CORE
Metadata, citation and similar papers at core.ac.uk
Provided by D-Scholarship@Pitt

2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
Daejeon Convention Center
October 9-14, 2016, Daejeon, Korea

Validation of Cognitive Models for Collaborative Hybrid Systems with Discrete Human Input

Abraham P. Vinod, Yuqing Tang, Meeko M. K. Oishi, Katia Sycara, Christian Lebiere and Michael Lewis

***Abstract*— We present a method to validate a cognitive model, based on the cognitive architecture ACT-R, in dynamic human-automation systems with discrete human input. We are inspired by the general problem of K-choice games as a proxy for many decision making applications in dynamical systems. We model the human as a Markovian controller based on gathered experimental data, that is, a non-deterministic control input with known likelihoods of control actions associated with certain configurations of the state-space. We use reachability analysis to predict the outcome of the resulting discrete-time stochastic hybrid system, in which the outcome is defined as a function of the system trajectory. We suggest that the resulting expected outcomes can be used to validate the cognitive model against actual human subject data. We apply our method to a two-choice game in which the human is tasked with maximizing net coverage of a robotic swarm that can operate under rendezvous or deployment dynamics. We validate the corresponding ACT-R cognitive model generated with the data from eight human subjects. The novelty of this work is 1) a method to compute expected outcome in a hybrid dynamical system with a Markov chain model of the human's discrete choice, and 2) application of this method to validation of cognitive models with a database of actual human subject data.**

***Index Terms*— Reachability analysis, Cognitive models, ACT-R, Human-in-the-loop systems, Markov controller, Hybrid systems, Human-automation interaction**

I. Introduction

Human-automation systems are ubiquitous, not only in commercial products, but also in complex and safety critical systems, such as power grid distribution systems, transportation systems, and biomedical devices. While extensive work has been done to assure safety and performance of fully automated systems, less is known about how to provide such assurances in human-in-the-loop systems. Indeed, human-automation interaction is often key for the overall performance of safety-critical human-automation systems. Recently, Google announced that 15 out of the 16 accidents involving Google's self-driving autonomous car were due to the other human drivers on the road [1]. Predicting expected behavior of human-automation systems is crucial for assuring safety as well as for optimizing performance.

A variety of methods have been considered to model the human in human-automation systems. Early models [2], [3], [4] for human drivers and pilots were based on transfer functions gathered from experimental data, and enabled characterization of pilot-induced oscillations and other problematic behaviors. However, in modern cyber-physical systems, human inputs may include non-trivial combinations of low-level continuous inputs as well as high-level discrete inputs (e.g., as in supervisory control schemes) [5], [6]. Recent investigations have explored models of high-level human decision making [7], [8], [9] and attention allocation [10] to facilitate design of decision support systems [11], [12] for dynamical systems.

An alternative approach makes use of a cognitive architecture by modeling human behavior according to known psychological phenomena. Computationally unified theories of cognition, known as cognitive architectures, have been developed to implement aspects of cognition that do not vary across human subjects, including the mechanisms and structures through which information is processed. They include limitations on attention and working memory, cognitive biases driven by the statistics of the environment (e.g., higher weight given to recent information) as well as statistical processes driving generalization such as similarity-based pattern matching. We have chosen ACT-R [13], [14] as our target cognitive architecture because 1) it has been used to develop a wide range of cognitive models, 2) it incorporates both cognitive capabilities and human limitations relevant to complex system control, and 3) it provides constraints on human performance through its mechanisms, e.g., memory retrieval and generalization. Such cognitive models are based on "first-principles" of human cognition and hence can be argued to be more general than e.g., experimentally obtained transfer function models for a specific subject.

In this paper, we use the framework of discrete-time stochastic hybrid systems to model human-in-the-loop cyber-physical systems [15] with discrete choice, and pose the question of expected outcome in terms of a stochastic reachability problem. We model the human as a non-deterministic Markov controller, generated from an ACT-R based cognitive model. We consider outcomes dependent on the execution of the hybrid system and formulate a forward reachability problem to predict expected behavior of the system. The resulting expected outcome can be compared against actual human subject data to validate a given cognitive model

This material is based upon work supported by the National Science Foundation. Oishi and Vinod are supported under Grant Number CMMI-1254990 (CAREER, Oishi) and CNS-1329878. Sycara, Ting, and Lebiere are supported under Grant Number CNS-1329986. Lewis is supported under Grant Number CNS-1329762. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

Abraham Vinod and Meeko Oishi are with Electrical and Computer Engineering, University of New Mexico, Albuquerque, NM; e-mail: aby.vinod@gmail.com, oishi@unm.edu (corresponding author)

Yuqing Tang and Katia Sycara are with Robotics Institute, Carnegie Mellon University, Pittsburg, PA; e-mail: {yuqing.tang,katia,cl}@cs.cmu.edu

Christian Lebiere is with Computer Science, Carnegie Mellon University, Pittsburgh, PA; e-mail: cl@cs.cmu.edu

Michael Lewis is with Information Sciences and Intelligent Systems, University of Pittsburgh, Pittsburgh, PA; email: ml@sis.pitt.edu

(reduced to a Markov chain controller). We apply this method to a two-choice game, conducted via Amazon Mechanical Turk [16], in which the human is tasked with maximizing net coverage of a robot swarm that can operate under rendezvous or deployment dynamics. The novelty of this work is 1) a method to compute expected outcome in a hybrid dynamical system with a nondeterminsitic Markov controller that describes the human input, and 2) an application of this method to validation of an ACT-R generated cognitive model using a database of actual human subject data.

The paper is organized as follows. We first describe the mathematical formalism of the human-in-the-loop hybrid system and formulate the problems to be solved in Section II. Section III describes the calculation of expected outcome and its comparison to experimental data for the general K-choice game model of the system. In Section IV, we apply this method to a robotic swarm problem and discuss validity of the cognitive model for this experiment. Lastly, we offer conclusions and directions for future work in Section V.

