IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received June 21, 2021, accepted July 13, 2021, date of publication July 19, 2021, date of current version July 27, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3098060

Modeling Operator Performance in Human-in-the-Loop Autonomous Systems

SHAHABUDDIN MUHAMMAD, (Member, IEEE)
Department of Computer Science, College of Computer Engineering and Science, Prince Mohammad Bin Fahd University, Al-Khobar 31952, Saudi Arabia
e-mail: smuhammad@pmu.edu.sa

ABSTRACT With the increasing role of human-in-the-loop (HITL) based autonomous systems, researchers have made several attempts to understand how an operator's performance is affected by various parameters. In such systems, the performance of the operator directly influences the overall system performance. Although operator performance has been extensively studied at various psychological, behavioral, and physical levels, to the best of our knowledge there is a lack of literature addressing how a variety of operator's internal characteristics and external environmental factors affect the performance of the system for various mission objectives. This paper addresses this issue and proposes a probabilistic model checking based approach to assess the performance of an HITL-based autonomous system. We model the system as a Markov decision process and use probabilistic model checking to assess the impact of various operator and environment parameters on application-specific mission objectives. In addition to considering key operator characteristics in the fatigue model, the proposed method captures dynamic workload, task type, and the impact of various break policies on overall mission objectives. The model can be adapted to carry out system analysis at a higher level of abstraction for a variety of applications. The proposed method is applied to assess various scenarios in a case study from the literature. The results obtained using the proposed method can help a system designer evaluate the impact of various operator and environment characteristics to improve application-specific mission objectives.

INDEX TERMS Operator modeling, semi-autonomous systems, probabilistic model checking, UAV.

I. INTRODUCTION

Human skills, expertise and participation play an important and unique role in various systems including manufacturing, surveillance, and security operation centers (SOCs). In the quest of making systems more proficient, there has been continuous effort to make them more autonomous. There are several tasks that require ultra-precision (or ultra-speed) and thus they may not be performed by humans. It is highly recommended to strive for achieving complete independence when an unsupervised operation is essential. Examples of automation can be seen in different aspects of our lives, including manufacturing and production systems, security and surveillance systems, and home appliances. Although machines may perform some tasks with high efficiency and accuracy, it is not true that full autonomy obviates the need for human involvement. Several deployments of autonomous systems (such as military UAV, NASA rovers, unmanned underwater vehicles, and disaster inspection robots) rely on human expertise in crucial roles. In the following, we discuss HITL-based systems and their applications in autonomous systems.

The associate editor coordinating the review of this manuscript and approving it for publication was Wen-Sheng Zhao.

A. THE HITL-BASED AUTONOMOUS SYSTEMS

Although automation offers its benefits, there is still a great need for HITL-based systems as complete autonomous systems are not desirable for various reasons [1]. For instance, automation may not always substitute human activities (such as those requiring sophisticated reasoning and decision making) without affecting the operation of the system negatively. Similarly, some systems can be made autonomous only by adding severe constraints on the task and the associated context. Best system performance is only possible via coordination and collaboration of humans and machines.

In several HITL-based systems, algorithms interact with agents and optimize their learning behavior through these interactions. This helps solve computationally hard problems by reducing exponential search space through heuristic selection of samples using human expertise [2]. In this way, HITL-based systems can reduce the complexity of an NP-hard problem by taking help from a human agent during the learning phase.

Several researchers have proposed HITL-based machine learning that can speed up the learning process by intelligently tracking the changes and intermediate results over time. The authors in [3] propose Helix, an attempt at such a system with speedups of up to ten times on typical iterative workflows against competing systems. Moreover, in specific domains, such as clinical medicine, several researchers have proposed conditions for AI-based technological solutions and argue that they should be only integrated if they fulfill the moral imperative of humanity [4]. The researchers in [5] have highlighted the advantages of human vs machines in HITL control systems.

Currently, semi-autonomous systems (SAS) are more viable where intended operations are performed in coordination with a human operator who supervises, guides, and collaborates with an autonomous system [6]–[9]. For example in SOCs, software agents analyze sensor inputs from various devices and alert human operators for appropriate action in non-obvious or peculiar situations. There are times when a fully autonomous component is either incapable or too costly in handling a situation that requires specific domain expertise or knowledge about that situation. For instance, the defense mechanism at a SOC relies on an analyst's expertise and knowledge (domain as well as situational) to analyze activities across the system and defend against attacks that go undetected by the auto-defense mechanisms, such as firewalls or intrusion prevention systems. The threat detection process is a knowledge-intensive task in which the expertise of a security analyst is leveraged to quickly eliminate false alerts and escalate the real alerts for further analysis and appropriate response [10]. Due to the complexity and size of the networks and the dynamic nature of the behavior of the attackers, fully automated systems can generate a high rate of false positives [11], which can be significantly reduced by the experiences of the HITL within the process of automation.

Failing to incorporate human operator in autonomous systems may result in severe accidents. For example, in the Global Hawk incident [12], the automated mission planning software caused a UAV to run off the runway and crash by accelerating to a ground speed much greater than the operator recommended taxi speed. To respond to such incidents and take appropriate control, it has been emphasized to include humans when designing safe autonomous systems. For example, the National Highway Traffic Safety Administration has given such recommendations for self-driving vehicles. They have also recommended for operator training and certification for these applications [13]. Similarly, the Federal Aviation Administration has recommended adding human interface for UAV applications [14].

An SAS is a system that can operate autonomously under some conditions, but may or may not require human intervention in order to achieve its assigned goals. In general,

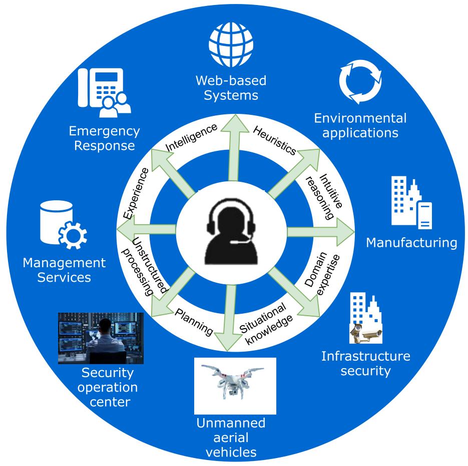

FIGURE 1. The operator in HITL-based autonomous system.

an SAS is categorized as type I (SAS-I) and type II (SAS-II). In SAS-I, planning process does not factor possible human interventions [15]. In this type, human intervention is either not needed or is unavailable and hence the system itself is incapable of fully analyzing goal reachability. Examples of SAS-I systems include Roomba vacuum cleaner and Curiosity Mars rover. SAS-I, being unable to reason about human interventions, cannot guarantee that the system will remain in a "live state" using the system's planning and reasoning capabilities. SAS of type II (SAS-II), on the other hand, involves systems where the planning process includes knowledge about possible human interventions for completing the assigned tasks. Therefore, planning a SAS-II includes human actions, and uncertainties in such interactions must be handled in the model.

MDPs (Markov decision process) and POMDPs (partially observable MDPs) are useful representative models to capture such systems. Particularly for domains where (1) a human operator may have superior abilities to observe or infer the current state of the process (e.g. driving a car), (2) a human operator may be able to perform actions that are not available to the SAS (e.g., climb stairs), or (3) a human operator may have a different level of competence in performing certain actions (e.g., removing a stuck light bulb without breaking it) [15]. There are several challenges associated with modeling human actions in SAS-II systems, including experience, context switching, activity time, and fatigue. Our paper is an attempt to analyze HITL-based autonomous systems (of type II) where achieving application-specific mission objectives depends on different operator and work environment based parameters. The need for an operator for various applications is shown in Fig. 1.

B. THE PERFORMANCE OF A SEMI-AUTONOMOUS SYSTEM

The performance of an HITL-based autonomous system can be defined based on the type of assigned tasks and the corresponding mission objectives. For instance, [16] defines task performance as the adjusted expected value on the UAV routing task, accuracy on the automated payload task, and throughput on the chat communication task. Various factors such as task type, current workload, and other environment variables may affect system performance. In addition to environment variables, the performance of an HITL-based autonomous system also depends on the characteristics of the operator. There are many research studies that analyze human characteristics and their impact on task performance [16]–[18]. The operator performance largely depends on operator attributes, work environment parameters, and other system variables. Operator attributes can be broadly categorized into two groups: long-term and short-term characteristics. The long-term operator characteristics include parameters such as age and experience. The short-term operator characteristics include parameters such as sleep schedule, base proficiency, and fatigue. The operator intrinsic features such as introvert or extrovert also influence the task performance [8]. In addition to operator characteristics, work related factors, such as shift time, shift duration, workload variations, and task variety, also influence the performance of an operator.

Human operator modeling is very helpful in getting insights into the system performance and maintaining a competitive advantage for an enterprise. Modeling helps in understanding the bottlenecks in the system, developing tailored motivational strategies, and recruiting suitable candidates to meet system objectives. Further, input/output data of such modeling can be mined to find unknown repeated behaviors which can increase understanding of the operator's action in particular situations. The authors in [5] highlight several such advantages of modeling an operator in HITL-based autonomous systems. They posit that manual control exists even in highly automatic intelligent systems, such as aircraft and ground vehicles, and a skilled human operator can help the system meet its desired proficiency. The authors in [19] emphasize the need for a detailed understanding of the types of HITL controls. They also highlight the need for techniques to derive models of human behaviors and the importance of determining how to incorporate human behavior models into the formal methodology of feedback control.

Researchers have done physiological, biological, and simulation-based studies for understanding factors that contribute to better operator performance [18], [20], [21]. Although these studies are of extreme importance, typically either they are too complex to be directly used by system designers involving human-automation interaction or the analysis in these studies does not consider environmental aspects such as workload and application dependent aspects such as mission objectives which are crucial in the analysis of SAS.

Formal verification techniques have gained popularity where systems are modeled and analyzed in a formal setting. Formal verification is used to assess the correctness of a design with respect to desired behaviors. Using such techniques, one can mathematically analyze the space of possible behaviors of a design, rather than focusing on computing results for particular values [22]. Due to the rigor of the formal framework, the verification achieves reliability of results. That is, if a design meets the given specification, it is guaranteed to be correct with respect the specification [23]. This is sometimes refers to as guaranteed contract compliance [24].

There is a growing need for rigorous, formal techniques to verify the correctness of systems with stochastic behavior [25]. Probabilistic model checking (PMC) is one of the most promising techniques to assess system design that exhibit stochastic behavior. PMC has already been used in a variety of systems to ascertain the correctness of a system with probabilistic evolution, perform quantitative analysis, and provide probabilistic guarantees. For instance, PMC has been used to verify several properties in embedded devices that are prone to failure, in communication networks where messages sent across the networks may get lost, and in wireless protocols such as Bluetooth and ZigBee where randomization is used [26].

