating Example

As an illustration of synthesis for autonomous systems that interact with human operators, we describe two variants of an example in which a remotely controlled UAV is used to perform intelligence, surveillance, and reconnaissance (ISR) missions over a road network. Fig. 1 shows a map of the road network, which has six surveillance waypoints labeled $w_1, w_2, \ldots, w_6$. Approaching a waypoint from certain angles may be better than others, e.g., in order to obtain desired look angles on a waypoint target using an ellipsoidal loiter pattern. Angles of approach are thus discretized in increments of 45° around each waypoint, resulting in eight angle points $a_1, a_2, \ldots, a_8$ around each waypoint. Roads connecting waypoints are discretized into road points $r_1, r_2, \ldots, r_9$. Red polygons represent "restricted operating zones" (ROZs), areas in which flying the UAV may be dangerous or lead to a higher chance of being detected by an adversary.

In current practice [32], at least two human operators are required for a UAV ISR mission: one to pilot the UAV, and the other to steer the onboard sensor and interpret the sensor imagery. Here, we assume the UAV has a certain degree of autonomy that is used to fulfill most of the piloting functions, e.g., maintaining loiter patterns around waypoints, selecting most of the points that comprise the route, and flying the route. The human operator primarily performs sensor tasks, e.g., steering the onboard sensor to capture imagery of targets at waypoints. However, the operator also retains the ability to affect some of the piloting functions of the UAV. In both variants of the scenario, the operator decides how many loiters to perform at each waypoint, since more loiters may be needed if the operator is not satisfied with the sensor imagery obtained on previous loiters. Additionally, waypoints w_2, w_5, and w_6 in Fig. 1 will be designated as checkpoints. At checkpoints, the operator can directly impact the choices made by the protocol we synthesize by selecting different roads to be taken between waypoints. In the second variant of this example, the operator may also specify the angle of approach to surveillance waypoints.

In both cases, the optimal piloting plan for the UAV varies depending on mission objectives. Specification patterns for a variety of UAV missions are presented in [8], including safety, reachability, coverage, and sequencing of waypoints. We consider a few concrete examples here, e.g., surveillance of the road network with minimum fuel consumption, or flying to certain waypoints while trying to avoid ROZs. Our goal is to synthesize the optimal UAV piloting plan for a specific mission objective, which would be implemented by the UAV's onboard automation interface to control the route. In particular, we would like to investigate how the uncertainties and imperfections of a human operator's behavior affect the optimal UAV piloting plan. Specifically, what is the influence of an operator's proficiency, workload, and fatigue level on UAV mission performance? Can we synthesize individualized optimal UAV piloting plans for different operators? Can the automation provide informative feedback to operators to assist them in decision-making?

In the following, we introduce formal specifications and models to study the above questions. We discuss two different models, one based on MDPs and the other on stochastic two-player games, in order to abstract different types (e.g., probabilistic versus worst-case) of prior knowledge about imperfections in the operator's behavior and interactions (e.g., the operator's interference in the choices made by the autonomy protocol) between the autonomy protocols and the operator.

IV. Preliminaries

We use $\mathbb{Q}$ and $\mathbb{R}$ to denote the rationals and reals, respectively. A discrete probability distribution over a (countable) set Q is a function $\mu : Q \to [0,1]$ such that $\sum_{q \in Q} \mu(q) = 1$. Let $Dist(Q)$ denote the set of distributions over Q, η_q denote the point distribution on $q \in Q$, and $\mu_1 \times \mu_2$ denote the product of distributions μ_1 and μ_2.

A. Markov Decision Processes

Markov decision processes (MDPs) [33] are widely used for modeling discrete-time stochastic control processes whose outcomes are partly random and partly decided by a controller. Formally, an MDP is a tuple $M = \langle S, \overline{s}, \mathcal{A}, \delta \rangle$ where S is a countable set of states, $\overline{s} \in S$ is an initial state, $\mathcal{A}$ is a set of actions, and $\delta \subseteq S \times \mathcal{A} \times Dist(S)$ is a transition relation that assigns at most one distribution per state and action. Let $A(s) \stackrel{\text{def}}{=} \{a \in \mathcal{A} \,|\, \exists \mu \in Dist(S)\,.\,(s,a,\mu) \in \delta\}$ be the set of enabled actions at state s; for any state $s \in S$, $A(s) \neq \emptyset$. A state s has a successor s', written $s \xrightarrow{a} s'$, if there is an action $a \in A(s)$ such that $(s,a,\mu) \in \delta$ and $\mu(s') > 0$. We sometimes write $\delta(s,a)(s')$ to denote $\mu(s')$. If there are multiple actions in $A(s)$, a nondeterministic choice needs to be made. We call an MDP a discrete-time Markov chain (DTMC) if $A(s)$ is a singleton set for any state $s \in S$.

An infinite path through an MDP M is a sequence of alternating states and actions $s_0 \xrightarrow{a_0} s_1 \xrightarrow{a_1} \cdots$. A finite path ρ is a prefix of an infinite path ending in a state, denoted $\mathsf{last}(\rho)$. Let $IPath_s$ and $FPath_s$ denote the set of infinite and finite paths starting from state s of M, respectively. Let $IPath \stackrel{\text{def}}{=} \bigcup_{s\in S} IPath_s$ and $FPath \stackrel{\text{def}}{=} \bigcup_{s\in S} FPath_s$. We use a strategy to resolve the nondeterministic choices in an MDP, which is defined formally as a function $\sigma : FPath \rightarrow Dist(\mathcal{A})$ such that $\sigma(\rho)(a) > 0$ only if $a \in A(\mathsf{last}(\rho))$. Under a particular strategy σ, the behavior of an MDP M is fully probabilistic, inducing a DTMC M^σ (implicitly starting at $\overline{s}$).

We can define a probability space over infinite paths $IPath$ of a DTMC, called the path distribution, in the standard way. For each finite path $\rho \in FPath$, the cylinder C_ρ is the set of all infinite paths with prefix ρ. Given a finite path $\rho = s_0 \xrightarrow{a_0} s_1 \xrightarrow{a_1} \cdots \xrightarrow{a_{n-1}} s_n$, the probability measure of its cylinder is defined as $\mathbf{P}(\rho) \stackrel{\text{def}}{=} \prod_{i=0}^{n-1} \delta(s_i,a_i)(s_{i+1})$. This measure uniquely extends to infinite paths due to Carathéodory's extension theorem. Given an MDP M and a strategy σ, we denote by Pr_s^σ the resulting probability measure over all infinite paths (starting at state s) of the induced DTMC M^σ.

We can also reason about the rewards (sometimes called costs) of certain events happening in an MDP. We define a reward structure as a function $r : (S \times \mathcal{A}) \rightarrow \mathbb{Q}$, and the total reward of an infinite path $\rho = s_0 \xrightarrow{a_0} s_1 \xrightarrow{a_1} \cdots$ as $\mathsf{rew}(r)(\rho) \stackrel{\text{def}}{=} \lim_{N\rightarrow\infty} \sum_{i=0}^{N} r(s_i,a_i)$, if the limit exists. The expected total reward computes the expectation of rewards accumulated along all the infinite paths $IPath_{\overline{s}}$ starting at the initial state $\overline{s}$ in an induced DTMC M^σ, denoted by $\mathbb{E}^\sigma[\mathsf{rew}(r)] \stackrel{\text{def}}{=} \int_{\rho\in IPath_{\overline{s}}} \mathsf{rew}(r)(\rho) dPr_{\overline{s}}(\rho)$. For example, we can define a reward/cost structure based on how much fuel a UAV uses during each (discrete) "fly" step and compute the expected total fuel consumption of a mission.

