Controller Synthesis for Autonomous Systems Interacting with Human Operators [*]

Lu Feng
Dept. of Computer & Information Science
University of Pennsylvania
lufeng@cis.upenn.edu

Clemens Wiltsche
Dept. of Computer Science
University of Oxford
clemens.wiltsche@cs.ox.ac.uk

Laura Humphrey
Control Science Center of Excellence
AFRL
laura.humphrey@us.af.mil

Ufuk Topcu
Dept. of Electrical & Systems Engineering
University of Pennsylvania
utopcu@seas.upenn.edu

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 operator-autonomy 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.

Categories and Subject Descriptors

I.2.2 [**Automatic Programming**]: Program synthesis; I.2.9 [**Robotics**]: Operator interfaces; H.1.2 [**User/Machine Systems**]: Human factors

General Terms

Verification, Human Factors

1. INTRODUCTION

Autonomous systems are becoming increasingly prevalent in today's society. For example, artificial pancreas systems have been used to help diabetes patients control blood sugar levels; prototypes of driverless cars have been developed and tested on public roads; and 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 have underscored the need for taking into account operator-autonomy interactions when designing an autonomous system. As an example, in the Global Hawk incident [22], the 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; the lack of proper coordination between the operator and automation caused the UAV to eventually run off the runway and crash. This issue has also recently been recognized by regulators. The National Highway Traffic Safety Administration advised that the amount of time and state of situational awareness needed by the 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 requirements [16]. The Federal Aviation Administration raised similar considerations for UAVs: for the purpose of collision avoidance, interfaces must display information about the state of the UAV and nearby air traffic, taking into account visual processing capabilities of the operator [9]. Indeed, vast amounts of data and literature from human factors research are becoming available, which can provide guidance for the design of autonomous systems.

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

[*] The authors are partially supported by James S. McDonnell Foundation Postdoctoral Fellowship, ERC Advanced Grant AdG-246967 VERIWARE, OSD, AFRL, AFOSR grant # FA9440-12-1-0302, AFOSR grant # 13RQ03COR and ONR grant # N000141310778.

automatically generate a protocol (or strategy) for controlling the system that satisfies or optimizes the property. Over 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 [23], while some recent advances in synthesis for probabilistic systems are presented in [4] and [13]. These techniques have been applied to real-world case studies such as UAV mission planning [10] and autonomous urban driving [4].

A key challenge for reactive synthesis is obtaining appropriate models of a human operator's behavior and performance with respect to operator-autonomy interactions. Work in this area is still limited, and available models are not necessarily well suited to reactive synthesis. For instance, [15] finds an upper bound on the number of autonomous vehicles a human operator can effectively supervise, and [18] derives task queueing policies the automation can employ to optimize operator workload. However, these models are relatively abstract and do not capture the types of detailed operator decision-making behaviors we would like to consider for reactive synthesis. A class of operator decision-making models can be found in [20], but this class is limited to tasks in which only two choices are available. Cognitive architectures have been used to model more complex operator decision-making behaviors and performance characteristics [21], but the resulting models are not expressed in a concise mathematical framework amenable to reactive synthesis.

We therefore develop a hypothetical model of operator behavior and performance amenable to reactive synthesis based on high-level trends induced from human factors literature. For instance, data from [1] demonstrate that on a wide variety of tasks, "human reliability" or rate of human error often increases with higher levels of stress and decreases with higher levels of operator 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. One study of vigilance tasks found declining response rates after as little as 3 minutes of task performance, with response rates eventually plateauing at 70-80% of initial rates [14]. 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 [11]. 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 [8].

Though a particular operator's behavior may be unknown at system design time, relevant statistics can be obtained, *e.g.,* via extracting possible operator behavior patterns from prior information such as training logs. Moreover, methods such as *cognitive task analysis* [6] can be applied to reveal how an operator would respond to various events. We expect advances in data-driven modeling to help create individualized libraries of operator models and support on-demand controller synthesis as operators, missions, and vehicles change.