In PMC, system parameters are captured in *states* and state *transitions* capture probabilistic as well as nondeterministic system evolution. The system is modeled at a higher level of abstraction that captures key parameters that influence the desired system behavior. Modeling and analyzing a system using such techniques offer many advantages such as:

- Since the system is represented by states and transitions in a well-studied formal system (such as Markov chain or Markov decision process), the system evolution can be written unambiguously and implicit assumptions can be articulated clearly.
- The inherent rigor in the formal methods provides a better understanding of the component interactions.
- Since the specifications that need to be verified are written in a formal language, usually some form of temporal logic, application objectives can be defined formally.
- Once a system is captured and specifications are written, the analysis can be carried out early on at the design stage. Probabilistic guarantees can be obtained, or input parameters may be tuned to obtain desired system behavior.

The power of probabilistic model checking lies in the exhaustive exploration of every possible state of the model. Moreover, a wide range of quantitative properties can be captured and verified using these techniques. In contrast to non-formal techniques (e.g. discrete-event simulation), which generate approximate results by averaging from a large number of random samples, probabilistic model checking employs numerical computation methods to calculate exact results [27]. Even though formal methods have been successfully used to analyze a variety of systems and several flaws have been identified in various systems using these methods, they cannot fully replace standard quality assurance techniques. Usually, they are assumed to complement conventional methods in system design.

This paper presents a formal verification-based approach to model and study the aspects of the human operator that monitors SAS. Such models can help designers of semi-automated systems to make predictions about system performance based on operator features, work characteristics, and mission objectives. The HITL-based autonomous system is inherently stochastic which is influenced by external nondeterministic events. Therefore, we use MDP to model such system and employ PMC to analyze its behavior for given specifications.

Probabilistic extension of linear temporal logic (LTL) is used to capture application-specific mission objectives. In particular, given the following:

- Working environment parameters such as workload pattern, task type, and work shift.
- Operator characteristics such as age, experience, task proficiency, and personality traits.
- External parameters such as the number of tasks, time per task, fatigue thresholds, break and policies.

A SAS designer can measure application-specific performance of an operator based on:

- Mission type
- Mission objectives, and
- For a single or multiple optimization variables.

The model is implemented and verified using a well-known probabilistic model checker (PRISM) [26]. PRISM uses given property specifications to verify whether the system meets the desired properties. The following are the key contributions of this paper:

- The model can be adapted to capture scenarios in a variety of applications where an operator helps a semi-autonomous system by providing feedback on the sensory output of the automated components.
- In addition to the task-based fatigue model of the operator, the system utilizes stochastic workload and task type models to capture multiple working environments as well as the relation between the operator's intrinsic characteristics with task types.
- The system also captures the effect of break policies on the operator performance. PRISM solves MDP to resolve nondeterminism in a way that optimizes desired objectives.
- For two competing objectives, the model provides Pareto optima that can be very helpful for a system designer.
- The efficacy of the proposed model is demonstrated through a case study that carries out a detailed analysis for a variety of system parameters. The model is compared with the existing state-of-art model in a UAV scenario.

The rest of the paper is organized as follows. Section II presents a brief background about operator modeling and factors influencing its performance. Section III discusses PMC related details. Section IV presents the details of the proposed operator model. Section V demonstrates the usefulness of the proposed model with the help of a case study. Finally, a brief discussion is presented in Section VI followed by concluding remarks in Section VII.

II. RELATED WORK

The research on human factors and their utilization to create operator performance models is not new. Several studies, especially in the military scenarios, have been conducted after the Second World War [1]. Because of numerous benefits, human operator modeling is done from various research aspects, including computer science and engineering, psychology, computational biology, neuroscience, and behavioral sciences [18]. The following presents a brief background about human operators and factors influencing application-specific task performance.

A simple conceptual model of operator and system consists of three main stages. Operators take input from the system/environment, process the input, and perform one or more actions on the system based on the result of processing. Usually, human operators monitor visual or audio inputs from the system, which are inevitably affected by some noise. An operator performance in these systems is modeled using the signal detection model, which uses parameters such as signal detectability and operator bias [18]. These types of models have been used for studying several research problems.

Operator performance depends on a variety of factors and hence numerous research studies have concentrated on quantifying the influence of key factors on operator performance. One of the main factors that influence performance is operator experience. Hence, several research studies have been carried out to understand the effect of experience gained while training the operators. For instance, the authors in [28] study the impact of three levels of trained operators (control, video game player (VGP), and pilot) on eight cognitive performance tasks related to unmanned flight platforms. They find that pilots, who are the most experienced in the flight tasks, significantly outperform the other two groups on multi-attribute cognitive tasks. On the contrary, VGPs outperform the other groups on the tasks (such as visual target acquisition, identification, and tracking) where they have more experience. On the landing tasks, VGPs and pilots having equal experience show similar performance.

In addition to the experience gained in training, the age of an operator also plays an important role in performing cognitive tasks for human-automation interaction systems. The effects of automation error types and age on automation reliance are studied in [29]. Both young and old adults tend to over-use available automated aids as long as they are reliable. The authors noticed that older users took more time to adjust to the characteristics of automation. The authors in [30] emphasize the importance of understanding the impact of age on automation so that individuals are trained properly to interact with complex systems. In general, the authors found that younger adults (aged between 18 and 28) outperformed older adults (aged between 65 to 75) on the tasks.

The workload is another key factor that influences an operator's performance [30]. Moreover, the load condition in an operator's working environment varies and affects system objectives. As [16] examines the effect of workload on the performance of an operator and demonstrates that the operator's accuracy is considerably influenced by workload. They used a simulator developed by the Navy research lab, which accurately simulates the supervisory control tasks usually executed by UAV operators in a dynamic and unknown environment. They studied the performance of 81 Navy trainee pilots under different workloads. High workloads decreased the performance of operators in automated payloads and communications tasks. However, higher workloads improved operator performance in UAV routing tasks.

The authors in [31] highlight the link between personality variables with cognitive functions. In particular, [32] examines the effect of different task types on cognitive performance. The authors consider *updating* tasks as those requiring the capacity to maintain information in an active state and integrate new information, *set shifting* tasks referring to the ability to switch between different task goals, and *inhibition* tasks as the ability to ignore superfluous information. It was found that introverts overall perform better on set-shifting tasks, whereas extroverts perform better on inhibition and updating tasks.

In addition to the internal and work-related external parameters affecting an operator's performance, another critical parameter that heavily influences an operator is short breaks after certain intervals. The alertness of an operator can be increased via short activity breaks. Studies have shown that operators involved in vigilance tasks get improved alertness via short breaks in highly automated systems [33].

The aforementioned operator and system parameters are very helpful in analyzing the performance of SAS. Several techniques that utilize these parameters to analyze system performance are needed. Formal verification methods have been successfully used for validating and evaluating human-automation systems. Authors in [34] discuss approaches for formal modeling of human-automation interfaces and verification of properties associated with the behavior of the interfaces. Enhanced Operator Function Model (EOFM), an operator task modeling language, is devised to enable assessment of task analytic human behavior with formal methods. The EOFM task models are seamlessly ported into Symbolic Analysis Laboratory (SAL) language. These models can be blended with other manually generated models for a complete system model using asynchronous composition. The SAL's Symbolic Model Checker is used to formally verify specification properties.

PMC has been proven to be a promising candidate that can be used to verify such systems via quantitative modeling of human-automation interaction [8]. Moreover, temporal logic based formal languages (such as LTL) are equipped with rich temporal operators to capture fine system requirements [35]. PMC is used to find probabilities or expected rewards to verify whether the system meets the desired mission objectives. The authors in [8] use PMC to verify system design and assess its performance in a UAV mission. The system model utilizes insights from the literature related to operator characteristics. The authors study human-automation interactions and synthesize control protocols in a UAV mission.

Formal verification techniques are also used to verify systems where robots are managed remotely in a large wireless sensor networks (WSNs) [9]. Remote operators control robot movements and also help them in managing sensors. The human operator and robot are modeled as DTMC and MDP respectively. The authors analyze system performance for operators under various workload, fatigue, and other WSN constraints. The system uses operators to manage emergency services in a smart city and is verified using PMC in [36]. The authors use DTMC to model an HITL-based emergency management unit. They investigate response time and service availability under various operator constraints (such as skill level and workload) for various smart city areas. PMC is also used to check the system model with Security Operations Center (SOC) operators in [37]. The authors examine various skilled operator performance under different workloads to mitigate the security risk of critical services.

Although several attempts have been made to model operators in a semi-autonomous system, more generic, comprehensive, and extensible operator models that incorporate key human characteristics are needed [8]. In addition to incorporating key operator characteristics, this paper attempts to provide a framework to assess the performance of a semi-autonomous system and evaluates the proposed operator model for a realistic scenario from the literature. The following are the salient features of our work that distinguishes it from others:

1) The presented fatigue model penalizes the performance of an operator incrementally based on the number of tasks performed. Further, two separate thresholds are used to capture the starting point when an operator begins to fatigue and the highest point of maximum fatigue after which the performance plateaus.
2) The workload model captures a dynamic workload pattern that evolves probabilistically where an operator may have low, medium, or high workload with probabilistic updates depending upon the current load pattern of the working environment.
3) The task update model is incorporated in a way where an operator's performance may be penalized based on the associated cost due to switching to a different task.
4) In addition to incorporating several key parameters (such as age, experience, personality traits, etc.) the presented model captures the effect of breaks that may influence an application's objective positively as well as negatively. Short breaks do result in increased productivity but at the same time may cause a performance hit for time-sensitive applications. With respect to various time off values, we have presented the effect of four break policies on mission objectives in the UAV application.
5) Our work presents the components of the system in a modular way where various components such

as operator, working environment, and external variables interact to optimize the mission-specific objective function of an application. The system can be modified for a variety of case studies where each of the modules can be enhanced and its effect can be analyzed for a particular application. Some areas where our work can be used to assess the system's performance are: smart city emergency services [36], cybersecurity [37], and wireless sensor and robot networks [9].

III. MODEL CHECKING PRELIMINARIES

Formal verification techniques have been widely used to verify system designs in several applications. Model checking is a well-known formal verification technique that has been used as a validation technique in a variety of domains such as communication networks [38] and Biology [39]. Verifying a system using PMC mainly consists of:

- Creating an abstract representation of the system and its probabilistic evolution in the form of a state-transition diagram.
- Specifying the system properties that need to be verified in the form of probabilistic extension of temporal logic formulas.
- Exhaustively checking the state space of the model using model checker to see if the model satisfies the properties with a certain probability.

This section provides a brief background on probabilistic model checking, a formal verification technique to verify systems with stochastic behavior [40]. We first describe the MDP, one of the several models to capture probabilistic as well as nondeterministic system evolution. We then briefly explain Linear Temporal Logic (LTL) which is used to capture system properties in a property specification language. Finally, we introduce PRISM, a well-known probabilistic model checker, that uses a system model (expressed as MDP) and verifies it against given specifications (expressed in probabilistic temporal logic).