A specification is a predicate on path distributions. For example, $\mathbb{E}^\sigma[\mathsf{rew}(r)] \leq v$ is a specification requiring the expected total reward/cost of r, under an MDP strategy σ, to be smaller than the *bound* v. We say that φ is achievable in an MDP M if there is a strategy σ such that φ holds for the induced DTMC M^σ, written $M^\sigma \models \varphi$. A conjunctive query (CQ) is a specification defined via the conjunction of multiple objectives. For example, a CQ

$$\mathbb{E}^\sigma[\mathsf{rew}(r_1)] \leq v_1 \wedge \mathbb{E}^\sigma[\mathsf{rew}(r_2)] \leq v_2$$

asks us to find a strategy σ that minimizes both rewards/costs to achieve bounds v_1 and v_2 simultaneously. Note that the quantification of strategies is over the entire CQ, i.e., it is not sufficient to find one strategy for each objective in isolation. Moreover, a CQ does not have a single achievable optimum; rather, it has several so-called Pareto optima. The intuition is that a Pareto optimum for several objectives cannot be improved in one dimension without degrading another dimension. For example, if $(v_1, v_2) = (4, 6)$ is a Pareto optimum for the above CQ, then $(4-\varepsilon, 6)$ and $(4, 6-\varepsilon)$ are not achievable for any $\varepsilon > 0$. The set of all Pareto optima is called the Pareto curve.

Given an MDP M and a specification φ, the synthesis problem aims to find a strategy σ in M such that $M^\sigma \models \varphi$. For specifications consisting of a single objective, an optimal strategy is one that achieves either the minimum or maximum value depending on the objective. For CQs about multiple objectives, e.g., for simultaneously minimizing several expected total rewards/costs, we speak of (Pareto) optimal strategies if they achieve a point above the Pareto curve. The method of finding optimal strategies can vary for different types of specifications. For example, an optimal strategy is obtained by choosing the locally optimal action in each state for minimum probability reachability in MDPs. Strategy synthesis for CQs reduces to a linear programming problem and can be solved in polynomial time. We refer to [7] for more details.

We often model parts of the system (e.g., the operator and the UAV) separately, and obtain a model for the entire system through composition. Formally, the composition of two MDPs M_1 and M_2 yields another MDP $M_1 \| M_2 = \langle S_1 \times S_2, (\overline{s}_1, \overline{s}_2), \mathcal{A}_1 \times \mathcal{A}_2, \delta \rangle$ where the transition relation δ is defined such that $((s_1,s_2), a, \mu_1 \times \mu_2) \in \delta$ iff one of the following holds: 1) $(s_1, a, \mu_1) \in \delta_1$, $(s_2, a, \mu_2) \in \delta_2$ and $a \in \mathcal{A}_1 \cap \mathcal{A}_2$; 2) $(s_1, a, \mu_1) \in \delta_1$, $\mu_2 = \eta_{s_2}$ and $a \in \mathcal{A}_1 \backslash \mathcal{A}_2$; and 3) $(s_2, a, \mu_2) \in \delta_2$, $\mu_1 = \eta_{s_1}$ and $a \in \mathcal{A}_2 \backslash \mathcal{A}_1$, that is, two MDPs are composed by synchronizing on common actions and interleaving otherwise.

Models such as the ones we use in this paper represent an abstraction of the scenario in which we are interested, and different types of models allow abstractions to various levels and types of detail. By modeling with MDPs, we choose an abstraction where all nondeterminism in the model is resolved by either the operator or the UAV via a strategy. Since we are interested in synthesizing UAV piloting plans (represented by strategies), we model that all nondeterminism is resolved by the UAV strategy, and this strategy should not influence the operator's choices. Hence, in the MDP modeling, we assign probabilities to the choices the operator can make in order to avoid nondeterminism wrongly resolved by the UAV, assuming the availability of concrete probability distributions constraining the possible operator behaviors.

B. Stochastic Two-Player Games

The previously mentioned restriction of MDPs (i.e., all nondeterminism is controlled by one strategy) can be relaxed by modeling the scenario as a game between two players (i.e., the operator and the UAV), in which each player has a separate strategy to control different actions. Synthesizing an optimal piloting plan requires finding a UAV strategy that achieves the mission specification under any possible operator strategy, with the interpretation that we do not want to constrain the operator's behavior more than necessary.

We hence augment MDPs to stochastic two-player games (or games) [34] by distinguishing two types of nondeterminism, each controlled by a player. We use $\mathsf{Player\ 1}$ to represent the controllable part of a system for which we want to synthesize a strategy and use $\mathsf{Player\ 2}$ to represent the uncontrollable environment. Formally, a game is defined as a tuple $G = \langle S, (S_1, S_2), \overline{s}, \mathcal{A}, \delta \rangle$ where the set of states S is partitioned into $\mathsf{Player\ 1}$ states S_1 and $\mathsf{Player\ 2}$ states S_2, while the initial state $\overline{s}$, the action set $\mathcal{A}$, and the translation relation δ are defined exactly as they are for MDPs. Indeed, an MDP can be considered as a special case of games, where $S = S_1$ and $S_2 = \emptyset$. A Player i strategy for $i \in \{1,2\}$ is a function $\sigma_i : FPath_i \rightarrow Dist(\mathcal{A})$ where $FPath_i$ is the set of finite paths ending in a Player i state. By fixing a strategy for one player, a game becomes an MDP, while applying a pair of strategies (σ_1, σ_2) for both players to a game G yields a DTMC, denoted by G^{σ_1,σ_2}. Thus, formalisms such as probability measures and rewards can be defined for games in a similar fashion as for MDPs. A Player 1 strategy σ_1 wins a game G for a specification φ if $G^{\sigma_1,\sigma_2} \models \varphi$ for any Player 2 strategy σ_2. The synthesis problem seeks to construct such a winning strategy. For single-objective specifications, the problem can be solved via a value iteration method [7]. For multi-objective synthesis, we compute successive under-approximations of the Pareto curves for Player 1 at each state, from which we construct winning strategies [6].

C. Tools and Implementations

We build concrete scenario models and perform experiments using the following tools. We use PRISM [35] for the modeling and synthesis of MDPs, use its extension PRISM-games [36] for strategy synthesis in stochastic games with single objectives, and use the implementation of [6] for multi-objective synthesis in games.

V. MDPs for Human–Automation Interactions

Consider the first variant of the scenario described in Section III. Recall that at each waypoint, the operator steers the sensors and decides if the UAV needs to continue loitering based on the quality of the captured sensor imagery, which is under the influence of human performance characteristics such as proficiency, workload, and fatigue. At checkpoints (i.e., w_2, w_5, and w_6), the operator selects the next road point for the UAV. Particular operator choices are unknown at design time. If there exists sufficient prior information on possible patterns in operator choices, relevant statistics may be obtained (e.g., from training logs), and a probabilistic model such as an MDP may be built as an appropriate abstraction. In this section, we consider such models and associated synthesis problems.

A. Modeling

1) Operator Model: As discussed in Section IV-A, we build abstractions of the operator's possible behavior as a fully probabilistic model M_{OP}. Fig. 2 shows a fragment of the model, representing the possible behavior at waypoint w_6. There is a non-negative integer variable k counting the number of sensor tasks performed by the operator since the beginning of the mission. The updates "k++" represent increasing the value of k by one. The purpose of using k in the model is to measure the operator's fatigue level. To obtain a finite state model, let the value of k stop increasing once it reaches a certain threshold T (a constant that will be used later in modeling fatigue).