II. Problem Formulation

The following mathematical notation will be used throughout the paper. For some $m \in \mathbb{N}$ and a set S, let S^m denote the Cartesian product of S taken with itself m times. As in [17], for a sample space Ω, X is a random variable if X is a Borel-measurable transformation from Ω to $\mathbb{R}$. The random variable X is defined on a measurable space $\{\Omega, \mathcal{F}(\Omega)\}$ where $\mathcal{F}(\Omega)$ denotes the sigma-algebra associated with a countable sample space Ω. When $\Omega \subset \mathbb{R}^m$, we denote the sigma-algebra of Ω as $\mathcal{B}(\Omega)$, the Borel-space associated with Ω. A probability measure μ may be assigned to the measurable space to obtain a probability space $\{\Omega, \mathcal{F}(\Omega), \mu\}$. For a m-dimensional random vector $\boldsymbol{X} = \{X_1, X_2, \cdots, X_m\}$ with X_i defined in $\{\Omega, \mathcal{F}(\Omega), \mu\}$ for $1 \leq i \leq m$, $\boldsymbol{X}$ is defined in the product space $\{\Omega^m, \mathcal{F}(\mathcal{F}(\Omega)^m), \mu_p\}$. Here, μ_p is the joint probability of the random variables that make up the random vector $\boldsymbol{X}$.

A. Discrete-time stochastic hybrid dynamical systems with discrete human input

Consider a discrete-time time-invariant hybrid system

$$q[k+1] = \boldsymbol{g}(q[k], x[k], u_h[k]). \tag{1}$$

$$x[k+1] = \boldsymbol{f}(q[k+1], x[k], u_a[k]) \tag{2}$$

with discrete state $q[k] \in Q \subset \mathbb{N}$, continuous state $x[k] \in \mathbb{X} \subset \mathbb{R}^n$, automation input $u_a[k] \in \mathbb{U}_A \in \mathcal{B}(\mathbb{R}^p)$, and human input $u_h[k] \in \mathbb{U}_H \in \mathcal{F}(\mathbb{N}^m)$. Note that the automation input only affects the continuous state evolution, and the human input only affects the evolution of the discrete state. By definition, the discrete dynamics are propagated before the continuous state dynamics so that the human input at time k influences $x[k+1]$. Further, the input set $\mathbb{U}_H$ is a finite set. We require $\boldsymbol{f}$ and $\boldsymbol{g}$ to be Borel-measurable functions to ensure measurability of the states, as well as Lipschitz continuity of $\boldsymbol{f}$. The hybrid state space is defined as $S \triangleq \bigcup_{q \in Q}\{q\} \times \mathbb{X}$, and $\pi_a : S \to \mathbb{U}_A$ is the deterministic control policy used by the automation. The time horizon of interest is $N < \infty$.

The human input is modeled as an event (action) which causes the discrete state transitions as described by $\boldsymbol{g}$ (1). We further specify that the human input is modeled using a Markov chain, hence is defined as a stochastic map, $\pi_h : S \to \mathbb{U}_H$, such that

$$u_h[k] = \pi_h(s[k]) \tag{3}$$

where $s[k] = (q[k], x[k])$ and $u_h[k]$ is defined in the probability space $\{\mathbb{U}_H, \mathcal{F}(\mathbb{U}_H), Pr_{s[k]}\}$. Here, $Pr_{s[k]}$ is defined using the transition probabilities associated with the Markov chain modeling the human dependent on $s[k]$. We denote the control action sequence of the human and the automation derived from the control laws π_h and π_a, respectively, as

$$\boldsymbol{u}_h = \{u_h[0], \cdots, u_h[N-1]\} \in \mathbb{U}_H^N, \tag{4}$$

$$\boldsymbol{u}_a = \{u_a[0], \cdots, u_a[N-1]\} \in \mathbb{U}_A^N. \tag{5}$$

We can equivalently write (1), (2), (3) in standard form as a discrete-time stochastic hybrid system (DTSHS) [15],

$$\mathcal{H} = (Q, \mathbb{X}, \mathbb{U}_A, \mathbb{U}_H, T_x, T_q) \tag{6}$$

with the following elements:

- $Q \subset \mathbb{N}$ is the finite set of discrete states of the system.
- $\mathbb{X} \subset \mathbb{R}^n$ is the set of continuous states, $S \triangleq \bigcup_{q \in Q}\{q\} \times \mathbb{X}$.
- $\mathbb{U}_A \in \mathcal{B}(\mathbb{R}^m)$ is the compact control space for the automation, with deterministic controller, π_a.
- $\mathbb{U}_H \in \mathcal{F}(\mathbb{N}^p)$ is a finite set of choices for the human, generated using π_h (3).
- $T_x : S \times \mathbb{U}_A \times \mathcal{B}(\mathbb{X}) \to [0,1]$ is an impulse function:
$$T_x(x|s, u_a) = \delta\Big(x - \boldsymbol{f}\big(\boldsymbol{g}(s, \pi_h(s)), x, \pi_a(s)\big)\Big)$$
where $\delta(y) : \mathbb{R} \to \{0,1\}$ is 1 if $y = 0$ and is 0 otherwise, since $\boldsymbol{f}(\cdot)$ is a deterministic function.
- $T_q : S \times \mathbb{U}_H \times Q \to [0,1]$ is a function of the transition probabilities of π_h,
$$T_q(q|s, u_h) = T_q(q|s, \pi_h(s)) = Pr_s\{q = \boldsymbol{g}(s, \pi_h(s))\}$$
that captures the effect of human action, thereby "closing the loop" on the human.

The initial condition $s_0 = \{q[0], x[0]\} \in S$ is either known or may be sampled from a known probability distribution [15]. Note that restrictions, for example, to disallow human events under various circumstances, can be modeled in T_q.