A. MARKOV DECISION PROCESS

MDP is one of the widely known models that are used to capture system behavior in reactive systems. In an MDP, system parameters are represented by the *states* of the system and the system evolution, triggered by an action chosen nondeterministically from a set of available actions at a state, is captured in discrete time via state *transitions*. If each state of the system evolves entirely based on some probability distribution, that is, there is at most one action available at each state, the state transition model is called discrete time Markov chain (DTMC). On the contrary, in an MDP a state may have more than one available action. In this case, probabilistic transitions based on an action's probability distribution can be made only after the action is chosen nondeterministically from a set of available actions. Nondeterminism is often resolved with the help of a scheduler (or controller) that chooses an action at a state based on a policy. Resolving nondeterminism at every state in an MDP transforms it into a DTMC.

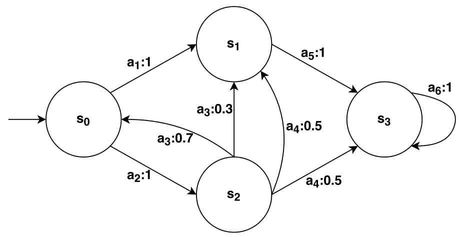

FIGURE 2. An example MDP.

Formally, an MDP is represented by a tuple $M = \langle S, s_0, A, \delta\rangle$ where S represents the set of states, s_0 represents the starting state, A represents the set of all actions in the system, and $\delta : S \times A \times S \rightarrow [0, 1]$ represents probability distribution for every state $s \in S$ and $a \in Act(s)$ such that $\sum_{s' \in S} \delta(s, a, s') = 1$. Here $Act(s) \subseteq A$ is the set of enabled actions at state s. The task of a scheduler is to break nondeterminism by choosing one of the possible actions from $Act(s)$. As mentioned above, for a DTMC $\forall s \in S, Act(s)$ is a singleton set.

Fig. 2 shows an example MDP with s_0 as the starting state. The rest of the parameters are as follows.

$$\begin{aligned}
S &= \{s_0, s_1, s_2, s_3\},\\
A &= \{a_1, a_2, a_3, a_4, a_5, a_6\},\\
\delta(s_0, a_1, s_1) &= \delta(s_0, a_2, s_2) = 1,\\
\delta(s_2, a_3, s_0) &= 0.7, \delta(s_2, a_3, s_1) = 0.3,\\
\delta(s_2, a_4, s_1) &= \delta(s_2, a_4, s_3) = 0.5,\\
\delta(s_1, a_5, s_3) &= \delta(s_3, a_6, s_3) = 1.
\end{aligned}$$

Here the sets of enabled actions are $Act(s_0) = \{a_1, a_2\}$, $Act(s_1) = \{a_5\}, Act(s_2) = \{a_3, a_4\}$, and $Act(s_3) = \{a_6\}$. Note that there is a nondeterministic choice between actions a_3 and a_4 at state s_2.

A path represents an execution of an MDP and is defined as a finite or infinite sequence of alternating states and chosen actions at that state. That is, if $s_i \in S$, $a_i \in Act(s_i)$, and $\delta(s_i, a_i, s_{i+1}) > 0$ then the following represents an infinite path starting at s_0:

$$\omega = s_0 \xrightarrow{a_0} s_1 \xrightarrow{a_1} s_2 \xrightarrow{a_2} s_3 \xrightarrow{a_3} s_4 \xrightarrow{a_4} \cdots$$

Path(s) and *Path*$_f(s)$ represent the sets of all infinite and finite paths starting at state s respectively. Let's denote the collection of all such paths from every state in S as *Path* and *Path*$_f$. That is:

$$Path = \bigcup_{s \in S} Path(s) \text{ and } Path_f = \bigcup_{s \in S} Path_f(s).$$

Let d be a discrete probability distribution function over a countable set A such that $\sum_{a \in A} d(a) = 1$. If $Dist(A)$ represents the set of distributions over A, a policy (or strategy) σ is defined as a function that, for a given path $w \in Path_f$, yields a distribution from enabled actions in the last state in w, $Act(last(w))$. That is,

$$\sigma : Path_f \rightarrow Dist(A)$$

For $a \notin Act(last(w)$, $\sigma(w)(a) = 0$. For instance, consider the following path:

$$w = s_0 \xrightarrow{a_0} s_1 \xrightarrow{a_1} s_2 \xrightarrow{a_2} s_3 \xrightarrow{a_3} s_4$$

such that $Act(last(w)) = Act(s_4) = \{a_5, a_6\}$, then σ breaks nondeterminism in the set of possible actions at s_4, that is, between a_4 and a_5. In other words, σ on an MDP M results in fully probabilistic DTMC M^σ. Probability space over infinite path in an induced DTMC can be defined using the standard concepts of measure and probability theory [41].

B. LINEAR TEMPORAL LOGIC (LTL)

Linear temporal logic (*LTL*) is an extension of classical propositional logic where time, as the modality, is used to reason about properties that are true in future states along a path [42]. Temporal logic provides a powerful way to capture a variety of qualitative properties in reactive systems [43]–[45]. If p represents an atomic proposition from the set of propositions AP, then the following grammar generates *LTL* formulas φ over AP.

$$\varphi ::= true|p|\neg\varphi|\varphi_1 \wedge \varphi_2|\mathsf{X}\varphi|\varphi_1 \mathsf{U} \varphi_2$$

Here X and U are temporal operators representing "next" and "until" respectively. Consider the tuple $M = \langle \mathbb{N}, I\rangle$, where I maps Natural numbers $\mathbb{N}$ to a set of propositions from 2^{AP}. An atomic proposition $p \in AP$ is true at a point 'i': $\langle M, i\rangle \models p$ if and only if $p \in I(i)$. Now, the semantics of each operator can be given as follows:

$$\begin{aligned}
&\langle M, i\rangle \models \neg\varphi \iff \langle M, i\rangle \not\models \varphi\\
&\langle M, i\rangle \models \varphi_1 \wedge \varphi_2 \iff \langle M, i\rangle \models \varphi_1 \text{ and } \langle M, i\rangle \models \varphi_2\\
&\langle M, i\rangle \models \mathsf{X}\varphi \iff \langle M, i+1\rangle \models \varphi\\
&\langle M, i\rangle \models \varphi_1 \mathsf{U} \varphi_2 \iff \exists j \geq i \wedge \langle M, j\rangle \models \varphi_2 \wedge\\
&\qquad\qquad \forall k(i \leq k < j) \implies \langle M, k\rangle \models \varphi_1
\end{aligned}$$

The semantics can be understood as follows. The formula φ is not true in the model M at state i ($\neg\varphi$), the formulas φ_1 and φ_2 are true in M at state i ($\varphi_1 \wedge \varphi_2$), the formula φ is true in the model M at the next state $i+1$ ($\mathsf{X}\varphi$), and the formula φ_1 remains true in the model M from state i to state $j-1$ until the formula φ_2 becomes true at state j ($\varphi_1 \mathsf{U} \varphi_2$).

C. PRISM–PROBABILISTIC MODEL CHECKER

Probabilistic model checking is a formal verification technique that has been used to verify systems that exhibit probabilistic behavior. PRISM is a well-known probabilistic model checker that has been successfully applied to analyze a variety of properties in several application domains [46]. PRISM supports a variety of probabilistic models, such as MDP, DTMC, and CTMC, to capture system behavior. These models are described in PRISM language where the system is often split into modules that can interact with each other. The overall system is constructed as the parallel compositions of each module. The properties that need to be verified are captured using system specification language that incorporates the probabilistic extension of temporal logic. The following are few example properties (taken from [26]) in PRISM specification language that can be verified by the probabilistic model checker:

- P>=1 [F *terminate*]: The algorithm terminates in future with probability 1.
- P=? [!*terminate_p*2 U *terminate_p*1]: The probability that process 1 terminates before process 2.
- P>0.99 [F (*request* & (X *ack*))]: The probability that, in future, a request is received followed by an immediate acknowledgement is greater than 99%.
- Rmin=? [x!=4 U x=1]: Finding minimum expected reward such that $x \neq 4$ in each state along the path until $x = 1$.

Given the model of a system M and the corresponding system specifications φ, PRISM can be used to either compute the probability with which M satisfies φ or the model can be extended with reward structure to compute reward-based properties.

IV. THE SYSTEM MODEL

Assuming that an operator working in an HITL-based system is performing several tasks. We define T as a set of task types or classes consisting of several application specific tasks such as vigilance task (t_v), visual detection task (t_d), and supervisory control task (t_c). Vigilance tasks represent the class of those tasks that require detecting simple infrequent signals over prolonged periods of time without rest. Visual detection tasks are those which need visual identification and classification. Supervisory control tasks are those where a single human operator oversees and intermittently interacts with multiple autonomous systems. Consider a set of missions $\Omega = \{m_1, m_2, m_3, \ldots\}$ where each m_i is a task-dependent mission defined by a mapping μ such that each $t_i \in T$ is mapped to zero or more elements in Ω, that is, $\mu : T \mapsto \mathcal{P}(\Omega)$. Here $\mathcal{P}(\Omega)$ represents the power set of Ω.

To capture the performance of an HITL-based system, we define the system as a tuple $S = \langle M_w, M_o, M_e\rangle$ where M_w captures work related parameters (such as workload and task types), M_o represents the operator model (capturing operator's characteristics), and M_e captures external parameters (such as *break* related updates in operator's performance). Each component of the system is described in the following subsections.

A. WORKING ENVIRONMENT MODEL

In addition to the operator related parameters, the performance of an operator also depends on work-related parameters. So we define a work model as a 3-tuple representing shift-type w_s, load update model M_l and task update model M_t. That is, $M_w = \langle w_s, M_l, M_t\rangle$. Here load and task update models are DTMCs that capture how workload and tasks change in a working environment.