As an illustrative example for this paper, we consider a simple scenario involving road network surveillance by a UAV. We first build abstractions for operator-autonomy interactions 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), as previously mentioned. The operator-autonomy 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 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 also demonstrate the impact of operator characteristics on UAV mission performance.

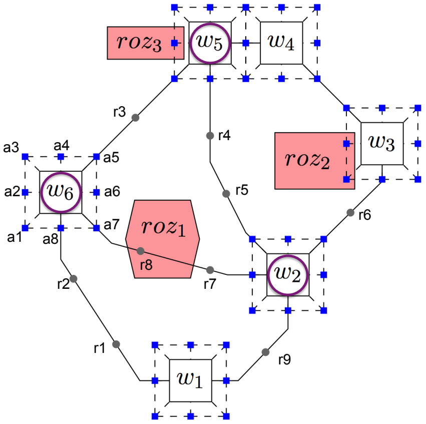

Figure 1: A road network for UAV ISR missions (adapted from [10]).

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 from the operator and the UAV, we augment the MDP models to *stochastic two-player games.* The goal is 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.,* the assignment of operator and autonomy to tasks). As with MDPs, we can similarly synthesize individualized UAV strategies and analyze mission objective trade-offs with games. In addition, we may guide the refinement of the *admissible operating region* and provide informative feedback to operators for achieving better mission performance.

The rest of the paper is organized as follows. We describe the motivating example in Section 2 and introduce formal specifications and models in Section 3. We present our modeling approach and experimental results for MDPs and stochastic two-player games in Section 4 and Section 5, respectively. Finally, we remark on potential future work in Section 6.

2. MOTIVATING EXAMPLE

As an illustration of synthesis for autonomous systems interacting 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. Figure 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 [5], 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 Figure 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 [10], including safety, reachability, coverage, sequencing of waypoints, *etc.* 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 vs. worst-case) of prior knowledge about imperfections in the operator's behavior and the interactions (*e.g.,* the operator's interference in the choices made by the autonomy protocol) between the autonomy protocols and the operator.

3. 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 $\mathit{Dist}(Q)$ 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.

3.1 Markov Decision Processes

Markov decision processes (MDPs) [17] 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 $\delta \subseteq S \times \mathcal{A} \times \mathit{Dist}(S)$ is a *transition relation*, assigning at most one distirbution per state and action. Let $A(s) \overset{\text{def}}{=} \{a \in \mathcal{A} \,|\, \exists \mu \in \mathit{Dist}(S) \,.\, (s, a, \mu) \in \delta\}$ 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 $\mathit{IPath} \overset{\text{def}}{=} \bigcup_{s \in S} \mathit{IPath}_s$ and $\mathit{FPath} \overset{\text{def}}{=} \bigcup_{s \in S} \mathit{FPath}_s$. We use a *strategy* to resolve the nondeterministic choices in an MDP, which is defined formally as a function $\sigma : \mathit{FPath} \to \mathit{Dist}(\mathcal{A})$ 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 $\rho \in \mathit{FPath}$, 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 $\mathbf{P}(\rho) \overset{\text{def}}{=} \prod_{i=0}^{n-1} \delta(s_i, a_i)(s_{i+1})$. 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 $r : (S \times \mathcal{A}) \to \mathbb{Q}$, and the *total reward* of an infinite path $\rho = s_0 \xrightarrow{a_0} s_1 \xrightarrow{a_1} \cdots$ as $\mathsf{rew}(r)(\rho) \overset{\text{def}}{=} \lim_{N \to \infty} \sum_{i=0}^{N} r(s_i, a_i)$, if the limit exists. The *expected total reward* computes the expectation of rewards accumulated along all the infinite paths $\mathit{IPath}_{\overline{s}}$ starting at the initial state $\overline{s}$ in an induced DTMC M^σ, denoted by $\mathbb{E}^\sigma[\mathsf{rew}(r)] \overset{\text{def}}{=} \int_{\rho \in \mathit{IPath}_{\overline{s}}} \mathsf{rew}(r)(\rho) dPr_{\overline{s}}(\rho)$. 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, *resp.*, 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 the minimum probability reachability in MDPs. The strategy synthesis of CQs reduces to a linear programming problem and can be solved in polynomial time. We refer to [13] 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 $\big((s_1, s_2), a, \mu_1 \times \mu_2\big) \in \delta$ 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 we are interested in, 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.