For the system (6) with finite time horizon N, the hybrid time trajectory [18] is defined as a finite sequence of intervals $\tau = \{I_i\}_{i=0}^T$, such that 1) $I_i = \{\tau_i, \tau_i + 1, \ldots, \tau_i' - 1\}$ for $i < T$, 2) $\tau_T' = N$, and 3) $\tau_i < \tau_i' = \tau_{i+1}$. Since $\mathcal{H}$ is time-invariant, we assume $\tau_0 = 0$ without loss of generality. We denote the length of interval I_i by $|I_i|$, with $\Delta\tau = \{|I_i|\}_{i=0}^T$.

Example 1: Consider $\mathbb{U}_H = \{0, 1, 2\}$, $N = 6$ and $\hat{\boldsymbol{u}}_h = \{0, 2, 0, 0, 0, 1\}$. Then $T = 2$ and $\tau = \{I_0, I_1, I_2\}$, with $I_0 = \{0\}$, $I_1 = \{1, 2, 3, 4\}$, $I_2 = \{5\}$, and $\Delta\tau = \{1, 4, 1\}$.

The system $\mathcal{H}$ models human-automation system with a series of K-choice games, in which the human takes an action, modeled as a stochastic map $\pi_h(\cdot)$, at every τ_i. This framework is general, however we are primarily interested in systems in which human actions occur infrequently ($|I_i|$ is large), as in many semi-autonomous systems under supervisory control [19].

The hybrid system execution is $\boldsymbol{\xi}(\cdot; s_0, \boldsymbol{u}_h, \boldsymbol{u}_a) = \{\xi[1], \cdots, \xi[N]; s_0, \boldsymbol{u}_h, \boldsymbol{u}_a\}$, where for $k \in \{0, \cdots, N\}$,

$$\xi[k] = (\xi_q[k], \xi_x[k]) \in S. \tag{7}$$

Here, $\xi_q[k]$ is the discrete mode at the kth instant and $\xi_x[k]$ is the continuous state at the kth instant. Note that $\boldsymbol{u}_h$ is a N-dimensional random vector defined by (3). We define

$$\boldsymbol{\xi}_q(\cdot; s_0, \boldsymbol{u}_h, \boldsymbol{u}_a) \triangleq \{\xi_q[1], \cdots, \xi_q[N]\}, \tag{8}$$

where $\boldsymbol{\xi}_q(\cdot; s_0, \boldsymbol{u}_h, \boldsymbol{u}_a)$ is a N-dimensional random vector defined in the probability space $\{Q^N, \mathcal{F}(Q^N), \mathbb{P}_{s_0}\}$. Here, $\mathbb{P}_{s_0}$ refers to the joint probability of the discrete states $\xi_q[\cdot]$ for a given initial state $s_0 \in S$ [17]. Similarly,

$$\boldsymbol{\xi}_x(\cdot; s_0, \boldsymbol{u}_h, \boldsymbol{u}_a) \triangleq \{\xi_x[1], \cdots, \xi_x[N]\}, \tag{9}$$

and $\boldsymbol{\xi}_x(\cdot; s_0, \boldsymbol{u}_h, \boldsymbol{u}_a)$ is a N-dimensional random vector defined in $\{\mathbb{X}^N, \mathcal{B}(\mathbb{X}^N), \mathbb{P}_{s_0}\}$ by (2). From (3), the non-deterministic control action sequence $\boldsymbol{u}_h$ is also defined in $\{\mathbb{U}_H^N, \mathcal{F}(\mathbb{U}_H^N), \mathbb{P}_{s_0}\}$. For a given initial state s_0 and mode sequence $\boldsymbol{\xi}_q$, $\boldsymbol{\xi}_x$ becomes a deterministic sequence as given by (2). Further, given a Markov model for the human, the mode sequence $\boldsymbol{\xi}_q$ becomes completely determined by (1), (2), (3). Let $\hat{\boldsymbol{u}}_h$ be an instantiation of $\boldsymbol{u}_h = \pi_h(\cdot)$ that models the human, i.e, $\hat{\boldsymbol{u}}_h$ is a sequence of actions taken by the human. The corresponding trajectory $(\boldsymbol{q}, \boldsymbol{x})$ with

$$\boldsymbol{q} = \boldsymbol{\xi}_q(\cdot; s_0, \hat{\boldsymbol{u}}_h, \boldsymbol{u}_a), \ \boldsymbol{x} = \boldsymbol{\xi}_x(\cdot; s_0, \hat{\boldsymbol{u}}_h, \boldsymbol{u}_a), \tag{10}$$

has a likelihood of

$$\mathbb{P}_{s_0}\Big\{\boldsymbol{\xi}(\cdot; s_0, \boldsymbol{u}_h, \boldsymbol{u}_a) = (\boldsymbol{q}, \boldsymbol{x})\Big\} = \mathbb{P}_{s_0}\{\pi_h(\cdot) = \hat{\boldsymbol{u}}_h\} \tag{11}$$

since $\boldsymbol{g}$ is a deterministic function and

$$\begin{aligned}\mathbb{P}_{s_0}\{\pi_h(\cdot) = \hat{\boldsymbol{u}}_h\} &= \prod_{n=0}^{N-1} Pr_{\hat{s}[n]}\big\{\hat{u}_h[n] = \pi_h(\hat{s}[n])\big\} \qquad (12)\\ &= \prod_{n=0}^{N-1} Pr_{\hat{s}[n]}\big\{\hat{q}^+[n] = \boldsymbol{g}(\hat{s}[n], \pi_h(\hat{s}[n]))\big\}.\\ &= \prod_{n=0}^{N-1} T_q(\hat{q}[n+1], \hat{q}[n], \hat{x}[n], \pi_h(\hat{s}[n])).\end{aligned}$$

where $\boldsymbol{x} = \{\hat{x}[1], \cdots, \hat{x}[N]\}$, $\boldsymbol{q} = \{\hat{q}[1], \cdots, \hat{q}[N]\}$ and $\hat{q}^+[n] = \hat{q}[n+1]$. We discuss computing these probability measures in Subsection II-B.