1) LOAD UPDATE MODEL

An operator's workload condition either remains the same during a short time span with probability p' or changes to the next state with probability p''. Workload conditions may go to extreme (low to high or vice versa) with relatively smaller probability p'''. If p^w_{ij} represents the probability with which the operator transits from one workload state i to another state j, then we can write:

$$p^w_{ij} = \begin{cases} p' & (i = j = a_w) \text{ or} \\ & (i \neq j) \text{ and } (a_w = j) \\ p'' & (j = 2) \text{ and } (a_w = 1) \text{ or} \\ & (i = j), (j \in \{1, 3\}) \text{ and } (a_w = 2) \text{ or} \\ & (j = 2) \text{ and } (a_w = 3) \\ p''' & (j = 3) \text{ and } (a_w = 1) \text{ or} \\ & (i \neq j), (i, j \in \{1, 3\}) \text{ and } (a_w = 2) \text{ or} \\ & (j = 1) \text{ and } (a_w = 3) \\ (1 - p')/2 & (i = 2), (j \in \{1, 3\}) \text{ and } (a_w = 2) \end{cases} \tag{1}$$

Here $(p' > p'' > p''')$ such that

$$\forall i \in \{1, 2, 3\} : \sum_{j=1}^{|wl|} p_{ij} = 1 \tag{2}$$

where $|wl|$ is the total number of workload states. In order to avoid state space explosion, we have restricted $|wl|$ to three (low, medium, high) from five different grades (very low, low, medium, high, and very high) suggested in [21]. To get the values of these probabilities, one could perform statistical analysis based on past observations about the workload pattern in an application. For instance, [47] surveys such methodologies for workload modeling. Similarly, the authors in [48] use human subject data to estimate workload-related transition and observation probabilities. The state transition diagram corresponding to workload model M_l is shown in Fig. 3.

2) TASK UPDATE MODEL

The working environment of an operator may involve performing different types of tasks. We capture the task update model via a DTMC that represents the type of tasks assigned to the operator. The task type t_i may remain the same with probability p_i or may change with probability $(1-p_i)/(n-1)$ during an operation. Here n is the total number of possible task types. The corresponding task model M_t is shown in Fig. 4.

B. THE OPERATOR MODEL

The operator model M_o is a 3-tuple that represents many factors based on the operator's characteristics (attributes) A, operator's intrinsic features F, and task types T that operator performs. That is $M_o = \langle A, F, T \rangle$. A is a 2-tuple representing operator's long-term attributes A_{lt} as well as short-term

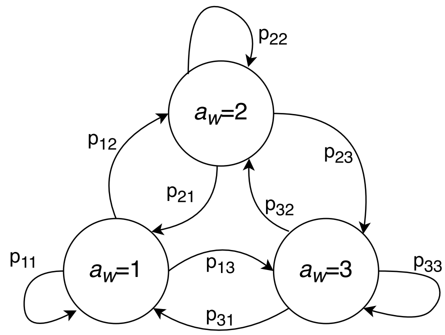

FIGURE 3. State transition diagram for workload model M_l.

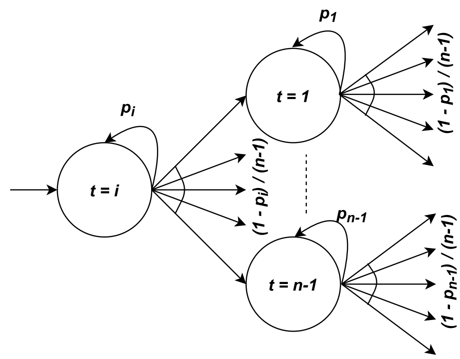

FIGURE 4. State transition diagram for task model M_t.

attributes A_{st}. That is, $A = \langle A_{lt}, A_{st} \rangle$. A_{lt} is a 2-tuple representing age a_a and experience a_e. That is, $A_{lt} = \langle a_a, a_e \rangle$. A_{st} is a 3-tuple representing sleep schedule a_s, base proficiency α, and fatigue Δ_f. That is, $A_{st} = \langle a_s, \alpha, \Delta_f \rangle$. Operator's intrinsic feature $f_i \in F$ where $F = \{f_{in}, f_{ex}\}$ is a set of two elements that represents whether the operator is introvert f_{in} or extrovert f_{ex}.

1) FATIGUE MODEL

Operator's fatigue Δ_f is captured as a function of discounts corresponding to sleep schedule, shift type, and an integer $k \in \mathbb{N}$ that models the number of tasks performed by the operator. The integer k increments by one for each completed task. The threshold τ_l measures the amount of work an operator can perform without being fatigued, whereas τ_h represents the high threshold after which operator performance plateaus by the discounted value δ_p. The fatigue discount due to the number of tasks performed is captured as follows:

$$\delta_f = \begin{cases} 1 & k \leq \tau_l \\ \delta_p & k \geq \tau_h \\ \delta_p + \dfrac{(1-\delta_p)\cdot(\tau_h - k - 1)}{\tau_h - \tau_l} & \text{otherwise} \end{cases} \tag{3}$$

That is, the operator does not get fatigued up to threshold τ_l, starts to get fatigued as k goes beyond τ_l, and reaches a maximum fatigue state at high threshold τ_h. Now, the overall discount factor due to fatigue can be given as:

$$\Delta_f = \delta_{ss} \cdot \delta_s \cdot \delta_f \tag{4}$$

where δ_{ss} and δ_s are discounts applied on the operator based on its sleep schedule a_s and shift type w_s respectively.

2) PROFICIENCY MODEL

Operator's base proficiency is modeled as the probability of accurately performing a task at $k = 0$ with low load condition (that is, α_l for a_w=1). Since proficiency decreases with increasing workload conditions, the operator's proficiency for various workload conditions is given as:

$$\gamma_w = \begin{cases} \alpha_l & \text{for } a_w = 1 \\ \alpha_m = (\delta_1^w \cdot \alpha_l), & \text{for } a_w = 2 \\ \alpha_h = (\delta_2^w \cdot \alpha_l), & \text{for } a_w = 3 \end{cases} \tag{5}$$

where δ_1^w and δ_2^w represent the discount factors for medium ($a_w = 2$) and high ($a_w = 3$) workload conditions respectively. Operator's proficiency based on age (γ_a) and experience (γ_e) are discounted similarly.

$$\gamma_a = \begin{cases} 1 & a_a = 1 \text{ (young)} \\ \delta_1^a & a_a = 2 \text{ (middle-aged)} \\ \delta_2^a & a_a = 3 \text{ (aged)} \end{cases} \tag{6}$$

and

$$\gamma_e = \begin{cases} 1 & a_e = 1 \text{ (untrained)} \\ \delta_1^e & a_e = 2 \text{ (amateur)} \\ \delta_2^e & a_e = 3 \text{ (trained)} \end{cases} \tag{7}$$

Since different operators tend to perform differently on a variety of tasks, this effect is captured by a discount γ_o applied as a compatibility based discount δ_o between operator type $f_i \in F$ and current task type $t_i \in T$:

$$\gamma_o = \begin{cases} 1 & f_i \text{ compatible with } t_i \\ \delta_o & \text{otherwise} \end{cases} \tag{8}$$

C. THE BREAK MODEL

One of the key design parameters that may influence operator performance is short breaks. M_e captures break as a tuple $\langle \gamma_b, \beta, \psi \rangle$ where γ_b represents break discount, β is break duration, and ψ represents break strategy. The operator may take a break after working for a while. In particular, it may take a break after τ_l which results in improved proficiency by a factor given as follows:

$$\gamma_b = \begin{cases} 0 & \tau_l \geq k \geq \tau_h \\ \dfrac{(\tau_h - k) \cdot (\gamma_b^{max} - 1)}{\tau_h - \tau_l} & \textit{otherwise} \end{cases} \tag{9}$$

That is, no break is taken when $k \leq \tau_l$ as the operator is not fatigued yet. At $k \geq \tau_h$ breaks are no more helpful as the operator is fully fatigued. Breaks are taken during these two extreme levels of activities such that discount value gradually decreases from its maximum value γ_b^{max}. Note that the fatigue and break are captured in such a way that the proficiency plateaus after some threshold. This ensures that the model is finite [8]. Now, the corresponding effective proficiency can be calculated as:

$$\Delta_p = \min\{\gamma_w, \gamma_w \cdot \gamma_a \cdot \gamma_e \cdot \gamma_o \cdot (1 + \gamma_b)\} \tag{10}$$

Algorithm 1 The System M

1: **procedure** HITL_SYSTEM
2: Init: $a \leftarrow \bot, \psi \leftarrow 0, p_b \leftarrow 0.5$ $\quad \triangleright \bot \overset{\text{def}}{=} \textit{false}$
3: Init: $s \leftarrow \bot, k \leftarrow 0, cwl \leftarrow 1, ct \leftarrow 1$ $\quad \triangleright \top \overset{\text{def}}{=} \textit{true}$
4: **while** *true* **do** $\quad \triangleright$ Continuously run for all epochs
5: **if** $\psi = 1$ **then** $a \leftarrow \bot$
6: **if** $\psi = 2$ **then** $a \leftarrow \top$
7: **if** $\psi = 3$ **then** $a \leftarrow \top$ with p_b,
8: $a \leftarrow \bot$ with $(1 - p_b)$
9: **if** $\psi = 4$ **then** $a \leftarrow$Nondet. $\triangleright$ resolved by PMC
10: $cwl \leftarrow M_l(cwl)$ $\quad \triangleright$ DTMC based cwl update
11: $ct \leftarrow M_t(ct)$ $\quad \triangleright$ DTMC based ct update
12: $s \leftarrow \top$ with $p = f(\Delta_f, \Delta_p)$
13: $\leftarrow \bot$ with $p = 1 - f(\Delta_f, \Delta_p)$
14: **if** $k \leq \tau_h$ **then** $k \leftarrow k + 1$
15: **if** $s = \bot$ **then go to** 5
16: Sync with application module
17: $epoch \leftarrow epoch + 1$
18: **if** $status(m \in \mu(ct)) = finish$ **then** break

D. THE HITL SYSTEM

The system model comprising of environment, operator, and break models, incorporates several internal and external parameters to assess application objectives. The overall working of the system is shown in Algorithm 1. Based on a break strategy ψ, the system chooses to take a break by setting the value of a (lines 5-6). A break is taken with probability p_b for stochastic break strategy (lines 7-8) but in the case of nondeterministic strategy (line 9) PMC solves the following Bellman's equation [49]:

$$v^*(s) = \max_a \{r(s, a) + \gamma \sum_{s' \in S} p(s' \mid s, a) v^*(s')\} \tag{11}$$

Here v^* is the optimal value of the reward r at state s for action a and S represents the set of all states in the MDP. γ is the discount on future rewards and $p(s' \mid s, a)$ represents the probability of moving to state s' from state s by taking action a. PMC can use value (or policy) iteration to solve this equation where v^* is computed iteratively until it converges. Now the optimal policy can be calculated as:

$$\pi^*(s) = \arg\max_a \{r(s, a) + \gamma \sum_{s' \in S} p(s' \mid s, a) v^*(s')\} \tag{12}$$

That is, the action is chosen that maximizes the reward. The current workload cwl and task type ct are updated based on their corresponding DTMCs M_l and M_t respectively (lines 10-11). These parameters serve as the input in finding the operator performance as a function of Δ_p and

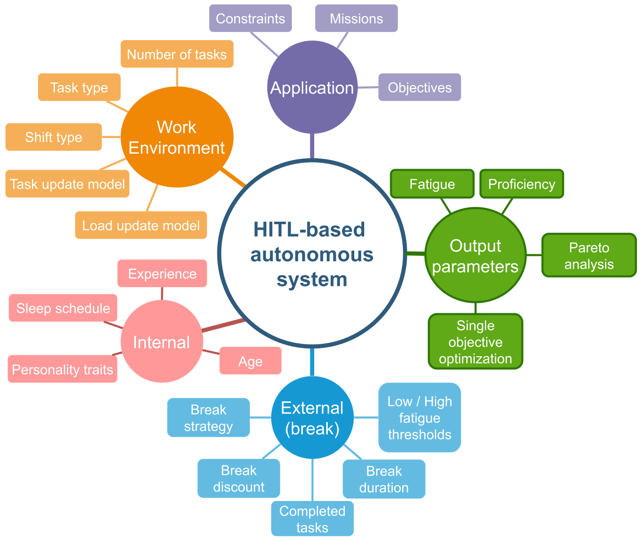

FIGURE 5. The HITL-based autonomous system.