3.2 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) [19] by distinguishing two types of nondeterminism, each controlled by a player. We use Player 1 to represent the controllable part of a system for which we want to synthesize a strategy and use 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 Player 1 states S_1 and Player 2 states S_2, while the initial state $\overline{s}$, the action set $\mathcal{A}$, and the translation relation δ are defined exactly the same as in 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 $\sigma_i : \mathit{FPath}_i \rightarrow \mathit{Dist}(\mathcal{A})$ 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) of 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 [13]. 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 [4].

3.3 Tools and Implementations

We build concrete scenario models and perform experiments using the following tools. We use PRISM [12] for the modeling and synthesis of MDPs, use its extension named PRISM-games [2] for the strategy synthesis in stochastic games with single objectives, and use the implementation of [4] for multi-objective synthesis in games.

4. MDP FOR OPERATOR-AUTONOMY INTERACTIONS

Consider the first variant of the scenario described in Section 2. 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 levels. 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

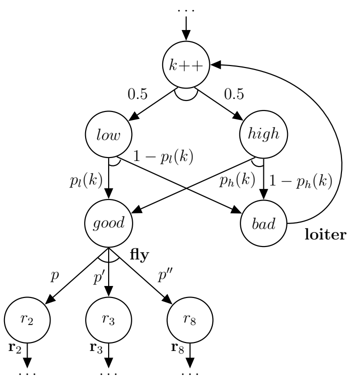

Figure 2: A fragment of the operator model M_{OP}, representing a possible operator's behavior at the waypoint w_6 (with omitted parts drawn as $\cdots$).

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.

4.1 Modeling

The operator model. As discussed in Section 3.1, we build abstractions of the operator's possible behavior as a fully probabilistic model M_{OP}. Figure 2 shows a fragment of the model, representing the possible behavior at the 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 the value of k increasing 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 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 [1]. Moreover, one study [14] 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 of 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 certain number of tasks. Analogously, we define the accuracy function $p_h(k)$ for *high* workload. 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 $p_h(0)$.

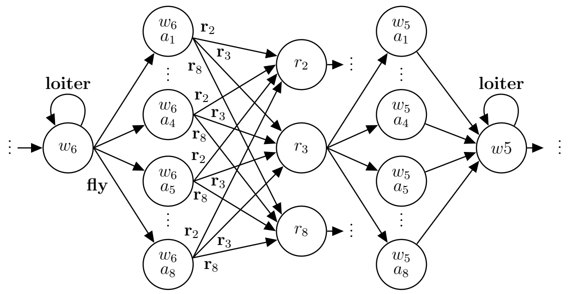

Figure 3: A fragment of the UAV model M_{UAV}, representing the UAV loitering and flying over the waypoints w_6 and w_5.

If the quality of the captured imagery is *bad*, the operator would ask the UAV to *loiter* at the current waypoint in order to collect more sensor imagery; otherwise, the operator allows the UAV to *fly* to another waypoint. Since w_6 is a checkpoint, the operator selects the next road for the UAV (at any non-checkpoint waypoint, the UAV controller chooses the road). We model the operator's choices 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 extract operator behavior patterns for generic classes of checkpoints using human factors research methods such as cognitive task analysis [6] 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 the operator behavior library statistics, *e.g.,* at "three-way crossing with a bridge"-type checkpoints, the operator chooses the bridge with probability 0.5.

The UAV model. We model the UAV's piloting behavior as an MDP M_{UAV}, which contains 63 states (6 waypoints, 6×8 angle points, and 9 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. Figure 3 shows a fragment of the model[1], 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 $a_1, \cdots, a_8$ 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 Figure 3).