In general, operators' workload levels are driven by a number of factors including mission characteristics, e.g., how many UAVs the operator supervises simultaneously and the phase of the mission. For simplicity and to reduce the complexity of the models (so that the results discussed later are easier to interpret), we model the operator's workload as a uniform

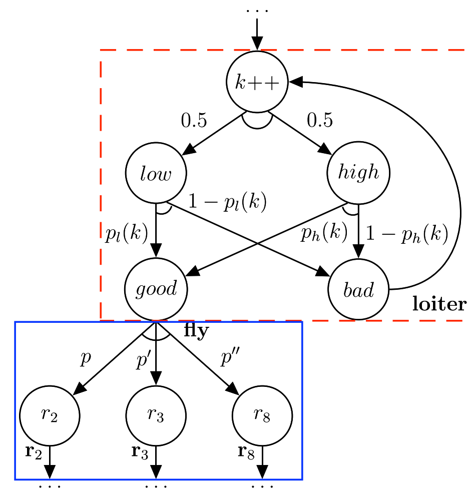

Fig. 2. Fragment of the operator model M_{OP} representing the operator's behavior. The red dashed square highlights the common behavior that is repeated at all waypoints, while the blue solid square indicates specific choices of roads at waypoint w_6. The operator's behavior at other waypoints is omitted from the figure, indicated by $\cdots$.

distribution over two levels: low and high. Operators' accuracy on vigilance tasks tends to decline with lower levels of proficiency and higher levels of workload [11]. Moreover, one study [12] finds that operators' performance gets discounted after a certain period of time, due to fatigue. Based on these facts, we model an operator's accuracy in steering sensors to capture high resolution imagery of targets as probability distributions that correlate with proficiency, workload, and fatigue. Specifically, when the operator's workload level is low, the probabilities of capturing good and bad quality imagery are $p_l(k)$ and $1-p_l(k)$, respectively. Here $p_l(k)$ is a function over the variable k such that $p_l(k) = p_l(0)$ if $k < T$ and $p_l(k) = f \cdot p_l(0)$ if $k \geq T$, where $p_l(0)$ is the initial parameter value of the accuracy function, T is the fatigue threshold mentioned earlier, and f is a fatigue discount factor. The intuition is that, due to fatigue, the operator's accuracy gets discounted after performing a certain number of tasks. We define the accuracy function $p_h(k)$ for high workload analogously. Note that $p_l(k) \geq p_h(k)$ for any k, modeling the fact that an operator tends to make more errors under higher levels of workload and stress. Furthermore, more proficient operators have higher values for the accuracy parameters $p_l(0)$ and $p_h(0)$. If the quality of the captured imagery is bad, the operator would ask the UAV to continue to loiter at the current waypoint in order to collect more sensor imagery; otherwise, the operator allows the UAV to fly to another waypoint. At each waypoint, the operator repeats the aforementioned behavior (the red dashed square in Fig. 2).

The operator selects the next road for the UAV at waypoints that are checkpoints (w_6 for example), while the UAV controller chooses the road at any non-checkpoint waypoint. As illustrated in the blue solid square in Fig. 2, we model the operator's choices at w_6 as following a certain probability distribution, i.e., picking the roads that connect to neighboring road points r_2, r_3, and r_8 with probabilities p, p', and p'', respectively (note that $p + p' + p'' = 1$). Such probability distributions may be obtained using data-driven approaches. For example, we may

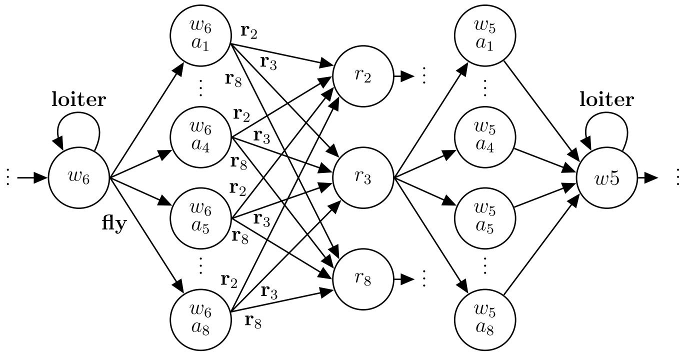

Fig. 3. Fragment of the UAV model M_{UAV}, representing the UAV loitering and flying over the waypoints w_6 and w_5.

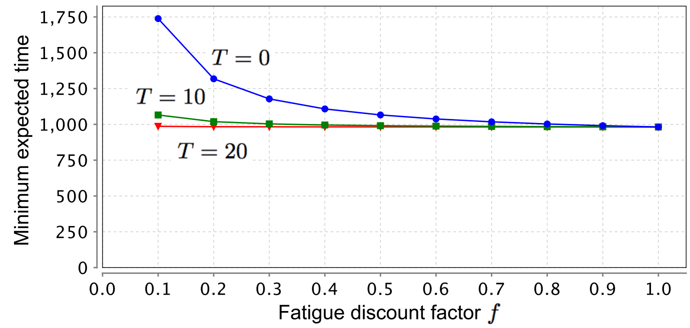

Fig. 4. Effect of operator fatigue on minimum expected mission completion time, for different values of the fatigue threshold T and discount factor f (with fixed parameters $p_t(0) = 0.9$ and $p_h(0) = 0.8$).

extract operator behavior patterns for generic classes of checkpoints using human factors research methods such as cognitive task analysis [30] and maintain a library of operator-dependent behavior patterns. We can then instantiate the probability distribution for a specific checkpoint in a map by matching operator behavior library statistics, e.g., at "three-way crossing with a bridge"-type checkpoints, the operator chooses the bridge with probability 0.5.

Suppose the operator chooses r_3. According to the map shown in Fig. 1, the next waypoint is w_5. For simplicity, the operator's behavior at w_5 is omitted from Fig. 2. If we were to illustrate the behavior, a duplicate set of states as highlighted in the red square would be drawn to connect with r_3 in Fig. 2.[1]

2) UAV Model: We model the UAV's piloting behavior as an MDP M_{UAV}, which contains 63 states (six waypoints, six $\times$ 8 angle points, and nine road points). At any waypoint or road point, the UAV can nondeterministically fly to a neighboring angle point or road point. These nondeterministic choices need to be resolved by a strategy. Fig. 3 shows a fragment of the UAV model[2], illustrating how the UAV loiters and flies over waypoints w_6 and w_5. If the UAV receives a loiter instruction from the operator, it loiters at the current waypoint, allowing the operator to capture more sensor imagery; otherwise, the UAV randomly picks one of the eight angle points $a_1, \ldots, a_8$ to exit w_6. Then, a nondeterministic choice between three roads r_2, r_3, and r_8 needs to be resolved. Suppose r_3 is chosen by the operator; then the UAV can fly to the waypoint w_5 and approach it via one of the eight angles, or the UAV can also fly back to the waypoint w_6 (for clarity, this choice is not drawn in Fig. 3).

3) Operator–UAV Interactions: We model interactions between the operator and the UAV by composing M_{OP} and M_{UAV}, which synchronize over common actions *loiter* and *fly*, and obtain a product MDP $M_{\mathsf{OP}} \| M_{\mathsf{UAV}}$. Synchronization between actions in our models abstracts the concrete process of interchanging information between the operator and the UAV, which is assumed via a reliable communication protocol. The model does not distinguish between "sender" and "receiver," or outputs and inputs, of a protocol. We can think of the operator initiating the loiter action, which the UAV can receive at any waypoint (see the self-loops at w_6 and w_5 in Fig. 3). The fly action is initiated as well by the operator, but we can think of the UAV deciding which flight direction to take (see the fly transitions at w_6 in Fig. 3). Note that synchronization between actions in the model also assumes that the operator and UAV synchronize their behaviors temporally.

Since M_{OP} is a DTMC and has no nondeterminism, synthesizing a strategy σ for the MDP $M_{\mathsf{OP}} \| M_{\mathsf{UAV}}$ yields a strategy σ' for M_{UAV} such that $(M_{\mathsf{OP}} \| M_{\mathsf{UAV}})^{\sigma} = M_{\mathsf{OP}} \| M_{\mathsf{UAV}}^{\sigma'}$. The strategy σ' operates on the MDP model M_{UAV} of the UAV. For σ' to be implemented as a flight controller for the UAV, it must know which state of M_{UAV} the UAV is in, and hence, from the point of view of the strategy, the model transitions are triggered by the UAV's behavior. Note that the probability distributions in the MDP model are not sampled at runtime, but the respective transitions are executed to correspond to the actual behavior of the UAV. In contrast, if the strategy σ' is randomized, the choice of the strategy has to be sampled from the distribution $\sigma'(\lambda)$ at runtime.

B. Analysis

We present experimental results using the above MDP model. In particular, we consider the following questions proposed in Section III: Does an operator's fatigue, proficiency, and workload level affect UAV mission performance? Can we synthesize individualized optimal UAV piloting plans for different operators? Can we provide informative feedback to operators to assist decision-making? Since we do not have access to a real operator's behavior data, our experiments are based on some fictional numbers. These parameter choices, however, are still consistent with the general trends shown in a large body of literature on human factors research as discussed earlier. Our primary goal is simply to demonstrate the capabilities of our approach.