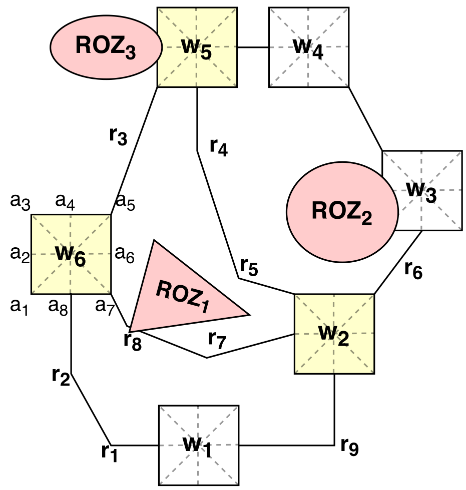

FIGURE 6. UAV flying zones for a mission (adapted from [8]).

TABLE 1. Symbols and their values.

Symbol	Meaning	Value range
a_a, a_e	Operator's age/experience	{1,2,3}
a_w	Current workload condition	{1,2,3}
f_i	Operator's type (introvert/extrovert)	{0,1}
$\alpha_l, \alpha_m, \alpha_h$	Accuracy at low, med, high workload	(0-1]
$\delta_i^w, \delta_i^a, \delta_i^e$	Discount for workload/age/experience	(0-1]
δ_f	Operator's fatigue discount	(0-1]
$\gamma_w, \gamma_a, \gamma_e$	Prof. discounts for workload/age/exp.	(0-1]
γ_o	Task-based discount for operator type	[0-1]
a_s	Operator's sleep schedule	{0,1}
p', p'', p'''	Prob. of changed workload	[0-1]
t_i	Type of the task	{1,···,n}
w_s	Shift type	{0,1,2}
δ_s, δ_{ss}	Discount for shift/sleep schedule	(0-1]
τ_l, τ_h	Low/High fatigue threshold	$\mathbb{N}$
δ_p	δ_f for $k \geq \tau_h$ where plateau occurs	(0-1)
γ_b	Increment factor due to break	[0-1)
k	Number of completed tasks	$\mathbb{N}$
β	Break duration	$\mathbb{N}$

Δ_f (lines 12-13). The parameter k maintains the number of tasks performed until it plateaus at τ_h (line 14). In case the operator didn't get expected results (s set to false), the algorithm starts over (line 15). Otherwise algorithm syncs with the application module in trying to achieve mission objectives (line 16-18). The overall system composed of various components is shown in Fig. 5 and the list of symbols used in the model is summarized in Table 1.

V. CASE STUDY: OPERATOR MANAGING A SEMI-AUTONOMOUS UNMANNED AERIAL VEHICLE (UAV)

To study the applicability of the proposed model in a HITL system, we consider a scenario given in [8] where a remotely managed UAV is sent for a surveillance mission. The surveillance area comprises of six waypoints (w_1 to w_6) where each waypoint has eight angle points (a_1 to a_8), which UAVs use to enter or exit. Road network is discretized into nine road points (r_1 to r_9). The corresponding UAV road network with the given waypoints, angle points, and road points are shown in Fig. 6. Here ROZs represent the zones where UAVs may get detected by an adversary. We have uploaded the PRISM code of this scenario as an open source in [50].

The operator primarily performs the task of maneuvering sensor controls to capture good quality images in any of the waypoints. If the quality of the image is not good, the operator may instruct the UAV to perform several loiters at each waypoint. Some waypoints (w_2, w_5, and w_6) are also called checkpoints where the operator may impact the selection of road points directly. For the human-automation system given in Fig. 6 the system consists of an operator with a varying number of input parameters and an environment in which the operator is performing a set of mission objectives for various break policies. In particular, the performance of an HITL-based UAV system is based on the following:

1) Operator Profile comprises of parameters related to age, experience, operator type, sleep pattern, and the corresponding discount factors. Operator performance is also affected by workload conditions under various task types. The operator's age and task type also help define the threshold when an operator starts to fatigue. In summary, an operator performance is captured by a tuple $P_o = \langle \gamma_a, \gamma_e, f_i, a_s, \gamma_w \rangle$.

2) Work Profile depends on various parameters related to company profile as well as task profile. In particular, work profile is a tuple $P_w = \langle t_{cpv}, w_s, M_l, M_t \rangle$ where $t_{cpv} \in T$ is cognitive task that involves psychomotor and visual skills; M_l and M_t are modeled as depicted in Figs. 3 and 4 respectively.

3) For the task t_{cpv}, we consider a set of missions given by $\mu(t_{cpv}) = \{m_1, m_2, m_2\}$ where m_1 represents UAV covering all six waypoints, m_2 represents UAV covering specific waypoints w_2, w_5 and w_6, and m_3 represents UAV covering arbitrarily chosen waypoints w_1, w_2 and w_6.

4) The main objective corresponding to the UAV missions is to complete the given mission in minimum time. Further, we consider another objective, that is, to minimize the number of ROZ visits. These mission objectives are in line with the ones given in [8].

5) We consider various break policies as external parameters that influence system performance. In particular, we consider fixed, stochastic, and nondeterministic break policies in this case study.

In the following subsections, we investigate how various operator, workload, and other related parameters influence UAV mission performance. We consider the operator presented in Feng's model [8] as the base model (referred to as FM) and use similar parameter values, wherever applicable, for comparison purposes. In general, for a particular application, the values of the corresponding parameters can be calculated from statistical analysis based on past experiences in that domain.

A. EFFECT OF INTERNAL AND EXTERNAL PARAMETERS

We study the effect of several operator and environment parameters on the mission goals of the application.

1) FATIGUE MODEL

The way fatigue is modeled affects the performance results of an operator. For instance, an operator fatigues differently on different types of work. Further, the model should capture fatigues based on the number of tasks performed. In particular, FM models operator fatigue as a step function, that is, fatigue jumps from 0 to high after a certain threshold. Our model, on the other hand, gradually increases the fatigue factor from low threshold τ_l to high threshold τ_h and plateaus thereafter.

We have used different operator's fatigue thresholds (τ_l, τ_h) to capture different operator-task related fatigue patterns. In particular, we have used three fatigue patterns corresponding to different types of operator-task interactions: T_1 where fatigue starts immediately but reaches to high threshold fast (0, 5); T_2 where fatigue starts immediately but increases at a slow pace (0, 10); and T_3 where fatigue starts after a certain amount of tasks but reaches maximum slowly (5, 10).

To demonstrate the effect of operator-task interaction and how it affects the fatigue model, we examine the mission completion time for different fatigue discounts as well as fatigue patterns for mission m_1. We compare the proposed fatigue discount model with FM. The corresponding reward structure used is:

$$R\{\text{``}time\text{''}\}min = ?[\mathsf{F}\ \ Status(m_i) = finish]$$

where m_i is the i^{th} mission, F is temporal operator for 'future', and *finish* is a predicate that is true when certain mission variables become true. "*time*" is defined as:

```
rewards "time"
[wait] true: 10;
[fly] true: 60;
endrewards
```

That is, loiter takes 10 time units and flying a road segment takes 60 time units. These time units may be adjusted for an application based on real-time values in minutes or seconds. The resulting plot for mission completion time is shown in Fig. 7. It can be observed from the figure that the results obtained using FM for T_1 and T_2 are exactly the same since the fatigue discount function in FM suddenly reaches its high value at τ_l. However, as mentioned earlier fatigue should affect an operator gradually. Further, different task types also affect an operator in different ways. For example, a task that needs lesser operator attention may fatigue an operator less than the task that is highly demanding. Therefore, less fatigue-prone tasks should help in faster mission completion. This reduced mission completion time can be observed in the figure for both fatigue patterns T_1 and T_2. FM also demonstrates this effect for T_3 where τ_l starts at 5. Finally, the figure shows that these effects are minimal at lower fatigue discount (higher δ_p values) as expected.

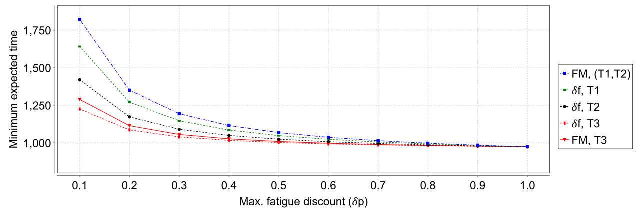

FIGURE 7. Mission completion time for various max. fatigue discount δ_p with α_h=0.5, α_l=0.9.

In the above analysis, we assumed that there is no limitation on mission completion time. However, not all applications are the same. Some of them require a quick and correct response from the operators. So, there is a limitation on the maximum time an operator has for mission completion. For this reason, we calculate the probability of mission completion for the three aforementioned missions m_1, m_2, and m_3. The reward structure used for calculating the probability of mission completion is as follows:

$$Pmax = ?[\mathsf{F} <= T\ \ Status(m_i) = finish]$$

The resulting plot for the probability of mission completion is shown in Fig. 8. The results show that modeling fatigue correctly not only captures mission objectives more realistically but also captures the influence of mission type on the mission objective. Further, the effect of the fatigue model on the probability of mission completion is more prominent when the mission is exhaustive (such as m_1).

2) WORKLOAD PATTERN

The load condition in which an operator is working influences the overall performance of the system. Besides, the load condition is generally not fixed but changes stochastically between different states. For instance, even under a predominantly low load environment, an operator may observe fleeting medium or high load patterns. We capture load patterns in such realistic environments by a DTMC M_l. In these environments, one is interested in calculating the time an operator takes to complete a mission on varying load conditions. We calculate mission completion time for two types of mission: intensive (m_1) and light (m_3), under three different load patterns: predominantly light, medium, and heavy. Workload

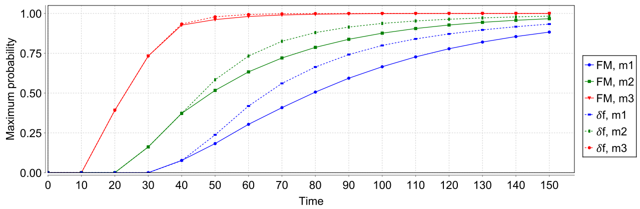

FIGURE 8. Probability of mission completion for three missions with α_h=0.5, α_l=0.9, τ_l=5, τ_h=20, δ_p=0.5.