The operator-autonomy interactions. We model the interactions between the operator and the UAV by composing M_{OP} and M_{UAV}, which synchronize over common actions

[1] 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.

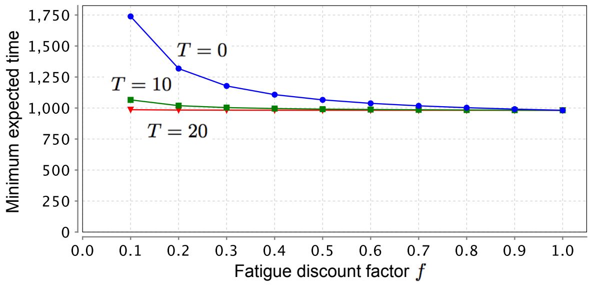

Figure 4: The effect of operator fatigue on minimum expected mission completion time, for different values of the fatigue threshold T and discount factor f (with fixed parameters $p_l(0) = 0.9$ and $p_h(0) = 0.8$).

loiter and *fly*, and obtain a product MDP $M_{\mathsf{OP}} \| M_{\mathsf{UAV}}$. 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 size of the model depends on the model parameters (*e.g.,* the fatigue threshold T). In our experiments, the MDPs generally have around $60,000$ states, while the PRISM tool [12] that we use can handle up to 10^{10} states for MDPs.

4.2 Analysis

We present experimental results using the above MDP model. In particular, we consider the following questions proposed in Section 2: 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.

Effects of operator fatigue. We first consider the UAV surveillance mission of covering all six waypoints in Figure 1. The objective is to complete the mission as fast as possible. Assume that each loiter takes 10 time units and flying between any neighboring waypoint and/or road point takes 60 time units. Figure 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 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.

Effects of operator proficiency and workload. The operator's accuracy in steering sensors and capturing good quality imagery are affected by proficiency and workload. Figure 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 of $p_l(0)$ 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.

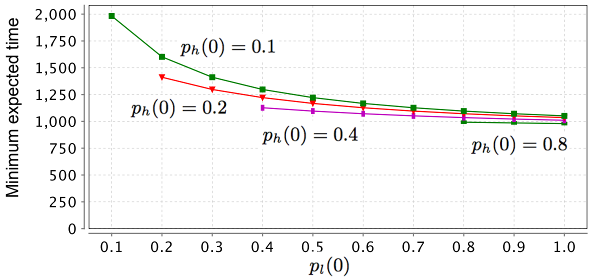

Figure 5: The effect of operator proficiency and workload on minimum expected mission completion time, for different initial values of the accuracy functions $p_l(0)$ and $p_h(0)$ (with fixed parameters $T = 10$ and $f = 0.7$).

Synthesizing individualized strategies. Suppose it is possible to parameterize the model for individual operators, *e.g.,* using the operator-dependent behavior patterns library as described in Section 4.1. 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 Figure 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 of loitering is negligible). The operator's choices at the 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:

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

costing 8 units of fuel. Suppose there is another operator whose behavior pattern library shows a higher likelihood of entering ROZs (*e.g.,* due to the lack of training 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:

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

which only requires 5 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 enhance the coupling of the operator's choice distributions and other operator characteristics such as fatigue and workload.

Trade-offs between mission objectives. Interviews with UAV domain experts [7] 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 and risk UAV survival). When there are multiple potentially conflicting mission objectives, conjunctive queries for MDPs (see Section 3.1) can help to investigate the trade-

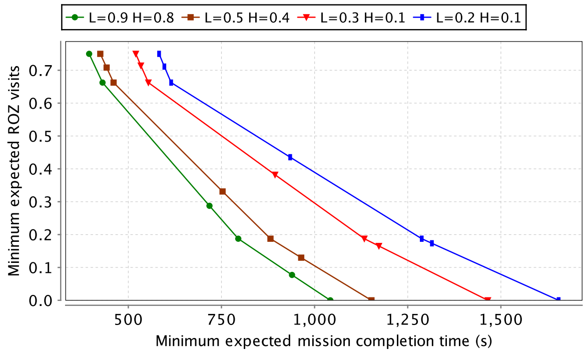

Figure 6: Pareto curves of two mission objectives for various accuracy function parameters (L is $p_l(0)$ and H is $p_h(0)$), with fixed fatigue parameters $T = 10$ and $f = 0.7$.