1) Effects of Operator Fatigue: Consider a UAV surveillance mission that requires covering all six waypoints in Fig. 1, where the objective is to complete the mission as fast as possible. Assume that each loiter takes ten time units and flying between any neighboring waypoint and/or road point takes 60 time units. Fig. 4 illustrates the influence of the operator's fatigue threshold T and discount factor f on the minimum expected time to complete the mission. The general trend is that the UAV completes the mission faster if the operator has a higher fatigue threshold

[1][Online]. Available: http://www.prismmodelchecker.org/casestudies/human-uav.php for the complete model states and transitions

[2]Our models are shown with several distributions associated with an action name but after composition this can easily be resolved through renaming, and we obtain MDPs.

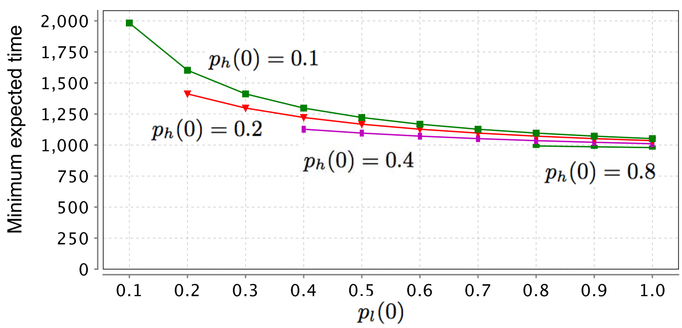

Fig. 5. Effect of operator proficiency and workload on minimum expected mission completion time, for different initial values of the accuracy functions $p_l(0)$ and $p_h(0)$ (with fixed parameters $T = 10$ and $f = 0.7$).

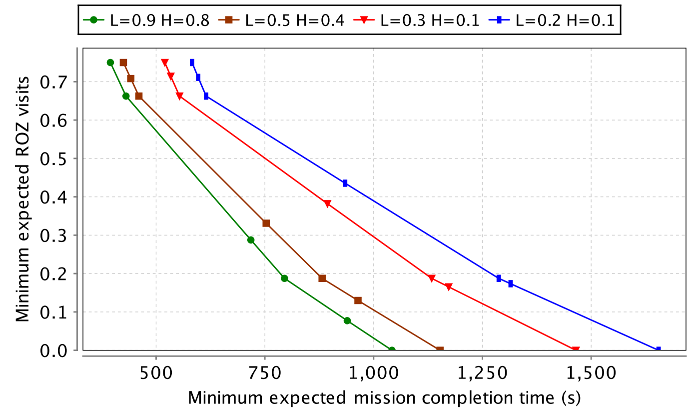

Fig. 6. Pareto curves of two mission objectives for various accuracy function parameters (L is $p_l(0)$ and H is $p_h(0)$), with fixed fatigue parameters $T = 10$ and $f = 0.7$.

T (i.e., less likely to get tired) or a larger value of f (i.e., the accuracy is less discounted). The best UAV performance (i.e., the smallest expected mission completion time) is achieved when $f = 1$, that is, there is no accuracy discount due to fatigue.

2) Effects of Operator Proficiency and Workload: The operator's accuracy in steering sensors and capturing good quality imagery are affected by proficiency and workload. Fig. 5 illustrates the influence of accuracy parameters $p_l(0)$ and $p_h(0)$ on the minimum expected time of finishing the mission (i.e., covering all six waypoints). The trends show that a more proficient operator who has higher values for $p_l(0)$ and $p_h(0)$ can complete the mission faster. In addition, the more accuracy declines due to high workload, i.e., the larger the gap between $p_l(0)$ and $p_h(0)$, the longer the time needed to complete the mission.

3) Synthesizing Individualized Strategies: Suppose it is possible to parameterize the model for individual operators, e.g., using an operator-dependent behavior patterns library as described in Section V-A. Then, we can synthesize individualized optimal UAV piloting plans for different operators. For example, consider the mission of covering waypoints w_1, w_2, and w_6 in Fig. 1, with the UAV starting at w_1. The objective is to minimize fuel consumption, where flying from a waypoint or road point to a neighboring point costs the UAV one unit of fuel (assume fuel consumption for loitering is negligible). The operator's choices at checkpoints w_2 and w_6 are instantiated with probability distributions from the (hypothetical) operator behavior patterns library as discussed earlier. Suppose there is a risk adverse operator who exhibits the behavior pattern of always avoiding ROZs, and the operator's choices are instantiated as picking (with probability 1) r_5 at w_2 and r_2 at w_6. The optimal UAV piloting plan fulfills the mission objective with the following path:

$$w_1 \rightarrow r_1 \rightarrow r_2 \rightarrow w_6 \rightarrow r_2 \rightarrow r_1 \rightarrow w_1 \rightarrow r_9 \rightarrow w_2$$

costing eight units of fuel. Suppose there is another operator whose behavior pattern library shows a higher likelihood of entering ROZs (e.g., due to risk-taking tendencies or not being able to recognize the danger when operating under high workload), and the choices are instantiated as picking r_7 at w_2 and r_3 at w_6. The synthesized optimal UAV piloting plan yields the path

$$w_1 \rightarrow r_9 \rightarrow w_2 \rightarrow r_7 \rightarrow r_8(roz_1) \rightarrow w_6$$

which only requires five units of fuel but flies through the ROZ. Such individualized results generalize to truly probabilistic operator models. In particular, the individualization would be more obvious if we enhanced the coupling of the operator's choice distributions and other operator characteristics such as fatigue and workload.

4) Tradeoffs Between Mission Objectives: Interviews with UAV domain experts [37] show there is a need for operators to understand the risk associated with flying in certain conditions and the priority of mission objectives (e.g., get the target versus risk UAV survival). When there are multiple potentially conflicting mission objectives, conjunctive queries for MDPs (see Section IV-A) can help to investigate the tradeoffs. Suppose that the UAV mission is to cover waypoints w_1, w_2, and w_6. There are two objectives: 1) minimizing the expected mission completion time, and 2) minimizing the risk of being detected by an adversary (measured by the expected total number of ROZ visits). Fig. 6 shows the Pareto curves of these two mission objectives for different initial values of the operator's accuracy functions $p_l(0)$ and $p_h(0)$. For a specific Pareto curve, any point in the area (upward closure) above the curve (called a Pareto set) represents a pair of mission objective values that is achievable by a UAV piloting strategy. For example, when $p_l(0) = 0.9$ and $p_h(0) = 0.8$, the minimum expected time of completing the mission is about 400 time units while the expected number of ROZ visits is 0.75; on the contrary, the likelihood of ROZ visits can be reduced to 0 if the mission completion time is allowed to go up to at least 1040 time units. Pareto curves provide a useful visualization of tradeoffs between different mission objectives and can help the operator prioritize objectives. Once the operator selects a combination of mission objective values from the Pareto set, a corresponding optimal UAV piloting plan can be automatically synthesized.

Different operators have different Pareto curves. Based on an operator's characteristics (e.g., how quickly one gets tired, the tendency of making errors under a high workload), we can predict the mission performance by drawing and comparing Pareto curves. For example, in Fig. 6, the curves shift towards the right when the initial values of the accuracy functions $p_l(0)$ and $p_h(0)$ decrease, representing an increase in the mission completion time. Fig. 7 shows the shift of Pareto curves when varying

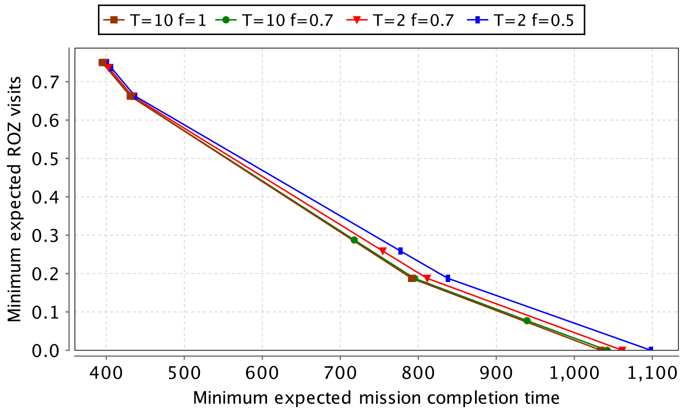

Fig. 7. Pareto curves of two mission objectives for various values of the operator's fatigue threshold T and discount factor f, with fixed accuracy function parameters $p_l(0) = 0.9$ and $p_h(0) = 0.8$.