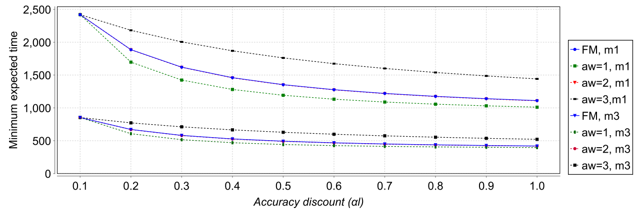

FIGURE 9. Mission completion time for various load patterns with δ_p=0.5. τ_l=5, τ_h=20 for m_1 and τ_h=10 for m_3.

states are updated with probabilities p'=0.7, p''=0.25, and p'''=0.05. The values of these probabilities are chosen based on existing studies such as [51]. Statistical analysis can be carried out on the load pattern in a specific working environment to obtain more precise values of these probabilities. The results are plotted in Fig. 9.

The figure shows that FM captures the working environment where the load pattern is medium. Our model approximates similar behavior for medium load patterns. However, our model captures the effect of high and low load patterns as well. Moreover, the figure shows that in addition to the load pattern, mission type also influences mission objectives as the effect of load pattern is more dominant for the high-intensity mission (m_1).

3) TRAINING AND AGE EFFECT

As discussed in Section II, training and age influence the performance of an operator. Further, mission objectives also play important role in determining the impact of the operator's parameters on its performance. To demonstrate the performance variations, we consider three types of operators: (1) FM, (2) young and untrained (YNE), and (3) aged and trained (OE). The operator is performing t_{cpv} task with M_l and M_t update models. We consider three aforementioned missions, m_1, m_2 and m_3, and calculate minimum mission completion time. In order to compare the effect of age and training with the base model FM, we use the same parameters and examine how mission completion time is affected with varying values of fatigue discounts δ_f.

The resulting plot for varying fatigue values is shown in Fig. 10. It can be seen that FM ignores the effect of operator related parameters since it does not cater age and experience factors. This is true for those applications where the fatigue factor is minimal with respect to age and experience. This can be seen in the figure as all the three operators show similar performance at high discounts values ($\delta_f = 0.9$).

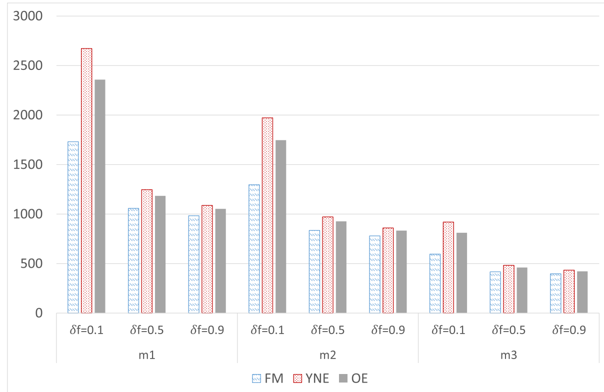

FIGURE 10. Mission completion time for three missions with varying fatigue discounts.

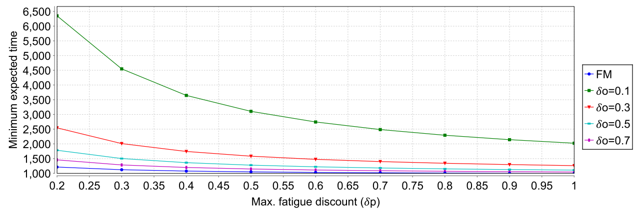

FIGURE 11. Operator-task effect on mission completion time for m_1 with α_h=0.5, α_l=0.9, τ_l=5, τ_h=20.

On the other hand, the performance of a realistic operator deteriorates with low values of fatigue discount, specially for intensive missions such as m_1.

4) EFFECT OF PERSONALITY TRAITS ON COGNITIVE FUNCTION

The proposed model allows a designer to compute mission objectives for operators with different personality traits. To examine this effect, we have considered operator-work relation to be matching (such as introverts working on set shifting tasks [32]) or mismatching. In case of a mismatch, we consider the discount factor γ_o described in Section IV to capture its effect on the mission objective. Fig. 11 shows mission completion time for intensive mission m_1 at four values of discount factors. From the figure, we can deduce that the effect of extreme operator-work incompatibility may impose severe constraints for mission completion time at low values of δ_p. For mission-critical applications with strict timing requirements, this additional constraint may render the application impractical.

B. BREAK POLICIES

Breaks help reduce workload by distributing the tasks in prolonged activity periods. In this section, we demonstrate how the proposed model can be used to scrutinize various break policies and their impact on mission objectives.

1) DETERMINISTIC BREAK POLICY

We consider a scenario where operators take mandatory breaks of fixed duration (β=30) in their working environment. Depending upon the types of the operator as well

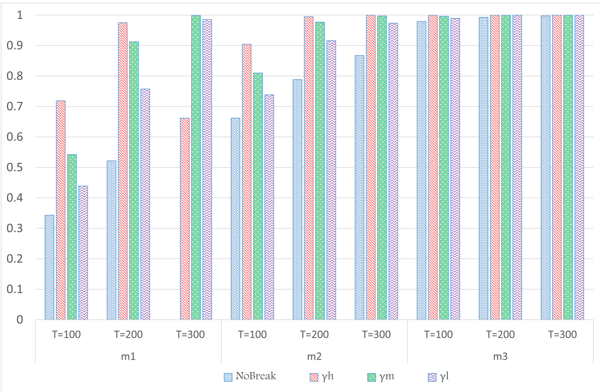

FIGURE 12. Probability of mission completion for break discounts.

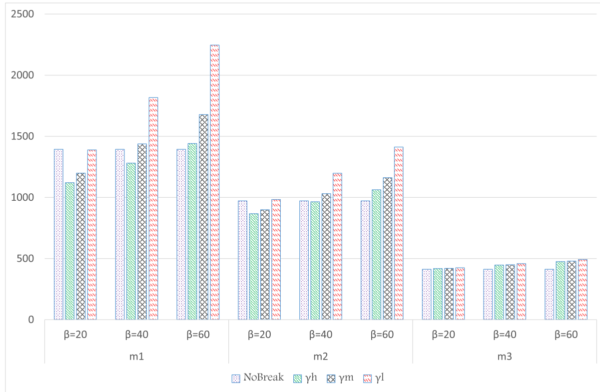

FIGURE 13. Mission completion time for different break discounts.

FIGURE 14. Mission completion time for different break policies at maximum break discount.

as how the break time is utilized, it may impact operator performance in a different way. Therefore, we consider three discount factors (γ_b^h, γ_b^m, γ_b^l) each capturing the impact of the time off. The high-impact break captures the situation where the operator gets maximum relaxation from the break and is modeled by resetting the value of k to 1. Similarly, medium-impact and low-impact breaks reset the value of k to τ_l and $(\tau_l + \tau_h)/2$ respectively. With these values of break discounts, we plot the effect of applying deterministic break policy in Fig. 12.

The figure plots the probability of success for missions m_1, m_2 and m_3 at three different maximum allowed mission completion times $T = 100, 200, 300$. It is clear from the figure that breaks improve the probability of mission success in all cases. The influence of time off is more prominent in the high-intensity mission (m_1). However, for a low-intensity mission (m_3), breaks impact mission completion for low threshold values of mission completion time. When time constraints are relaxed for less-intensity missions, it makes little difference whether a break is taken. It can be seen in the figure for m_3 at $T = 300$.

Although deterministic break policy increases the probability of mission success, the duration of break also influences mission completion time. On one hand, a short hiatus improves operator performance by reducing the fatigue level thereby increasing the probability of successfully completing the mission. However, the duration of the break may negatively impact mission completion in time-sensitive applications. To examine this effect, we plot mission completion time for various break duration β as shown in Fig. 13. The figure shows that short break duration (β=20) results in completing the mission faster than when no break is taken. However, depending upon the type of mission, prolonged break duration may negatively impact its completion. This is evident from the figure as m_1 is more negatively affected than m_3 at β=60.

In addition to a deterministic break policy, where either there is no break in the working environment or a mandatory break of fixed duration is taken, the next subsection studies additional break policies and their impact on mission completion.

2) NONDETERMINISTIC BREAK POLICY

This subsection considers two additional break policies deciding whether a break should be taken based on DTMC and MDP model. The DTMC-based break policy makes a probabilistic choice whether break needs to be taken, whereas the MDP-based policy models break choices nondeterministically. Fig. 14 compares four different break policies for a range of β values when the benefit of break is high (γ_b^h). With small values of β, the benefit of deterministic policy (that is, reduced fatigue resulting in faster mission completion) outweighs the cost incurred due to break duration. The figure shows that MDP resolves nondeterministic break choices much better than both aggressive choices (no break, deterministic) as well as probabilistic break choice (DTMC).

To examine the effect of break policies where the benefit of break discount is low (γ_b^l), we plot mission completion time in Fig 15. In the case where breaks do not offer high benefits (γ_b^l), all deterministic and probabilistic break policies result in high mission completion time except when β is very low. However, MDP-based break policy intelligently resolves nondeterminism and takes breaks appropriately to optimize mission objectives. After the value of β reaches a certain threshold, MDP eventually chooses to switch to a no-break policy (in this case, at β threshold value of 70).

So far we have evaluated the effect of break policies for different values of break discounts. However, depending upon

FIGURE 15. Mission completion time for different break policies at maximum break discount.

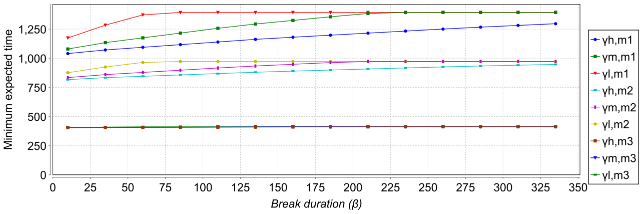

FIGURE 16. Mission completion time for different break discounts in MDP policy.

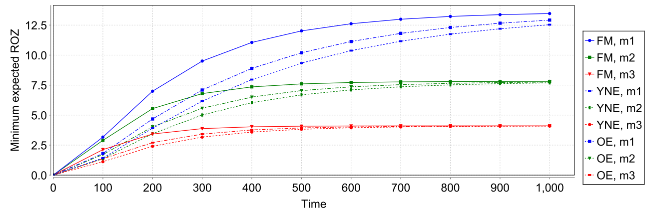

FIGURE 17. Minimum number of ROZ visits for various operators with τ_l=0, τ_h=0, α_h=0.5, α_l=0.9, δ_p=0.7.