We define a performance metric as a Borel-measurable reward function, $r : S^{N+1} \to \mathbb{R}$, which assigns a reward to every system execution. We also define a specific reward function, $R : S \times \mathbb{U}_H^N \to \mathbb{R}$, which captures the reward attainable from $s_0 \in S$ for the human action sequence $\boldsymbol{u}_h$. Since $\boldsymbol{u}_a$ is a known, deterministic automation control policy,

$$R(s_0, \boldsymbol{u}_h) = r(s_0, \boldsymbol{\xi}(\cdot; s_0, \boldsymbol{u}_h, \boldsymbol{u}_a)). \tag{13}$$

From (13), $R(s_0, \boldsymbol{u}_h)$ is a random variable defined in the probability space $\{\mathbb{R}, \mathcal{B}(\mathbb{R}), \mathbb{P}_{s_0}\}$, induced from the probability space defined for $\boldsymbol{u}_h$.

For the reward defined in (13), we define a trial of the K-choice game as an execution of $\mathcal{H}$ (6) in which the participant is presented with s_0, and the participant provides a strategy $\hat{\boldsymbol{u}}_h$ to maximize $R(s_0, \hat{\boldsymbol{u}}_h)$. We define an experiment as a collection of consecutive trials, and define training to be a series of trials completed prior to the experiment (so that the user is presumed to be fully trained at the start of the experiment). At the beginning of each trial, $\mathcal{H}$ is initialized from a hybrid state associated with the trial.

B. Cognitive models and the Markov controller

The cognitive model is implemented in a neurally-inspired cognitive architecture, ACT-R, via instance-based learning (IBL). Decisions in IBL are made primarily based on experiences of a task and its associated reward function. These experiences are stored as "chunks" of ACT-R's declarative memory, with each chunk corresponding to a relevant experience. Instance chunks typically contain a description of the context $s[k] \in S$ in which each decision is made, the decision itself, and the outcome of that decision. Important features $F(s[k])$ of the system influencing the decision are identified by the modeler to minimally express the context. For example, when dealing with the robot swarms, the features $F(s[k])$ could be the centroid and the dispersion of the initial swarm configuration [20].

The cognitive model ACT-R is initialized using prototypical parameters from human psychology that are representative of a typical response to the first trial in training. For every trial, ACT-R is provided with the same information presented to a human participant. The cognitive model extracts the features, then estimates the reward function (13) using ACT-R's blending mechanism [21]. The estimation is performed by partially matching the given features with the chunks representing instances stored in the ACT-R's declarative memory. The partial matching approach uses a linear similarity-based pattern matching function between the features and the activation of retrieval functions of ACT-R declarative memory based on recency and noise. The model then selects the decision that produces the largest reward. Finally, ACT-R compares the resulting reward to an internally generated estimate for these values and creates a new instance for the analyzed set of values for the features. This process is repeated for every trial faced by ACT-R similar to the experiments with the human. In short, ACT-R models a typical human who aims to maximize a reward function for a system given by (1), (2).

We create the Markov model of the human from simulated data gathered from ACT-R completing each experiment M times, where M is typically a large number (e.g., we used

$M = 1000$ in the robotic swarm game). The inherent stochasticity in ACT-R means that when faced with the same configuration, ACT-R may choose different actions in different trials. We denote the actions of ACT-R by $u_h^- \in \mathbb{U}_H$. We define the likelihood of ACT-R choosing a specific decision $u_h = \hat{u}_h$ for a given trial with state $s[k]$ as

$$Pr_{s[k]}\left\{u_h = \hat{u}_h\right\} = \sum_{j \in \mathbb{U}_H} Pr_{s[k]}\left\{u_h = \hat{u}_h \middle| u_h^- = j\right\} \times Pr_{s[k]}\left\{u_h^- = j\right\}. \quad (14)$$

The prior likelihood $Pr_{s[k]}\left\{u_h^- = j\right\}, j \in \mathbb{U}_H$, is simply the relatively frequency of action j in the M experiments. The conditioned likelihood $Pr_{s[k]}\left\{u_h = \hat{u}_h \middle| u_h^- = j\right\}$ is the relative frequency of action $\hat{u}_h$ when preceded by action j in the previous trial. Both of these expressions characterize the Markov chain associated with state $s[k]$ and are computed via batch processing of the data from the M experiments completed by ACT-R. Note that the Markov model is developed independently of the experiments involving the human participants.

Substituting (14) into (11) and (12), we obtain the probability of every trajectory $(\boldsymbol{q}, \boldsymbol{x})$ of (6) and $\mathbb{P}_{s_0}\left\{\pi_h(\cdot) = \hat{\boldsymbol{u}}_h\right\}$.

C. Problem statement

The following two problems are addressed in this paper.

Problem 1: Given a discrete-time stochastic hybrid system $\mathcal{H}$ (6) with discrete human input captured by a Markov model (3), and a performance metric R (13), find the expected performance of the system for a typical human subject.

We focus in particular on a class of reward functions that are functions of the forward reach tube because they provide a simplified solution to Problem 1. Once we have computed expected performance, we can pose the following validation problem for a given cognitive model.

Problem 2: Determine the validity of a cognitive model by comparing expected outcome using the cognitive model (Problem 1) with actual outcome from human subject experiments.

III. Computing Expected Outcome via Forward Reachability

A. Expected outcome

The performance for a typical human operator can be defined in terms of the expectation of the reward function (13)

$$\mathbb{E}_{s_0}\left[R(s_0, \pi_h(\cdot))\right] = \sum_{\hat{\boldsymbol{u}}_h \in \mathbb{U}_H^N} \Big[\mathbb{P}_{s_0}\Big\{\pi_h(\cdot) = \hat{\boldsymbol{u}}_h\Big\} \times r\big(s_0, \boldsymbol{\xi}(\cdot; s_0, \hat{\boldsymbol{u}}_h, \boldsymbol{u}_a)\big)\Big]. \quad (15)$$

While equation (15) theoretically solves Problem 1 for any $R(\cdot)$ that is a function of the system execution, numerical computation may be challenging for the most general case.