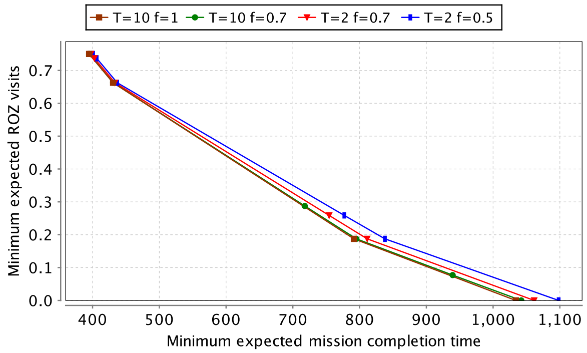

Figure 7: Pareto curves of two mission objectives for various values of the operator's fatigue threshold T and discount factor f, with fixed accuracy function parameters $p_l(0) = 0.9$ and $p_h(0) = 0.8$.

offs. 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). Figure 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 1,040 time units. Pareto curves provide a useful visualization of trade-offs 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 Figure 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. Figure 7 shows the shift of Pareto curves when varying values of the operator's fatigue threshold T and discount factor f, which is not as dramatic as in Figure 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 $p_h(0)$. Figure 8 illustrates various Pareto curves for operators with different probability distributions of route choices at the checkpoints. The green lines (with dots) across Figures 6, 7 and 8 are drawn with the same set of model parameters, serving as a baseline for comparison. Figure 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).

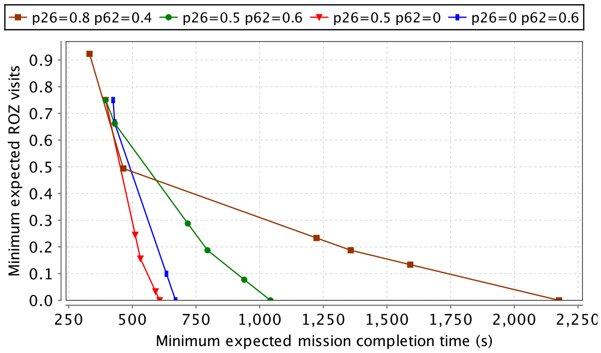

Figure 8: Pareto curves of two mission objectives for various probability distributions of operator choices at checkpoints (*e.g.,* p_{ij} is the probability of choosing the direct route from waypoint w_i to w_j), with fixed parameters $p_l(0) = 0.9$, $p_h(0) = 0.8$, $T = 10$ and $f = 0.7$.

5. STOCHASTIC GAMES FOR OPERATOR-AUTONOMY 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.

5.1 Modeling

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.

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 run time. 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.

The stochastic game model. Figure 9 shows a fragment of our game model, in which states controlled by the UAV and operator are drawn in circles and boxes, respectively. 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 picks the flying route nondeterministically; otherwise, the route would be decided by the operator.

5.2 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 10 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.

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 $1 - p_{\text{del}}$, the operator picks the next road. As shown in Figure 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.