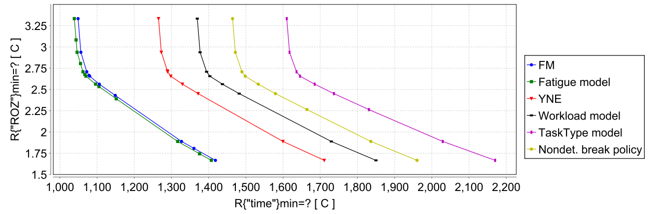

FIGURE 18. Pareto curves for two mission objectives: minimize mission completion time and minimize number of ROZ visits.

the type of mission, break discounts may have a different influence on mission objectives. For the MDP-based break policy, we can examine how the mission type is affecting the threshold for β after which break discount has no effect on the mission objective. The results obtained using the proposed model are shown in Fig. 16.

C. MISSION OBJECTIVES

The aforementioned analysis focuses on key objectives such as mission completion time and the probability of mission completion. However, the proposed model allows a designer to analyze other application-specific objectives as well. Moreover, compound objectives where more than one mission variable needs to be optimized can also be analyzed. The following subsection explains these points.

1) ROZ AVOIDANCE

Often a UAV mission demands that certain paths are avoided due to operation-related risks. In such missions, the objective is to visit a minimum number of ROZs rather than minimizing mission completion time. Therefore, we compute the minimum number of ROZ visits using the proposed model for operators FM, YNE, and OE. The obtained results are plotted in Fig. 17 for missions m_1, m_2, and m_3. The corresponding reward structure used is as follows:

$$R\{\text{“}ROZ\text{”}\}min =?[\mathsf{C} <= T]$$

It can be seen from the figure that the proposed model captures mission-specific objectives in terms of three valuable system parameters: 1) mission type, 2) operator type, and 3) the time duration in which these parameters have a meaningful effect on minimum ROZ avoidance. For instance, in mission m_3, the operator type becomes irrelevant after 700 time units if ROZ visits need to be minimized. A system designer can use these insights to select a proper operator for a given mission and time duration to achieve application-specific mission objectives.

2) PARETO CURVES FOR TWO MISSION OBJECTIVES

There are situations when more than one mission objective needs to be optimized. Here we use mission m_1 and consider multiobjective optimization with two competing objectives, that is, minimize the time to complete the mission and reduce the number of ROZ visits, for various fatigue and proficiency related parameters. This results in Pareto optimum, that is, a pair (x_1, x_2) of optimal values for two competing objectives obj_1 and obj_2 such that any value x_1 (or x_2) can not be further optimized without compromising the other x_2 (or x_1). The set of all such Pareto optima, called Pareto curve, are plotted in Fig. 18. The corresponding reward structure used for this purpose is as follows:

$$multi(R\{\text{“}time\text{”}\}min =?[\mathsf{C}], R\{\text{“}ROZ\text{”}\}min =?[\mathsf{C}])$$

Since the Pareto curve represents the optimal pair of values for the two objectives, taking the upward closure represents the area in which any pair of values is suboptimal and hence achievable.

The figure shows how the Pareto curve mostly shifts to the right implying increased mission completion time when we compare it with the base model (FM). The curve shifts left initially when we apply our fatigue model. The curve shifts to the right when the operator's specific parameters (age/experience) are taken into account as well as when work environment (workload and work type) models are used. Finally, the break model shifts the curve significantly to the left for a nondeterministic break policy with a discount value of γ_b^h. These results can be very helpful where a semi-autonomous system designer can select appropriate

Pareto optima according to given application constraints. For example, an impossible constrained multiobjective mission such as m_1 with the given constraints (#ROZ visits$\leq$2 and completion time$\leq$2000) can be achieved by choosing nondeterministic break policy.

VI. DISCUSSION

Considering the operator and environment parameters and constraints inherent in the system, designing a semi-autonomous system involves employing numerous scenarios to assess whether a system meets the desired objectives for various types of missions. The analysis of realistic scenarios can be carried out at the design phase before the actual deployment of the system. With the help of PMC, one can assess the behavior of such systems and perform various analyses for a variety of design parameters at an early stage. The presented work models an operator in a semi-autonomous system and provides a framework to assess various application-specific missions. The proposed model can be used for studying various operator characteristics, environmental configurations, and mission objectives. The operator work interaction and its impact on domain-specific missions in practical scenarios evolve. The proposed model can be adapted to analyse desired objectives for known as well as hypothetical future operator-work interactions.

To demonstrate the usefulness of the proposed approach, we have considered a semi-autonomous system consisting of a remotely operated UAV with certain mission objectives. With the help of the proposed methodology, we have assessed key mission objectives for several system parameters. Some of the key design questions we addressed are:

- What is the mission completion time in an application with various fatigue constraints.
- What is the probability that a certain mission can be completed in a given amount of time.
- How do the workload and task type affect mission completion time.
- Which break policy can optimize mission objectives.
- What are the possible optimal values where two competing objectives are optimized simultaneously.

Although the presented work incorporates key operator and working environment characteristics that can be used to assess scenarios in a variety of applications, several challenges remain. Below we discuss the important issues one should consider related to the applicability of our model.

1) Applicability of the model: We have presented a detailed case study to illustrate the applicability of our model. We posit that the model can be adapted for many applications where an operator helps a semi-autonomous system by providing feedback on the sensory output of the automated components. In such applications, the presented model can be used to assess various mission objectives. We have designed the operator-automation interaction in a modular fashion, where the operator interacts with the sensory output of the automation by module synchronization. In this way, from the application side, the domain-specific details may be updated separately without affecting the overall system behavior. Similarly, from the operator side, the parameters affecting the operator performance may be modified, without any changes in the application logic.
   There have been several studies in the literature that rely on such human-automation interaction. The authors in [52] survey HITL-based autonomous systems for infrastructure visual inspection and provide various levels of automation from level 1 (total human control) to level 8 (no human control). The survey shows that in such an application, most of the human-automation interaction is carried out by an operator providing feedback to the sensory output of the automation, as captured by the example scenario in this research work. However, our work is not directly applicable with trivial changes for those applications where human-automation interaction is intricate and can not be modularized in this way.
2) Domain-specific parameter values: To make our model applicable in various domains, the probabilities of discount parameters used in the model should be known. Various domain-specific studies have been carried out where one can obtain specific discount values for a particular domain in consideration. For example, [53] provides various insightful values for human fatigue-related parameters in the marine industry. The proposed model relies on such values/scores and the relevant domain knowledge to assess a given system.
3) Precision in result: Like other models, our results do not represent an ultimate precision. Rather, they serve as a guiding point for the system designer. Moreover, if operator parameters, working environment dependencies, mission definition, or objectives have changed, the proposed model can be used to assess their impact on the overall system with the new configurations.
4) Granularity of abstraction: When modeling such a wide-scope system it is important to tune the granularity of the abstraction so that each component of the system is captured at the desired level while abstracting away all the unnecessary details. This is essential not only to keep the focus on the analysis of the desired objectives but also to circumvent state-space explosion, a well-known problem in model checking. Many existing research studies analyze human fatigue at a detailed behavioral, physiological, and biological level. Although these results are of extreme importance, typically they are not directly used at the system level by managers working in HITL-based autonomous facilities. Our approach benefits from such research studies and facilitates a system designer to examine the overall system at a higher level of abstraction. However, the system designer needs to have sufficient domain

knowledge to understand the environment parameters where an operator is interacting with the automation. It is expected from a system designer to have situational and domain knowledge to effectively assess the productivity of the organization.

Deciding the right level of abstraction is an important design issue. The granularity of abstraction is the key that makes it possible to perform model checking in large and complex systems. The abstract model can be constructed directly from the high-level description of the system, even before the concrete model of the system is available.

5) Application-specific fine-tuning: The model does not capture a comprehensive set of application-specific fine-tuned parameters as it needs to maintain a certain level of abstraction to avoid the state space explosion problem in model checking. More expressive models may be needed in specific situations where an exhaustive set of system parameters are necessary to capture system evolution.

VII. CONCLUSION

We have presented a PMC-based approach to verify an HITL system for given specifications. Various components of the system, such as workload, task type, and operator characteristics are captured and their impact on given objectives is examined. The model also captures the effect of short breaks and the impact of associated policy on the missions. With the help of a detailed case study, we have presented how several critical parameters can help a designer assess whether the system would be able to meet the desired objectives with several constraints. With the help of Pareto curves, we have identified multi-objective optimization regions. We posit that, for a specific application scenario, the presented analysis could provide better understanding of human-automation interaction and guide an HITL-based system designer towards achieving optimal mission objectives.

ACKNOWLEDGMENT

The authors would like to thank the Prince Mohammad Bin Fahd Center for Futuristic Studies (PMFCFS) at Prince Mohammad Bin Fahd University for supporting this research. They would also like to thank Dr. Nazeeruddin Mohammad for valuable insights and discussions on case studies.

REFERENCES

[1] J. M. Bradshaw, R. R. Hoffman, M. Johnson, and D. D. Woods, "The seven deadly myths of 'autonomous systems,'" *IEEE Intell. Syst.*, vol. 28, no. 3, pp. 54–61, May 2013.

[2] A. Holzinger, "Interactive machine learning for health informatics: When do we need the human-in-the-loop?" *Brain Informat.*, vol. 3, no. 2, pp. 119–131, Jun. 2016.

[3] D. Xin, L. Ma, J. Liu, S. Macke, S. Song, and A. Parameswaran, "Accelerating human-in-the-loop machine learning: Challenges and opportunities," in *Proc. 2nd Workshop Data Manage. End-End Mach. Learn.*, Jun. 2018, pp. 1–4.

[4] F. Jotterand and C. Bosco, "Keeping the 'human in the loop' in the age of artificial intelligence," *Sci. Eng. Ethics*, vol. 26, no. 5, pp. 2455–2460, Oct. 2020.

[5] M. A. Mabrok, H. K. Mohamed, A.-H. Abdel-Aty, and A. S. Alzahrani, "Human models in human-in-the-loop control systems," *J. Intell. Fuzzy Syst.*, vol. 38, no. 3, pp. 2611–2622, Mar. 2020.

[6] M. Gil, M. Albert, J. Fons, and V. Pelechano, "Designing human-in-the-loop autonomous cyber-physical systems," *Int. J. Hum.-Comput. Stud.*, vol. 130, pp. 21–39, Oct. 2019.

[7] M. Cummings and D. Britton, "Regulating safety-critical autonomous systems: Past, present, and future perspectives," in *Living With Robots*. Amsterdam, The Netherlands: Elsevier, 2020, pp. 119–140.