We consider the specific case when the reward function is restricted to be a function of the forward reach tube for a known human input,

$$R(N, \mathcal{S}_0, \hat{\boldsymbol{u}}_h) = J\big(ReachTube(N, \mathcal{S}_0, \hat{\boldsymbol{u}}_h)\big) \quad (16)$$

where $\mathcal{S}_0 \subseteq S$ is a set of initial configurations, $J : \mathcal{B}(S) \to \mathbb{R}$ is a set function defined on S, and the forward reach tube

$$ReachTube(t, \mathcal{S}_0, \hat{\boldsymbol{u}}_h) = \bigcup_{0 \le k \le t} Reach(k, \mathcal{S}_0, \hat{\boldsymbol{u}}_h) \subseteq S \quad (17)$$

is the union of forward reach sets over all time steps until $t \le N$. The forward reach set, $Reach(k, \mathcal{S}_0, \hat{\boldsymbol{u}}_h)$, consists of those states that can be reached at exactly time k,

$$Reach(k, \mathcal{S}_0, \hat{\boldsymbol{u}}_h) = \{y \in S : \exists s_0 \in \mathcal{I}, \text{ s.t. } \boldsymbol{\xi}(k; s_0, \hat{\boldsymbol{u}}_h, \boldsymbol{u}_a) = y\} \subseteq S \quad (18)$$

with $Reach(0, \mathcal{S}_0, \hat{\boldsymbol{u}}_h) = \mathcal{S}_0$.

While computation of (17), (18) is, in general nontrivial because all instantiations of $\boldsymbol{\xi}_q$ must be explored, the particular form of hybrid system we consider lends itself to realistic computation in specific cases.

First, we note that in the trivial case in which the initial set $\mathcal{I}$ is a singleton, computing (18) can be accomplished through simulation. Second, we can exploit the fact that in many systems, human input may occur infrequently.

Since the stochasticity in (6) is due solely to the human input, we consider portions of the trajectories (7) that result from a single human action, that is, that start when one human action occurs and end when the next human input occurs. For $I_i \in \tau$, let $\boldsymbol{v}_h(I_i) = \{v, 0, 0, \cdots, 0\} \in \mathbb{U}_h^{|I_i|}$ be an input sequence of length $|I_i|$ with some non-zero human input $v \in \mathbb{U}_H$ and $u_h = 0$ indicating that no human input occurs. Hence characterizing the reach set and reach tube between human-triggered events becomes a deterministic reachability problem that can exploit efficient computational tools [22], [23], [24], [25], [26] for certain classes of dynamics and initial sets.

For ease of notation, we define

$$Reach_{\text{auto}}(|I_i|, \mathcal{S}_0, v) = Reach(|I_i|, \mathcal{S}_0, \boldsymbol{v}_h(I_i)), \quad (19)$$

$$ReachTube_{\text{auto}}(|I_i|, \mathcal{S}_0, v) = ReachTube(|I_i|, \mathcal{S}_0, \boldsymbol{v}_h(I_i)). \quad (20)$$

B. Computing forward reach tube for systems with infrequent human actions

The forward reach tube $ReachTube(\cdot)$ for DTSHS (6) with infrequent human inputs can be computed using Algorithm 1. For a given human action sequence $\hat{\boldsymbol{u}}_h$, Algorithm 1 splits the execution of the hybrid system into segments in which the evolution occurs only under the action of automation. The deterministic reach tubes $ReachTube_{\text{auto}}(\cdot)$ in these segments can be computed efficiently using existing tools for deterministic reachability, and $ReachTube(\cdot)$ is computed by combining $ReachTube_{\text{auto}}(\cdot)$ using (17).

Algorithm 1 Computing the reach tube (17) for DTSHS with discrete human input (6) and occasional events

Input: $\mathcal{S}_0 \subseteq S$, $\Delta\tau$, $\hat{\boldsymbol{u}}_h$, $ReachTube_{\text{auto}}(dt, \mathcal{S}_0, v)$
Output: $RTube$ – the forward reach tube as defined in (17)

```
procedure ReachTubeCompute(S_0, û_h, Δτ)
    i ← 0                                   ▷ Initialize iteration variables
    RTube ← ∅
    RS_i ← S_0
    while i ≤ T do
        RT_i ← ReachTube_auto(|I_i|, RS_i, û_h[τ_i])   ▷ (20)
        RTube ← RTube ∪ RT_i                            ▷ (17)
        RS_{i+1} ← Reach_auto(|I_i|, RS_i, û_h[τ_i])   ▷ (19)
        i ← i + 1
    end while
end procedure
```

To demonstrate the use of Algorithm 1, consider Example 1. Recall that the hybrid time trajectory is $\tau = \{[0], [1, 2, 3, 4], [5]\}$ and the durations of the intervals is $\Delta\tau = \{1, 4, 1\}$. From (17), we compute the corresponding reach tube as $ReachTube(6, \mathcal{S}_0, \hat{\boldsymbol{u}}_h) = \cup_{0 \leq i \leq 2} RT_i$ with

$$RT_0 = ReachTube_{\text{auto}}(1, \mathcal{S}_0, 0), \tag{21}$$

$$RT_1 = ReachTube_{\text{auto}}(4, RS_0, 2) \tag{22}$$

$$RT_2 = ReachTube_{\text{auto}}(1, RS_1, 1) \tag{23}$$

The terminal reach sets of the first two time intervals are RS_0 and RS_1, with $RS_0 = Reach_{\text{auto}}(1, \mathcal{S}_0, 0)$ and $RS_1 = Reach_{\text{auto}}(4, RS_0, 2)$.