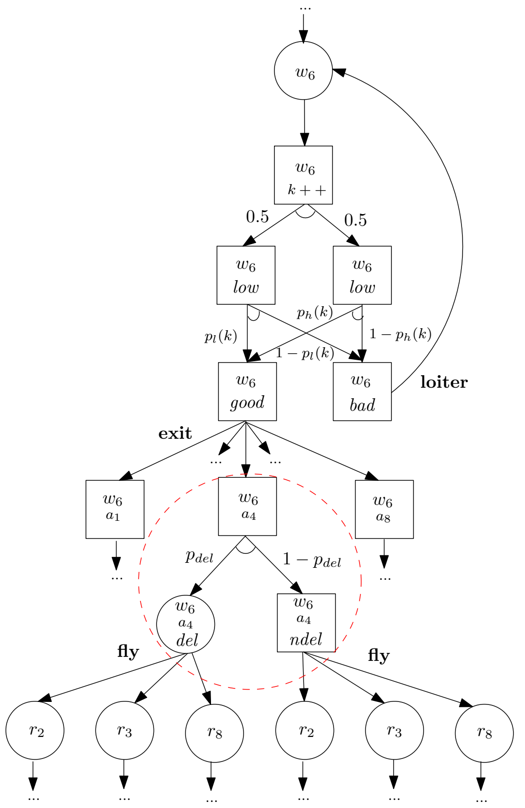

Figure 9: A fragment of the game between the operator and UAV. States controlled by the UAV and operator are drawn in circles and boxes respectively.

We plot the expected mission completion time as a function of delegation probability in Figure 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 Figure 4 for MDPs. The curves for different f values do not intersect in Figure 10, because in the current model the fatigue discount factor f only affects the UAV loitering time and is independent of the delegation probability.

Trade-off analysis. As with MDPs, we can analyze the trade-offs between multiple mission objectives by asking conjunctive queries in the game models. For example, consider the query of minimizing the expected number of ROZ visits

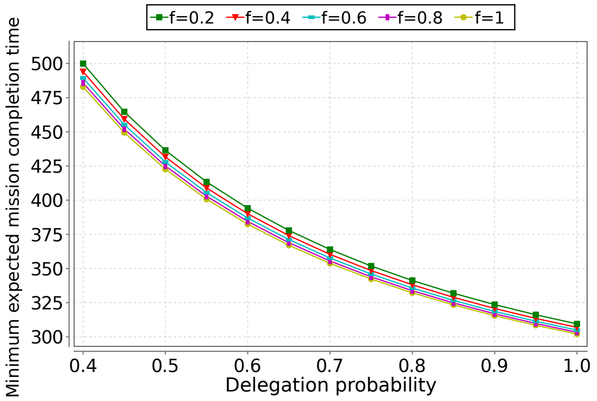

Figure 10: Expected mission completion time as a function of delegation probability p_{del} and for various fatigue discount factor values f. The fatigue threshold is set to $T = 0$.

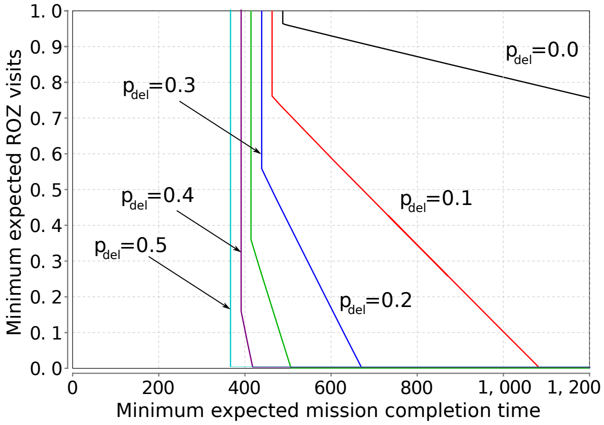

Figure 11: Pareto curves of two mission objectives for various values of the delegation probability p_{del}. We set $p_l(0) = 0.9$, $p_h(0) = 0.8$, $T = 2$ and $f = 0.7$.

and simultaneously minimizing the expected time of mission completion. Figure 11 shows the Pareto curves of these two objectives, for different values of the delegation probability. The trends show that more visits to ROZs are expected in order to complete the mission more quickly and, as the delegation probability decreases, longer missions result from mitigating the chances of entering ROZs. We also change values of the operator accuracy parameters $p_l(0)$ and $p_h(0)$ and plot the corresponding Pareto curves in Figure 12. Note that, though we use the same set of accuracy parameter values as in Figure 6, a direct comparison of the results is not possible, because here we set the delegation probability $p_{\mathrm{del}} = 0.1$ whereas the results of Figure 6 are based on specific probability distributions of operator choices at checkpoints in the MDP models. Nevertheless, the general trends of both Figures are consistent: that is, the higher the operator accuracy, the less time needed to complete the mission (because of less loitering).