[8] L. Feng, C. Wiltsche, L. Humphrey, and U. Topcu, "Synthesis of human-in-the-loop control protocols for autonomous systems," *IEEE Trans. Autom. Sci. Eng.*, vol. 13, no. 2, pp. 450–462, Apr. 2016.

[9] S. Muhammad, N. Mohammad, A. Bashar, and M. A. Khan, "Designing human assisted wireless sensor and robot networks using probabilistic model checking," *J. Intell. Robot. Syst.*, vol. 94, nos. 3–4, pp. 687–709, Jun. 2019.

[10] N. Ben-Asher and C. Gonzalez, "Effects of cyber security knowledge on attack detection," *Comput. Hum. Behav.*, vol. 48, pp. 51–61, Jul. 2015.

[11] R. O. Andrade and S. G. Yoo, "Cognitive security: A comprehensive study of cognitive science in cybersecurity," *J. Inf. Secur. Appl.*, vol. 48, Oct. 2019, Art. no. 102352.

[12] K. W. Williams, "Human factors implications of unmanned aircraft accidents: Flight-control problems," Federal Aviation Admin., Office Aerosp. Med., Washington, DC, USA, Tech. Rep. DOT/FAA/AM-06/08, 2006.

[13] N. H. T. S. Administration *et al.*, "Preliminary statement of policy concerning automated vehicles," Washington, DC, USA, 2013, p. 14, vol. 1.

[14] M. Huerta, *Integration of Civil Unmanned Aircraft Systems (UAS) in the National Airspace System (NAS) Roadmap*. Washington, DC, USA: Federal Aviation Administration, Dec. 2013.

[15] S. Zilberstein, "Building strong semi-autonomous systems," in *Proc. AAAI Conf. Artif. Intell.*, 2015, vol. 29, no. 1, pp. 1–5.

[16] S. M. Sherwood, "The effect of task load, automation reliability, and environment complexity on UAV supervisory control performance," Ph.D. dissertation, College Arts Sci., Embry-Riddle Aeronaut. Univ., Daytona Beach, FL, USA, 2018. [Online]. Available: https://commons.erau.edu/cgi/viewcontent.cgi?article=1433&context=edt

[17] M. Rasmussen and K. Laumann, "The evaluation of fatigue as a performance shaping factor in the petro-HRA method," *Rel. Eng. Syst. Saf.*, vol. 194, Feb. 2020, Art. no. 106187.

[18] N. Li, J. Huang, and Y. Feng, "Human performance modeling and its uncertainty factors affecting decision making: A survery," *Soft Comput.*, vol. 24, no. 4, pp. 2851–2871, Feb. 2020.

[19] S. Munir, J. A. Stankovic, C.-J. M. Liang, and S. Lin, "Cyber physical system challenges for human-in-the-loop control," in *Proc. 8th Int. Workshop Feedback Comput.*, 2013, pp. 1–4.

[20] H. T. Peng, F. Bouak, W. Wang, R. Chow, and O. Vartanian, "An improved model to predict performance under mental fatigue," *Ergonomics*, vol. 61, no. 7, pp. 988–1003, Jul. 2018.

[21] N. Li, H. Kong, Y. Ma, G. Gong, and W. Huai, "Human performance modeling for manufacturing based on an improved KNN algorithm," *Int. J. Adv. Manuf. Technol.*, vol. 84, no. 1, pp. 473–483, 2016.

[22] E. Seligman, T. Schubert, and M. A. K. Kumar, *Formal Verification: An Essential Toolkit for Modern VLSI Design*. San Mateo, CA, USA: Morgan Kaufmann, 2015.

[23] H. Chockler, A. Ivrii, A. Matsliah, S. Moran, and Z. Nevo, "Incremental formal verification of hardware," in *Proc. Formal Methods Comput.-Aided Design (FMCAD)*, 2011, pp. 135–143.

[24] A. Müller, S. Mitsch, W. Retschitzegger, W. Schwinger, and A. Platzer, "A component-based approach to hybrid systems safety verification," in *Proc. Int. Conf. Integr. Formal Methods*. Cham, Switzerland: Springer, 2016, pp. 441–456.

[25] M. Kwiatkowska, G. Norman, and D. Parker, "Advances and challenges of probabilistic model checking," in *Proc. 48th Annu. Allerton Conf. Commun., Control, Comput. (Allerton)*, Sep. 2010, pp. 1691–1698.

[26] *Prism Webpage*. Accessed: Jun. 1, 2021. [Online]. Available: https://www.prismmodelchecker.org

[27] M. Kwiatkowska, G. Norman, and D. Parker, "PRISM: Probabilistic model checking for performance and reliability analysis," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 36, no. 4, pp. 40–45, Mar. 2009.

[28] R. A. McKinley, L. K. McIntire, and M. A. Funke, "Operator selection for unmanned aerial systems: Comparing video game players and pilots," *Aviation, Space, Environ. Med.*, vol. 82, no. 6, pp. 635–642, Jun. 2011.

[29] J. Sanchez, W. A. Rogers, A. D. Fisk, and E. Rovira, "Understanding reliance on automation: Effects of error type, error distribution, age and experience," *Theor. Issues Ergonom. Sci.*, vol. 15, no. 2, pp. 134–160, Mar. 2014.

[30] S. E. McBride, "The effect of workload and age on compliance with and reliance on an automated system," Ph.D. dissertation, Georgia Inst. Technol., Atlanta, GA, USA, 2010.

[31] M. D. Lieberman, "Introversion and working memory: Central executive differences," *Personality Individual Differences*, vol. 28, no. 3, pp. 479–486, Mar. 2000.

[32] A. M. Campbell, D. B. Davalos, D. P. McCabe, and L. J. Troup, "Executive functions and extraversion," *Personality Individual Differences*, vol. 51, no. 6, pp. 720–725, Oct. 2011.

[33] M. M. Mallis, S. Banks, and D. F. Dinges, "Aircrew fatigue, sleep need and circadian rhythmicity," in *Human Factors in Aviation*. Amsterdam, The Netherlands: Elsevier, 2010, pp. 401–436.

[34] 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, May 2013.

[35] M. Z. Kwiatkowska, G. Norman, and D. Parker, "Quantitative analysis with the probabilistic model checker PRISM," *Elect. Notes Theor. Comput. Sci.*, vol. 153, no. 2, pp. 5–31, 2006.

[36] N. Mohammad, S. Muhammad, A. Bashar, and M. A. Khan, "Formal analysis of human-assisted smart city emergency services," *IEEE Access*, vol. 7, pp. 60376–60388, 2019.

[37] N. Mohammad, "A multi-tiered defense model for the security analysis of critical facilities in smart cities," *IEEE Access*, vol. 7, pp. 152585–152598, 2019.

[38] T. Kapus, "Using PRISM model checker as a validation tool for an analytical model of IEEE 802.15.4 networks," *Simul. Model. Pract. Theory*, vol. 77, pp. 367–378, Sep. 2017.

[39] O. Pârvu and D. Gilbert, "Automatic validation of computational models using pseudo-3D spatio-temporal model checking," *BMC Syst. Biol.*, vol. 8, no. 1, p. 124, Dec. 2014.

[40] C. Baier and J.-P. Katoen, *Principles of Model Checking*. Cambridge, MA, USA: MIT Press, 2008.

[41] E. M. Clarke, T. A. Henzinger, H. Veith, and R. Bloem, Eds., *Handbook of Model Checking*. Cham, Switzerland: Springer, 2018.

[42] A. Pnueli, "The temporal semantics of concurrent programs," *Theor. Comput. Sci.*, vol. 13, no. 1, pp. 45–60, 1981.

[43] B. Ramasubramanian, A. Clark, L. Bushnell, and R. Poovendran, "Secure control under partial observability with temporal logic constraints," in *Proc. Amer. Control Conf. (ACC)*, Jul. 2019, pp. 1181–1188.

[44] F. M. Favarò and J. H. Saleh, "Application of temporal logic for safety supervisory control and model-based hazard monitoring," *Rel. Eng. Syst. Saf.*, vol. 169, pp. 166–178, Jan. 2018.

[45] R. V. Cowlagi and Z. Zhang, "Route guidance for satisfying temporal logic specifications on aircraft motion," *J. Guid., Control, Dyn.*, vol. 40, no. 2, pp. 390–401, Feb. 2017.

[46] M. Kwiatkowska, G. Norman, and D. Parker, "PRISM 4.0: Verification of probabilistic real-time systems," in *Proc. 23rd Int. Conf. Comput. Aided Verification (CAV)*, in Lecture Notes in Computer Science, vol. 6806, G. Gopalakrishnan and S. Qadeer, Eds. Berlin, Germany: Springer, 2011, pp. 585–591.

[47] D. G. Feitelson, "Workload modeling for performance evaluation," in *Proc. IFIP Int. Symp. Comput. Perform. Modeling, Meas. Eval.* Berlin, Germany: Springer, 2002, pp. 114–141.

[48] K. Akash, K. Polson, T. Reid, and N. Jain, "Improving human-machine collaboration through transparency-based feedback—Part I: Human trust and workload model," *IFAC-PapersOnLine*, vol. 51, no. 34, pp. 315–321, 2019.

[49] R. Bellman, "A Markovian decision process," *Indiana Univ. Math. J.*, vol. 6, no. 4, pp. 679–684, 1957.

[50] S. Muhammad. (2021). *GitHub Link for HITL Code*. Accessed: Jun. 1, 2021. [Online]. Available: https://github.com/shahabuddinmuhammad/HITL-code

[51] V. Honsel, S. Herbold, and J. Grabowski, "Hidden Markov models for the prediction of developer involvement dynamics and workload," in *Proc. 12th Int. Conf. Predictive Models Data Anal. Softw. Eng.*, Sep. 2016, pp. 1–10.

[52] S. Agnisarman, S. Lopes, K. C. Madathil, K. Piratla, and A. Gramopadhye, "A survey of automation-enabled human-in-the-loop systems for infrastructure visual inspection," *Autom. Construct.*, vol. 97, pp. 52–76, Jan. 2019.

[53] N. Vagias, "A Bayesian network application for the prediction of human fatigue in the marine industry," Ph.D. dissertation, School Nav. Archit. Mar. Eng., Nat. Tech. Univ. Athens, Athens, Greece, 2010.

SHAHABUDDIN MUHAMMAD (Member, IEEE) received the B.E. degree from the NED University of Engineering and Technology, Pakistan, in 1999, and the Ph.D. degree from the University of Central Florida, Orlando, FL, USA, in 2007. He has extensive software development experience at various companies in USA, including Adaptec Inc., Orlando, FL, Riverbed Technologies Inc., San Francisco, CA, USA, and Microsoft, Redmond, WA, USA. He is currently working as an Assistant Professor with the College of Computer Engineering and Science, Prince Mohammad Bin Fahd University, Saudi Arabia. His research interests include probabilistic model checking, semi-autonomous systems, security protocols, mobile communication, and the Internet of Things.