We now have all elements necessary for Problem 1, as well as computationally tractable solutions in some cases. The Markov controller for the human (3) is constructed from the cognitive model, and the corresponding reward function (16) is based on the forward reachability tube (computed using Algorithm 1). Hence the expected outcome can be computed for the cognitive model using (15), solving Problem 1.

C. Discussion regarding the assumptions used

For K-choice problems, Algorithm 1 and (15) solves Problem 1 under the assumption of 1) structure in the reward function, and 2) infrequent inputs from the human.

The reward function for the K-choice game is assumed to be a function of the reachable sets, as this allows for a simple solution to Problem 1. This assumption is fairly weak since the reach sets and tubes capture the execution of the system (6), and the reward functions are typically functions of the execution of the hybrid system. However, reward functions based on the safety of the system cannot be accommodated in this formulation since forward reachability analysis can not guarantee safety [27].

The assumption of infrequent human inputs allows us to solve Problem 1 using the existing tools in the literature for deterministic reachability analysis via Algorithm 1 and the computation of expected outcome using (15). Frequent human inputs would require validation based on stochastic reachability analysis which is computationally expensive [15].

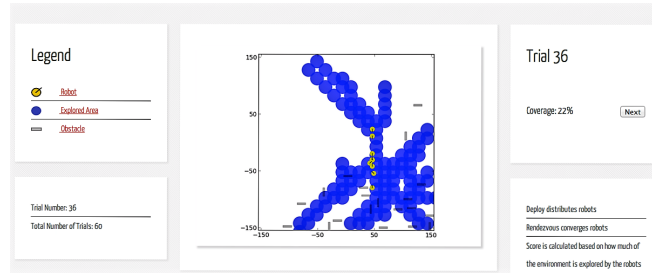


Fig. 1: User interface for the robotic swarm experiment [20].

D. Validation of the cognitive model

We solve Problem 2 by comparing the expected outcome (15) with the experimental mean of the rewards obtained during human trials,

$$\text{ExpMean}[R(s_0, \pi_h(\cdot))] = \frac{1}{N_{\text{obs}}} \sum_{i=1}^{N_{\text{obs}}} R(s_0, \pi_h(\cdot)) \tag{24}$$

where N_{obs} is the number of human participants. For each trial of the experiment, we analyze the difference

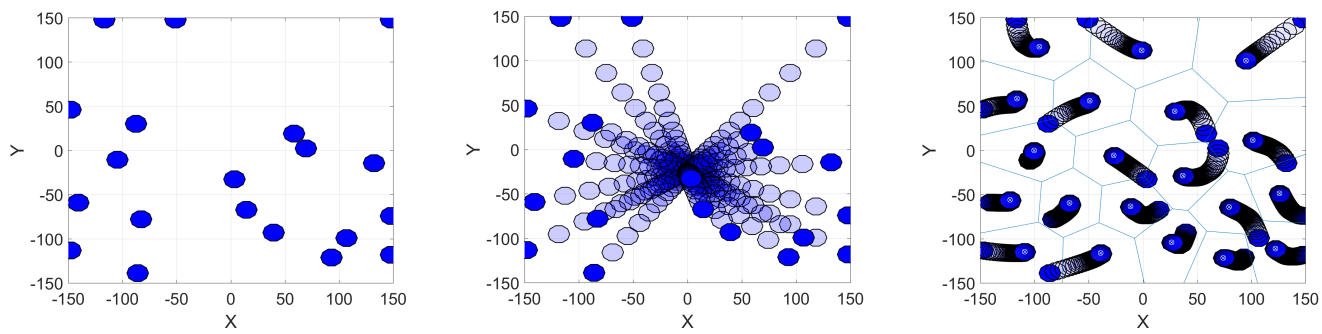
$$\epsilon(s_0) \triangleq \mathbb{E}_{s_0}[R(s_0, \pi_h(\cdot))] - \text{ExpMean}[R(s_0, \pi_h(\cdot))]. \tag{25}$$

The validation of the cognitive model is done by computing the mean (bias), the variance, and the maximum absolute value of $\epsilon(\cdot)$ over all initial configurations in the experiment. We also evaluate whether the expected outcome predicted by ACT-R is dissimilar to the expected outcome observed with human participants. This is evaluated via the Student's T-test, to assess the accuracy of the null hypothesis that the difference between expected outcomes (for experiments conducted by ACT-R or by the set of human subjects) is statistically significant. Evaluating whether these derived metrics are within the specifications of the problem completes the validation.

IV. Robotic Swarm Two-Choice Game

A. Experimental Setup

The robotic swarm two-choice game involves a swarm of twenty simulated point-mass robots [20]. The human controls the swarm indirectly by choosing between two strategies, which correspond to different robot dynamics: rendezvous or deploy. The rendezvous strategy causes the robots to converge to a common location, and the deploy strategy causes the robots to diverge to different locations in the game space. The simulated environment also contains a set of twenty fixed obstacles. Each robot has an associated sensing area, that is, a circular area (due to omni-directional sensors) that the robot can 'cover'. The participant's goal in each trial is to select the strategy that maximizes the accumulated area sensed. The only information that the participant has to make this decision is the initial configuration of the robots and obstacles. Additional details can be found in [20].



(a) (b) (c)

Fig. 2: (a) Initial configuration $x[0]$ of the robot swarm (without obstacles). (b) Obstacle-free simulation from $x[0]$ for $q[0] = \text{Rendezvous}$. (c) Obstacle-free simulation from $x[0]$ for $q[0] = \text{Deploy}$, with centroidal Voronoi tessellation of $x[N]$.

The game interface is shown in Figure 1. Each participant was presented with sixty trials. In each trial, the participant was shown the initial configuration and was asked to choose between the two strategies. In the first 10 trials (training trials), the outcomes using both the strategies were shown. In the remaining 50 trials, the participants were only shown the outcome for the strategy they selected. The configurations were chosen randomly. For the validation of the cognitive model, we ignore the configurations during training, as we are interested in comparing the cognitive model's performance to that of a typical trained human. A total of 50 subjects were recruited; 10 completed at least 15 trials of the experiment, while 8 completed all 50 trials in the experiment.