5.3 Extensions

Admissible Operating Regions. It is a weaker assumption on the operator to model the choices with nondeterminism than with probability distributions. Therefore, the

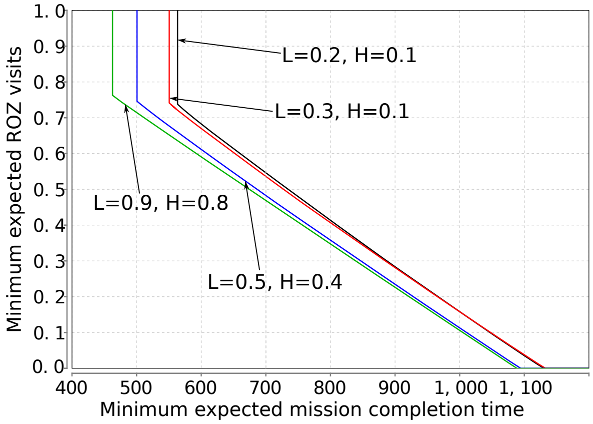

Figure 12: Pareto curves of two mission objectives for various accuracy function parameters (L is $p_l(0)$ and H is $p_h(0)$), with delegation probability $p_{\mathrm{del}} = 0.1$ and fatigue parameters $T = 10$ and $f = 0.7$.

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.,* $p_{\mathrm{del}} = 0$), whereas the time decreases significantly if we fix the operator's distribution to obtain an MDP (see Figure 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 Figure 13 (left), 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 Figure 13 (right). 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 Figure 13 (left), 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 synthesied using AORs, this plan depends on the operators' behaviors staying with in 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 synthesised UAV plans more robust against unknown or changing operator behavior.

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 Figure 13 (middle), 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 Figure 13 (right), 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

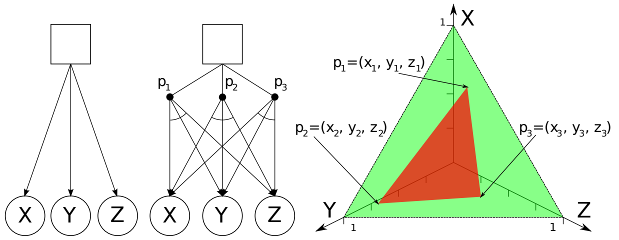

Figure 13: The AOR of the operator modeled fully nondeterministic (left) is endowed with learned distributions p_i (middle), each of which corresponds to a corner of the refined AOR (right).

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.

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 $\varphi^A \to \varphi^G$ using methods of [3], 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).

6. CONCLUDING REMARKS

In this paper, we present an approach to synthesize control protocols for autonomous systems interacting with human operators. Depending on the type (i.e., probabilistic and/or nondeterministic) of knowledge about the uncertainties and imperfections in the operator-autonomy interactions, we use abstractions based on MDPs and augment these models to stochastic two-player games. We demonstrate the influence of operator characteristics (*e.g.,* workload, proficiency, and fatigue) on the mission performance. We also analyze the trade-offs between multiple mission objectives via Pareto curves. If models for individual operators are available, our approach can even be applied to synthesize individualized control protocols for different operators.

There are a few directions for future work. First, in addition to the extensions discussed in Section 5.3, we would like to develop bigger and more representative models including human factors (*e.g.,* operator response time) beyond what we have considered in this paper. For example, we may incorporate uncertainties in the model parameters and consider richer operator interactions. Second, we would like to consider a variety of application scenarios: for instance, the distributed control of multiple UAVs by a single operator, the loss of situational awareness of the operator, and limitations due to operator-autonomy interfaces. Last but not least, we would like to build a library of data-driven models for operators, which is a critical step towards personalized on-demand controller synthesis.