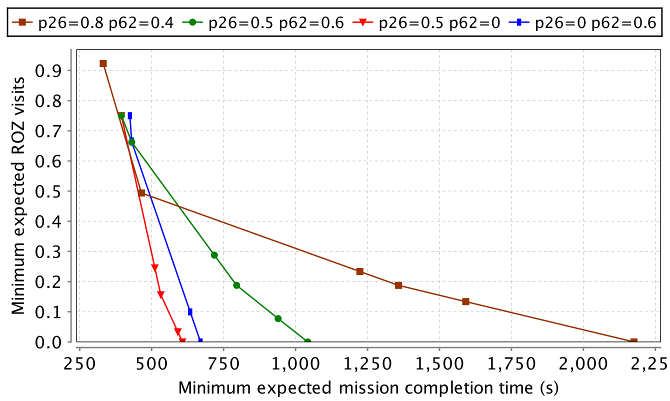

Fig. 8. Pareto curves of two mission objectives for various probability distributions of operator choices at checkpoints (e.g., p_{ij} is the probability of choosing the direct route from waypoint w_i to w_j), with fixed parameters $p_l(0) = 0.9$, $p_h(0) = 0.8$, $T = 10$, and $f = 0.7$.

values of the operator's fatigue threshold T and discount factor f, which is not as dramatic as in Fig. 6, indicating that the influence of fatigue parameters T and f may be less significant than that of the accuracy function parameters $p_l(0)$ and $p_h(0)$. Fig. 8 illustrates various Pareto curves for operators with different probability distributions of route choices at the checkpoints. The green lines (with dots) across Figs. 6–8 are drawn with the same set of model parameters, serving as a baseline for comparison. Fig. 8 shows the most dramatic (and irregular) shift of Pareto curves: it seems that, the more likely it is that an operator chooses a route that visits ROZs (e.g., the route between the waypoints w_2 and w_6 overlapping with roz_1), the more the Pareto curve shifts towards the upper-right (i.e., larger chances of entering ROZs, and more compromises of the mission completion time are needed in order to mitigate the likelihood of ROZ visits).

VI. Stochastic Games for Human-Automation Interactions

It may be necessary to model the operator's choices with nondeterminism, instead of fixing all probability distributions as in the previous MDP models. In the following, we model the operator's choices of roads at any checkpoint and of angles at any waypoint nondeterministically. Since there are now two different sources of nondeterminism (i.e., from the operator and the UAV), we augment the MDP models to stochastic two-player games. We are interested in synthesizing an optimal piloting plan for the UAV, to react to any choice of the operator. In other words, we are interested in finding an admissible strategy for the UAV against all (potentially non-cooperative) strategies of the operator. While we do not assume that the operator is adversarial, this competitiveness interpretation is appropriate when we are not able, or willing, to place restrictions on the operator.

A. Modeling

1) Delegation of Choices at Checkpoints: In the MDP models, the operator picks a road at each checkpoint for the UAV with some predefined probability distribution. In other words, synthesis for the UAV assumes that the operator's choices follow this distribution. However, it may be difficult to obtain and enforce such a distribution. Moreover, the operator may occasionally intervene to dynamically re-task the UAV to address unforeseen and unmodeled situations, so we should put less constraints on the operator's choices. Stochastic games allow the modeling of the operator's choices with nondeterminism, i.e., no assumption on the distributions. If we let the operator have complete power at checkpoints, the game semantics allow the operator to behave completely adversarially in worst-case scenarios. For example, the operator could ask the UAV to fly in the loop $w_2, w_6, w_5, w_2, w_6, \ldots$ forever, resulting in the UAV never being able to cover all the waypoints and complete the mission. To avoid such unrealistic solutions, we define a delegation probability, denoted p_{del}, with which the operator delegates the UAV automation the task of picking the next road. Imposing the delegation probability is a weaker assumption on the operator than a distribution on the actual choices. Moreover, the delegation probability is not specific to any particular map or mission, and thus is easier to quantify, e.g., by measuring how often an operator delegates to the UAV automation during training or past missions.

2) Choices of the Angle Points: Previously, we modeled with MDPs that the UAV automatically chooses which angle to approach or exit a waypoint. By contrast, here we allow the operator to select an angle point nondeterministically at runtime. This could be beneficial, because the operator may have knowledge about the best angle to approach a waypoint in order to capture high-quality sensor imagery.

3) Stochastic Game Model: Fig. 9 shows a fragment of our game model, in which states controlled by the UAV (Player 1) and operator (Player 2) are drawn in circles and boxes, respectively. As with MDPs, this game can be thought of as the product of the two individual player models, where each state is controlled by either the UAV or the operator. After entering waypoint w_6, the UAV is ready for the operator to steer the sensors. The operator's possible behavior of capturing sensor imagery is modeled much as it was in the MDPs. If the quality of the captured imagery is good, then the operator nondeterministically chooses an angle for the UAV to exit waypoint w_6. The red dashed circle highlights how the operator delegates the UAV automation the task of choosing roads based on the delegation distribution. With probability p_{del}, the UAV takes control and

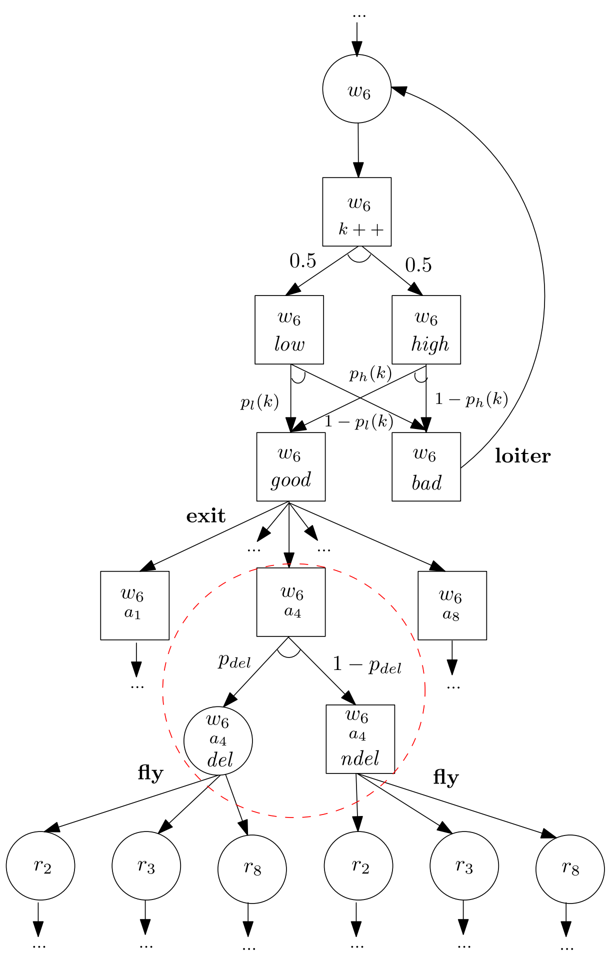

Fig. 9. Fagment of the game between the operator and UAV. States controlled by the UAV and operator are drawn in circles and boxes, respectively.

picks the route nondeterministically; otherwise, the route would be decided by the operator.

B. Analysis

In the following, we report experimental results of applying the game models to an example UAV mission of covering waypoints w_1, w_2, and w_6 (starting at w_1). Assume each loiter takes ten time units and flying each road segment takes 60 time units. We first investigate the minimum expected mission completion time, then analyze the trade-offs between the likelihood of visiting ROZs and the mission completion time. Note that our results represent the worst-case mission performance under all strategies the operator could follow.