B. Hybrid human-in-the-loop system

We model the human-swarm system as the tuple $\mathcal{H} = (Q, \mathbb{X}, \mathbb{U}_A, \mathbb{U}_H, T_x, T_q)$, with

- $Q = \{\text{Rendezvous}, \text{Deploy}\}$,
- $\mathbb{X} = \mathcal{A}^{20}$, with $\mathcal{A} = [-150, 150]^2$ the simulated environment with 20 robots and the initial configuration denoted as $x[0] \in \mathbb{X}$,
- $\mathbb{U}_A \in \mathcal{B}(\mathbb{R}^{40})$, with feedback-based controller $\pi_a : S \to \mathbb{U}_A$,
- $\mathbb{U}_H = \{\text{Dep}, \text{Rend}\}$ that corresponds to selection of deploy or rendezvous, with the human's choice described by $\pi_h : \mathbb{X} \to \mathbb{U}_H$,
- $T_q(q|s) = \delta(q - g(\pi_h(x[0])))$ where $g(\cdot) : \mathbb{U}_H \to Q$ is Rendezvous if $\pi_h(x[0]) = \text{Rend}$ and is Deploy otherwise (and $q[k] = q[1]\ \forall k \geq 1$),
- $T_x : S \times \mathbb{U}_A \times \mathcal{B}(\mathbb{X}) \to [0, 1]$, described by

$$T_x(x|(q, x'), u_a) = \delta\big(x - \boldsymbol{f}(q[1], x', \pi_a(q[1], x'))\big).$$

We denote the time horizon by N, and for a given an initial configuration $x[0]$, denote the initial hybrid state as $s_0 = \{q[0], x[0]\}$. Note that T_q permits human input only at the start of the trial; once the participant has made this one decision, no further human input is allowed. Therefore, $\pi_h(\cdot)$ is a random variable and $\mathbb{P}_{s_0}\{\cdot\} = Pr_{s[0]}\{\cdot\}$ by (12). The index set of the robots is $\Gamma = \{1, 2, \cdots, 20\}$ and $\mathcal{O} \in \mathbb{X}$ is the set of the locations of the obstacles.

We define the position of the ith robot as $\boldsymbol{p}_i[k] = [x_i[k]\ y_i[k]]^\top \in \mathcal{A}$ and the continuous state of the swarm system as $x[k] \triangleq [\boldsymbol{p}_1[k], \boldsymbol{p}_2[k], \cdots, \boldsymbol{p}_{20}[k]] \in \mathbb{X}$. The dynamics for the ith robot in Deploy mode are

$$\boldsymbol{p}_i[k+1] = \boldsymbol{f}_{nl,i}(\text{Deploy}, x[k]) \tag{26}$$

$$= \boldsymbol{p}_i[k] + \alpha_1(\boldsymbol{c}_i[k] - \boldsymbol{p}_i[k]) + \alpha_2 \sum_{\boldsymbol{p}_{obs} \in \mathcal{O}} \frac{\boldsymbol{p}_i[k] - \boldsymbol{p}_{obs}[k]}{\|\boldsymbol{p}_i[k] - \boldsymbol{p}_{obs}[k]\|^2} \tag{27}$$

with $\boldsymbol{c}_i[k] \in \mathcal{A}$ the centroid of the Voronoi cell associated with i^{th} robot [28]. The dynamics in Rendezvous mode are

$$\boldsymbol{p}_i[k+1] = \boldsymbol{f}_{nl,i}(\text{Rendezvous}, x[k]) \tag{28}$$

$$= \boldsymbol{p}_i[k] + \frac{\alpha_3}{20}\left(\sum_{j \in (\Gamma \setminus \{i\})} (\boldsymbol{p}_j[k] - \boldsymbol{p}_i[k])\right) + \alpha_4 \sum_{\boldsymbol{p}_{obs} \in \mathcal{O}} \frac{\boldsymbol{p}_i[k] - \boldsymbol{p}_{obs}[k]}{\|\boldsymbol{p}_i[k] - \boldsymbol{p}_{obs}[k]\|^2} \tag{29}$$

with controller gains $\boldsymbol{\alpha} = \{\alpha_1, \alpha_2, \alpha_3, \alpha_4\} \in \mathbb{R}^4$. However, since obstacle locations are shown to not influence the participant's mode selection [20], for the purpose of our analysis, we ignore the obstacles (by setting $\alpha_2 = \alpha_4 = 0$) to obtain the simplified dynamics

$$\boldsymbol{f}_i(\text{Deploy}, x[k]) = \boldsymbol{p}_i[k] + \alpha_1(\boldsymbol{c}_i[k] - \boldsymbol{p}_i[k]) \tag{30}$$

$$\boldsymbol{f}_i(\text{Rendezvous}, x[k]) = \boldsymbol{p}_i[k] + \frac{\alpha_3}{20}\left(\sum_{j \in (\Gamma \setminus \{i\})} (\boldsymbol{p}_j[k] - \boldsymbol{p}_i[k])\right) \tag{31}$$

where $\boldsymbol{f}_i : Q \times \mathbb{X} \to \mathcal{A}$ and $i \in [1, 20]$. Note that $\boldsymbol{c}_i[k]$ makes $\boldsymbol{f}$ nonlinear in Deploy mode. Figure 2 shows the evolution of the system under Deploy and Rendezvous modes, generated using MPT [23], for a fixed initial condition.