7. REFERENCES

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

[2] T. Chen, V. Forejt, M. Kwiatkowska, D. Parker, and A. Simaitis. PRISM-games: A model checker for stochastic multi-player games. In *TACAS*, pages 185–191. Springer, 2013.

[3] T. Chen, V. Forejt, M. Kwiatkowska, A. Simaitis, and C. Wiltsche. On stochastic games with multiple objectives. In *MFCS*, pages 266–277. Springer, 2013.

[4] T. Chen, M. Kwiatkowska, A. Simaitis, and C. Wiltsche. Synthesis for multi-objective stochastic games: An application to autonomous urban driving. In *QEST*, pages 322–337. Springer, 2013.

[5] N. J. Cooke and H. K. Pedersen. Unmanned aerial vehicles. In *Handbook of Aviation Human Factors.* CRC Press, 2009.

[6] B. Crandall, G. Klein, and R. R. Homan. *Working Minds: A Practitioner's Guide to Cognitive Task Analysis.* MIT Press, 2006.

[7] J. A. DeJoode, N. J. Cooke, S. M.Shope, and H. K. Pedersen. Guilding the design of a deployable uav operations cell. In *Human Factors of Remotely Operated Vehicles*, volume 7, pages 311–327, 2006.

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

[9] Federal Aviation Administration. *Integration of Civil Unmanned Aircraft Systems (UAS) in the National Airspace System (NAS) Roadmap*, 2013.

[10] L. Humphrey, E. Wolff, and U. Topcu. Formal specification and synthesis of mission plans for unmanned aerial vehicles. In *Proc. of the AAAI Spring Symposium*, 2014.

[11] H. S. Koelega. Extraversion and vigilance performance: 30 years of inconsistencies. *Psychological Bulletin*, 112(2):239 – 258, 1992.

[12] M. Kwiatkowska, G. Norman, and D. Parker. PRISM 4.0: Verification of probabilistic real-time systems. In *CAV*, pages 585–591. Springer, 2011.

[13] M. Kwiatkowska and D. Parker. Automated verification and strategy synthesis for probabilistic systems. In *ATVA*, pages 5–22. Springer, 2013.

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

[15] M.L.Cummings. Operator interaction with centralized versus decentralized UAV architectures. In *Handbook of Unmanned Aerial Vehicles.* Springer, 2015. in press.

[16] National Highway Traffic Safety Administration. *Preliminary Statement of Policy Concerning Automated Vehicles*, 2013.

[17] M. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming.* John Wiley and Sons, 1994.

[18] K. Savla and E. Frazzoli. A dynamical queue approach to intelligent task management for human operators. *Proc. of the IEEE*, 100(3):672–686, 2012.

[19] L. S. Shapley. Stochastic games. *Proc. Natl. Acad. Sci. USA*, 39(10):1095, 1953.

[20] A. Stewart, M. Cao, A. Nedic, D. Tomlin, and N. E. Leonard. Towards human–robot teams: Model-based analysis of human decision making in two-alternative choice tasks with social feedback. *Proc. of the IEEE*, 100(3):751–775, 2012.

[21] G. Trafton, L. Hiatt, A. Harrison, F. Tamborello, S. Khemlani, and A. Schultz. ACT-R/E: An embodied cognitive architecture for human-robot interaction. *Journal of Human-Robot Interaction*, 2(1):30–55, 2012.

[22] K. W. Williams. Human factors implications of unmanned aircraft accidents: Flight-control problems. In *Human Factors of Remotely Operated Vehicles*, volume 7, pages 105–116, 2006.

[23] T. Wongpiromsarn, U. Topcu, and R. M. Murray. Synthesis of control protocols for autonomous systems. *Unmanned Systems*, 1(1):21–39, Jul 2013.