Assume each loiter takes ten time units and flying each road segment takes 60 time units, that is, we define a reward structure $\text{time}(loiter) = 10$ and $\text{time}(fly) = 60$. We also define the reward structure $\text{ROZ}(s) = 1$ for all fly-transitions that are outgoing from states corresponding to positions in ROZs (see Fig. 1). To encode the mission objective, we introduce self-loops with zero rewards at the states w_1, w_2 and w_6 of the game model to make them terminal states.

1) Expected Mission Completion Time: Starting at waypoint w_1, the UAV automation chooses a road leading to either w_2 or w_6. Suppose the UAV flies to w_6 first. Since w_6 is a checkpoint, after capturing good quality sensor imagery, with probability p_{del} the operator delegates the UAV automation the authority to choose the next road; otherwise, with probability $1 - p_{del}$, the operator picks the next road. As shown in Fig. 9 (bottom), the operator may pick any of r_2, r_3, and r_8 nondeterministically, among which only r_8 leads directly to w_2 (and the mission completes). If the operator picks r_2 or r_3, it takes the UAV more time to eventually reach w_2 and complete the mission. In the game formulation, we synthesize the optimal UAV strategy for all possible operator strategies, so we need to consider the worst-case scenario in which the operator does not pick the mission favorable choice r_8.

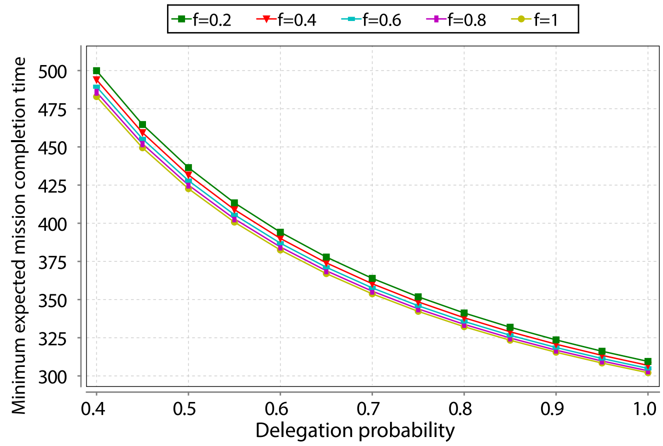

Fig. 10. Expected mission completion time as a function of delegation probability p_{del} and for various fatigue discount factor values f. The fatigue threshold is set to $T = 0$.

We plot the expected mission completion time as a function of delegation probability in Fig. 10. It shows a trend consistent with our previous analysis: the higher the delegation probability (i.e., the less operator intervention), the faster the mission can be completed. Moreover, we vary the values of the operator fatigue discount factor f and observe that the expected mission completion time decreases as the value of f increases, which is consistent with the results reported in Fig. 4 for MDPs. The curves for different f values do not intersect in Fig. 10, because in the current model the fatigue discount factor f only affects the UAV loitering time and is independent of the delegation probability.

2) Tradeoff Analysis: As with MDPs, we can analyze the tradeoffs between multiple mission objectives by making conjunctive queries in the game models.

We consider minimizing the expected time of mission completion and simultaneously minimizing the expected number of ROZ visits, and we investigate the achievable tradeoffs by computing approximations of the Pareto sets. First, we show the effect of the operator accuracy parameters $p_l(0)$ and $p_h(0)$ in Fig. 11. While we observe that higher operator accuracy leads to less time needed to complete the mission (because of less loitering), the UAV performance is robust to within at most 10% deviation in mission completion time (where we fix $p_{\text{del}} = 0.5$, $T = 10$, and $f = 0.7$). Second, we investigate the effect of the delegation probability p_{del} in Fig. 12 (where we fix $p_l(0) = 0.9$, $p_h(0) = 0.8$, $T = 2$, and $f = 0.7$). As the delegation probability decreases, the mission time increases because ROZs have to be avoided, incurring increased delays. The sensitivity of the

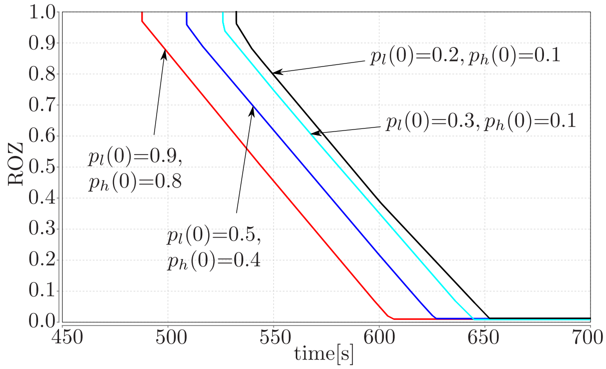

Fig. 11. Tradeoffs under varying operator accuracy function parameters. We set $p_{\mathsf{del}} = 0.5$, $T = 2$, and $f = 0.7$.

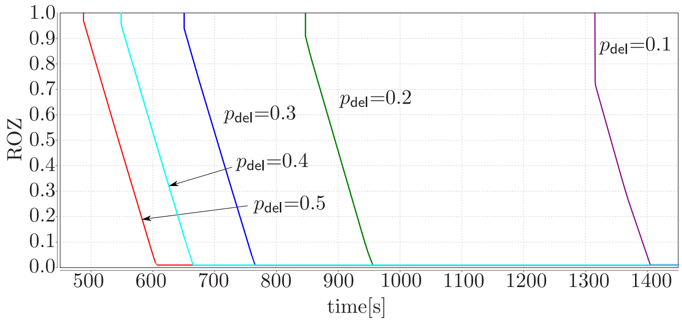

Fig. 12. Tradeoffs under varying delegation probability p_{del}. We set $p_l(0) = 0.9$, $p_h(0) = 0.8$, $T = 2$, and $f = 0.7$.

UAV performance to the delegation probability is higher than to the accuracy parameters $p_l(0)$ and $p_h(0)$. This suggests that, in the design of UAVs interacting with human operators, a key aspect is deciding under which circumstances an operator can outright delegate the flying plan when latent mission objectives exist for the UAV. Note that the Pareto set for $p_l(0) = 0.9$ and $p_h(0) = 0.8$ in Fig. 11 corresponds to the Pareto set for $p_{\mathsf{del}} = 0.5$ in Fig. 12. Furthermore, the general trends of both Figs. 11 and 12 are consistent, that is, the higher the operator accuracy, the less time needed to complete the mission (because of less loitering).

C. Extensions

1) Admissible Operating Regions: It is a weaker assumption on the operator to model choices with nondeterminism than with probability distributions. Therefore, the UAV mission performance obtained in the game models is poorer than that in the MDPs. For example, the expected mission completion time is around 3,500 time units in the game model (when the operator delegates no choice at checkpoints to the UAV, i.e., $p_{del} = 0$), whereas the time decreases significantly if we fix the operator's distribution to obtain an MDP (see Fig. 6). In order to achieve better UAV performance, we may refine the game model by strengthening assumptions on the operator. Consider a (hypothetical) checkpoint where the operator has choices of X, Y, and Z. The fully nondeterministic model, shown in Fig. 13(a), allows the operator to pick any distribution with a randomized strategy. We call such a set of distributions an admissible operating region (AOR) and represent it graphically by the green triangle in Fig. 13(c). Each corner point of an AOR corresponds to a worst-case distribution over the choices, either of a single operator repeatedly entering the checkpoint corresponding to the AOR, or even of a set of operators we wish to consider in our synthesis. In Fig. 13(a), the AOR does not impose any constraints on the worst-case distributions. An AOR is an assumption placed on the operator behavior, and so, if a UAV plan is synthesized using AORs, this plan depends on the operators' behaviors staying within the AOR. Note that in MDPs the distributions of the operator are fixed, and hence represent an AOR consisting of just a single point. Hence, a larger AOR represents a weaker assumption on the operator behavior, making the synthesized UAV plans more robust against unknown or changing operator behavior.

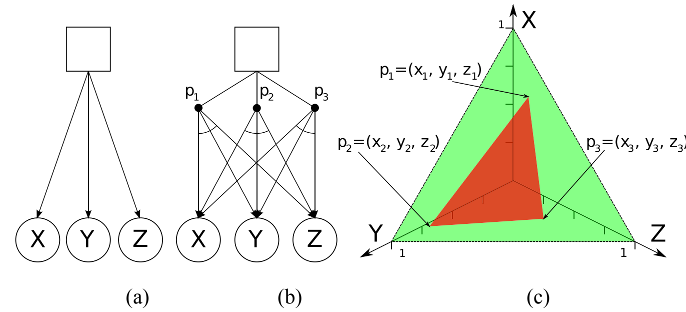

Fig. 13. AOR of (a) the operator modeled fully nondeterministic is endowed with (b) learned distributions p_i, each of which corresponds to (c) a corner of the refined AOR.

2) Directed Game Refinement: If we know about the possible distributions of the operator's choices, we may refine the model and obtain, for example, the one shown in Fig. 13(b), where p_i for $i \in \{1, 2, 3\}$ represent probability distributions (e.g., choosing X with probability x_i). The corresponding AOR, drawn as the red triangle in Fig. 13(c) , is more constrained than the green one, representing more restricted operator behavior and yielding better UAV mission performance. A directed game refinement process involves the following steps: 1) determine an AOR corner point that is the bottleneck for mission performance through analyzing the game model and 2) guide empirical analysis on possible operator behaviors and choices in order to improve the representative statistical model for the AOR. This process of game analysis and distribution learning can be repeated until satisfactory mission performance is obtained.

3) Other Extensions: First, we can build into the model a correlation between the delegation probability and other operator characteristics such as workload and fatigue. For example, we can model that the operator tends to delegate to the UAV automation more often under higher workload or fatigue levels. Modeling in such a way would increase the *coupling* between the operator and the UAV, which is useful in synthesizing individualized UAV piloting plans for different operators. Second, we can synthesize specifications of the form $\varphi^A \rightarrow \varphi^G$ using the methods of [38], where φ^A represents assumptions on the operator and φ^G represents guarantees on UAV mission performance. Hence, instead of explicitly encoding assumptions on the operator's behavior (e.g., via concrete probability distributions in the model), we can use implicit assumptions/constraints expressed in the specifications (e.g., temporal logics).

VII. Critique

We have demonstrated, via a detailed UAV mission planning case study, that MDPs and stochastic two-player games can be used to model human-automation interactions and synthesize human-in-the-loop control protocols. The novelty of our approach lies in building operator behavior models by incorporating insights from human factors literature and analyzing the influence of operator characteristics (e.g., workload, proficiency, and fatigue) on UAV mission performance. We have also applied state-of-the-art techniques in quantitative verification and synthesis to illustrate trade-offs between multiple mission objectives via Pareto curves.

The complexity of the operator model that we consider in this paper grows with the number of road point choices at different waypoints and with larger parameter values (e.g., the fatigue threshold T), while the complexity of the UAV model grows with the size of the mission map. For MDPs, strategy synthesis with multiple objectives takes time polynomial in the size of the model and doubly exponential in the size of the property (for LTL objectives) [39]. For stochastic games, an upper bound is known for single objectives [40] but remains unknown for multiple objectives [38]. Though challenges remain in improving scalability of strategy synthesis, the PRISM tool [35] (which can handle up 10^{10} states) and the approach in [38] were found to be more than sufficient to synthesize strategies for the problem described in this paper.

What we have achieved, however, is just a first step towards controller synthesis for human-in-the-loop autonomous systems. To enable personalized on-demand controller synthesis, many challenges still remain.

First, more expressive operator models are needed. This includes bigger and more representative models that account for additional human characteristics (e.g., operator response time) as well as richer models that incorporate uncertainties in model parameters. A promising direction in this area is to apply data-driven approaches to build libraries of operator models (e.g., by mining behavior patterns from previous operation logs).

Second, more approaches are needed for synthesizing robust controllers that can tolerate inaccuracies in operator models (e.g., due to parameter estimation errors), as discussed in for stochastic games in Section VI-C and for MDPs in [41]. Moreover, since human behaviors may adapt or evolve over time, new approaches will be needed for controller synthesis at runtime. Toward that end, the scalability of synthesis procedures must be improved.

Finally, other application scenarios should be considered, as new scenarios will inevitably reveal further needs for modeling and synthesis. For instance, distributed control of multiple UAVs by a single operator would require accounting for operator situational awareness with respect to each UAV as well as different multi-UAV control modes provided by the automation. Other application domains beyond UAV mission planning could also be considered, for example, autonomous driving.

References

[1] L. Feng, C. Wiltsche, L. Humphrey, and U. Topcu, "Controller synthesis for autonomous systems interacting with human operators," in *Proc. ACM/IEEE Int. Conf. Cyber-Physical Syst.*, 2015, pp. 70–79.

[2] K. W. Williams, "Human factors implications of unmanned aircraft accidents: Flight-control problems," *Human Factors of Remotely Operated Vehicles*, vol. 7, pp. 105–116, 2006.

[3] "Preliminary statement of policy concerning automated vehicles," Nat. Highway Traffic Safety Admin., 2013.

[4] "Integration of civil unmanned aircraft systems (UAS) in the national airspace system (NAS) Roadmap," FAA, 2013.

[5] T. Wongpiromsarn, U. Topcu, and R. M. Murray, "Synthesis of control protocols for autonomous systems," *Unmanned Syst.*, vol. 1, no. 1, pp. 21–39, Jul. 2013.

[6] T. Chen, M. Kwiatkowska, A. Simaitis, and C. Wiltsche, "Synthesis for multi-objective stochastic games: An application to autonomous urban driving," in *QEST*. Berlin, Germany: Springer, 2013, pp. 322–337.

[7] M. Kwiatkowska and D. Parker, "Automated verification and strategy synthesis for probabilistic systems," in *ATVA*. Berlin, Germany: Springer, 2013, pp. 5–22.

[8] L. Humphrey, E. Wolff, and U. Topcu, "Formal specification and synthesis of mission plans for unmanned aerial vehicles," in *Proc. AAAI Spring Symp.*, 2014, pp. 116–121.

[9] M. A. Goodrich and A. Schultz, "Human-robot interaction: A survey," *Foundations and Trends in Human-Computer Interaction*, vol. 1, no. 3, pp. 203–275, 2007.

[10] J. Chen and M. Barnes, "Supervisory control of multiple robots: Effects of imperfect automation and individual differences," *Human Factors: J. Human Factors and Ergonomics Soc.*, vol. 54, no. 2, pp. 157–174, 2012.

[11] K. R. Boff and J. E. Lincoln, *Engineering Data Compendium: Human Perception and Performance*. Wright Patterson AFB, OH, USA: AAMRL, 1988.

[12] S. Makeig, F. S. Elliott, M. Inlow, and D. A. Kobus, "Predicting lapses in vigilance using brain evoked responses to irrelevant auditory probes," Naval Health Research Center, Tech. Rep. TR 90-39, 1990.

[13] H. S. Koelega, "Extraversion and vigilance performance: 30 years of inconsistencies," *Psych. Bull.*, vol. 112, no. 2, pp. 239–258, 1992.

[14] D. Donath, A. Rauschert, and A. Schulte, "Cognitive assistant system concept for multi-UAV guidance using human operator behaviour models," in *Proc. HUMOUS*, 2010.

[15] J. W. Crandall, M. Goodrich, D. R. Olsen Jr., and C. W. Nielsen, "Validating human-robot interaction schemes in multitasking environments," *IEEE Trans. Syst. Man Cybern. A., Syst. Humans*, vol. 35, no. 4, pp. 438–449, Jul. 2005.

[16] M. L. Cummings and P. J. Mitchell, "Predicting controller capacity in supervisory control of multiple UAVs," *IEEE Trans. Syst. Man Cybern. A., Syst. Humans*, vol. 38, no. 2, pp. 451–460, Mar. 2008.

[17] R. M. Yerkes and J. D. Dodson, "The relation of strength of stimulus to rapidity of habit-formation," *J. Comparative Neurol. and Psychol.*, vol. 18, no. 459-482, 1908.

[18] B. Donmez, C. Nehme, and M. L. Cummings, "Modeling workload impact in multiple unmanned vehicle supervisory control," *IEEE Trans. Syst. Man Cybern. A., Syst. Humans*, vol. 40, no. 6, pp. 1180–1190, Nov. 2010.

[19] K. Savla and E. Frazzoli, "A dynamical queue approach to intelligent task management for human operators," *Proc. IEEE*, vol. 100, no. 3, pp. 672–686, Mar. 2012.

[20] L. Breslow, D. Gartenberg, J. McCurry, and J. G. Trafton, "Dynamic operator overload: A model for predicting workload during supervisory control," *IEEE Trans. Human–Machine Syst.*, vol. 44, no. 1, pp. 30–40, Jan. 2014.

[21] P. Langley, J. Laird, and S. Rogers, "Cognitive architectures: Research issues and challenges," *Cognitive Syst. Res.*, vol. 10, no. 2, pp. 141–160, 2009.

[22] G. Gunzelmann, J. B. Gross, K. A. Gluck, and D. F. Dinges, "Sleep deprivation and sustained attention performance: Integrating mathematical and cognitive modeling," *Cognitive Sci.*, vol. 33, pp. 880–910, 2009.

[23] J. G. Trafton, A. Jacobs, and A. M. Harrison, "Building and verifying a predictive model of interruption resumption," *Proc. IEEE*, vol. 100, no. 3, pp. 648–659, Mar. 2012.

[24] K. R. Koedinger, J. R. Anderson, W. H. Hadley, and M. A. Mark, "Intelligent tutoring goes to school in the big city," *Int. J. Artif. Intell. Education*, vol. 8, pp. 30–43, 1997.

[25] M. L. Bolton, E. J. Bass, and R. I. Siminiceanu, "Using formal verification to evaluate human-automation interaction: A review," *IEEE Trans. Syst. Man Cybern., Syst.*, vol. 43, no. 3, pp. 488–503, Apr. 2013.

[26] J. C. Campos and M. D. Harrison, *Formally Verifying Interactive Systems: A Review*. Vienna, Austria: Springer, 1997.

[27] J. Rushby, "Using model checking to help discover mode confusions and other automation surprises," *Reliabil. Eng. Syst. Safety*, vol. 75, no. 2, pp. 167–177, 2002.

[28] J. Crow, D. Javaux, and J. Rushby, "Models and mechanized methods that integrate human factors into automation design," in *Proc. Int. Conf. on Human–Comput. Interaction in Aeronatuics: HCI-Aero*, 2000, pp. 163–168.

[29] J. J. Cañas, A. Antolí, and J. F. Quesada, "The role of working memory on measuring mental models of physical systems," *Psicológica*, vol. 22, pp. 25–42, 2001.
[30] B. Crandall, G. Klein, and R. R. Homan, *Working Minds: A Practitioner's Guide to Cognitive Task Analysis*. Cambridge, MA, USA: MIT, 2006.
[31] W. Li, D. Sadigh, S. S. Sastry, and S. A. Seshia, "Synthesis for human-in-the-loop control systems," in *TACAS*. Berlin, Germany: Springer, 2014, pp. 470–484.
[32] N. J. Cooke and H. K. Pedersen, "Unmanned aerial vehicles," in *Handbook of Aviation Human Factors*. Boca Raton, FL, USA: CRC, 2009.
[33] M. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. New York, NY, USA: Wiley, 1994.
[34] L. S. Shapley, "Stochastic games," *Proc. Nat. Acad. Sci. USA*, vol. 39, no. 10, p. 1095, 1953.
[35] M. Kwiatkowska, G. Norman, and D. Parker, "PRISM 4.0: Verification of probabilistic real-time systems," in *Proc. Comput. Added Verification*, 2011, vol. 6806, pp. 585–591.
[36] T. Chen, V. Forejt, M. Kwiatkowska, D. Parker, and A. Simaitis, "PRISM-games: A model checker for stochastic multi-player games," in *Proc. 19th Int. Conf. Tools Algorithms Constructions Anal. Syst.*, 2013, vol. 7797, pp. 185–191.
[37] J. A. DeJoode, N. J. Cooke, S. M. Shope, and H. K. Pedersen, "Guiding the design of a deployable UAV operations cell," *Human Factors of Remotely Operated Vehicles*, vol. 7, pp. 311–327, 2006.
[38] T. Chen, V. Forejt, M. Kwiatkowska, A. Simaitis, and C. Wiltsche, "On stochastic games with multiple objectives," in *Proc. 38th Int. Symp. Math. Foundations Comput. Sci.*, Aug. 2013, vol. 8087, pp. 266–277.
[39] V. Forejt, M. Kwiatkowska, G. Norman, D. Parker, and H. Qu, "Quantitative multi-objective verification for probabilistic systems," in *Proc. 17th Int. Symp. Tools Algorithms Constructions Anal. Syst.*, Mar. 2011, vol. 6605, pp. 112–127.
[40] A. Condon, "The complexity of stochastic games," *Inf. Comput.*, vol. 96, no. 2, pp. 203–224, 1992.
[41] E. M. Wolff, U. Topcu, and R. M. Murray, "Robust control of uncertain Markov decision processes with temporal logic specifications," in *Proc. 51st IEEE Conf. Decision and Control*, 2012, pp. 3372–3379.

Lu Feng (M'16) received the B.S. degree from Beijing University of Posts and Telecommunications, Beijing, China, in 2007, the M.S. degree from the University of Cambridge, Cambridge, U.K., in 2008, and the D.Phil. degree in computer science from the University of Oxford, Oxford, U.K., in 2014.

She is a Postdoctoral Fellow with the University of Pennsylvania, Philadelphia, PA, USA. Her research focuses on modeling, design and analysis of cyber-physical systems, with applications in medical devices and autonomous robots.

Prof. Feng was a recipient of the James S. McDonnell Foundation Postdoctoral Fellowship Award, UK Engineering and Physical Sciences Research Council (EPSRC) Graduate Scholarship, and Cambridge Trust Scholarship.

Clemens Wiltsche received the B.A. degree in computer science from the University of Cambridge, Cambridge, U.K., and the M.Sc. degree (with distinction) in electrical engineering and information technology from the Swiss Federal Institute of Technology (ETH) Zürich, Switzerland. He is currently working toward the D.Phil. degree in computer science at the University of Oxford, Oxford, U.K.

His research interests are in formal methods for the analysis of stochastic systems and games applied to controller synthesis for cyberphysical systems, with a specific focus on strategy synthesis in competitive environments and multi-objective settings. He was a Research Visitor with the California Institute of Technology in 2012 and with the University of Pennsylvania in 2014.

Mr. Wiltsche was is a recipient of a DAAD (German Academic Exchange Service) Doctoral Grant.

Laura Humphrey (M'12) received the D.Phil. degree in electrical and computer engineering from the Ohio State University, Columbus, OH, USA, in 2009.

She is a Researcher with the Autonomous Control branch of the Aerospace Systems Directorate, Air Force Research Laboratory, Wright-Patterson Air Force Base. Her research interests are in cooperative control systems for unmanned aerial vehicles, formal methods for verification and validation of autonomous systems, and human-automation systems.

Ufuk Topcu (M'08) received the B.S. degree from Bogazici University, Istanbul, Turkey, in 2003, the M.S. degree from the University of California, Irvine, CA, USA, in 2005, and the Ph.D. degree from the University of California, Berkeley, CA, USA, in 2008.

He is an Assistant Professor with The University of Texas at Austin, Austin, TX, USA. Previously, he was a Postdoctoral Scholar with the California Institute of Technology and a Research Assistant Professor with the University of Pennsylvania. His research focuses on the design and verification of autonomous networked systems.