A cognitive model was constructed using ACT-R. Two features, distance of the centroid of the initial position of the robots from the center of $\mathcal{A}$ (eccentricity), and its dispersion (sum of squares of the robot's distance from

centroid), characterize the initial configuration of the robots in each trial [20]. The cognitive model was developed as in Section II-B, then subjected to 60 trials, just as human participants (the first 10 trials constitute training ([20, Figure 4]); the remaining 50 trials constitute the experiment). ACT-R's blending mechanism is used to estimate the area coverage under either strategies [21] for each of these trials; the strategy with maximum accumulated coverage is selected. The model receives feedback from the environment (just as human participants) and uses it to update its estimates. Upon completion of a trial, the representation of the problem is added as a new chunk in ACT-R's declarative memory.

A Markov controller $\pi_h(x[0])$ was then abstracted from the cognitive model. The controller is dependent on the two features, rather than the swarm state. The possible values that the features can take was partitioned according to a sufficiently fine grid, so that 27 of the 50 configurations were distinct, meaning that they mapped to the different cells of the partition. The Markov chain associated with the cell corresponding to $x[0]$ characterizes the likelihood of the action taken by ACT-R through (14). The conditional and the prior probabilities in (14) were defined for the trials with non-distinct eccentricity and dispersion using data from $M = 1000$ repetitions of the experiment.

C. Computation of expected outcome and validation of the cognitive model

The performance metric of interest is the *accumulated* area coverage. Unlike the typical area coverage metric (see [28, Equation 2.3.1]), accumulated area coverage is the net area covered throughout the entire system execution (not just at a given instant) and depends on the initial configuration as well as the strategy chosen by the human. We define the performance metric as

$$R(s_0, u_h) = \frac{\mathrm{m}(SensedS(s_0, u_h))}{\mathrm{m}(\mathcal{A})} \tag{32}$$

with $u_h = \pi_h(s_0)$ and Lebesgue measure $\mathrm{m}(\cdot)$. The set of states sensed by the swarm $SensedS(s_0, u_h) \subseteq \mathcal{A}$ is

$$SensedS(s_0, u_h) = \{p \in \mathcal{A} : \exists h \in ReachTube(N, s_0, \boldsymbol{u}_h) \text{ s.t. } \|p - h\|_2 \leq c_s\} \tag{33}$$

with $\boldsymbol{u}_h = [u_h, 0, \ldots, 0] \in \mathbb{U}_H^N$ and omni-directional sensor radius $c_s > 0$. Using Algorithm 1, the forward reach tube is $ReachTube(N, s_0, \boldsymbol{u}_h) = ReachTube_{\text{auto}}(|I_0|, s_0, u_h)$ with $|I_0| = N$. We calculate $ReachTube_{\text{auto}}(N, s_0, u_h)$ via simulation since only a single initial configuration is considered. The expected outcome (15) for a trial with initial configuration $x[0]$ and initial hybrid state $s_0 = (q[0], x[0])$ is

$$\mathbb{E}_{s_0}[R(\cdot)] = \sum_{\hat{u}_h \in \mathbb{U}_H} \mathbb{P}_{s_0}\{\pi_h(x[0]) = \hat{u}_h\} R(s_0, \hat{u}_h). \tag{34}$$

Figure 3 shows that the model simulates the typical human performance well, with $|\epsilon(\cdot)| \leq 0.01$. The bias, variance, and maximum absolute difference of $\epsilon(\cdot)$ characterize relevant properties of the cognitive model: unbiasedness, mean square error, and the magnitude of error in the outliers, respectively.

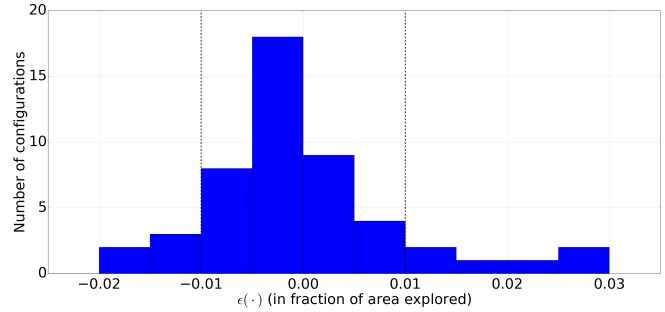


Fig. 3: Histogram of the prediction error $\epsilon(\cdot)$ (25). Most configurations (39 out of 50) have $|\epsilon(\cdot)| \leq 0.01$.

Mean (Bias)	Variance	Maximum absolute difference	Student's T-test (p-value)
1.06×10^{-4}	8.82×10^{-5}	0.028	0.9801

TABLE I: Statistical characterization of prediction error $\epsilon(\cdot)$ (25) across 50 configurations with 8 participants.

The bias and variance of $\epsilon(\cdot)$ across the configurations tested is quite small (Table I), indicating that that the cognitive model is unbiased with respect to the configurations considered and has negligible mean square error. The T-test between the experimental and predicted performance test showed that the null hypothesis ("means are equal") cannot be rejected (see Table I). While these metrics and the statistical test confirms that the validity of the cognitive model, the maximum absolute difference of approximately 0.028 points to the presence of certain configurations where the cognitive model may not predict the typical human performance within the tolerance limits. For these configurations, the features used may not be sufficient for the cognitive model to discern the optimal strategy.

We additionally investigated the possibility that our results may be influenced by configuration properties, namely, the distinctiveness of the outcomes under each strategy. We evaluate the possible correlation between the difference in reward under both strategies with the prediction error in Figure 4. With the coefficient of determination (R-squared) of 0.0034, no significant correlation exists. (Prototypical scenarios in Figures 4b and 4c show configurations with high and low error in prediction (25), respectively.) To the contrary, we suggest that configurations with many robots near the boundary of the environment are responsible for the outlier results.

V. Conclusions and Future Work

We model human-automation systems as discrete-time stochastic hybrid systems with stochastic discrete human inputs. We suggest a method to compute the expected outcome for reward functions based on reachable sets, and apply this method to the validation of a cognitive model for a robotic swarm problem. Our method is most computationally tractable for experiments with systems with LTI dynamics

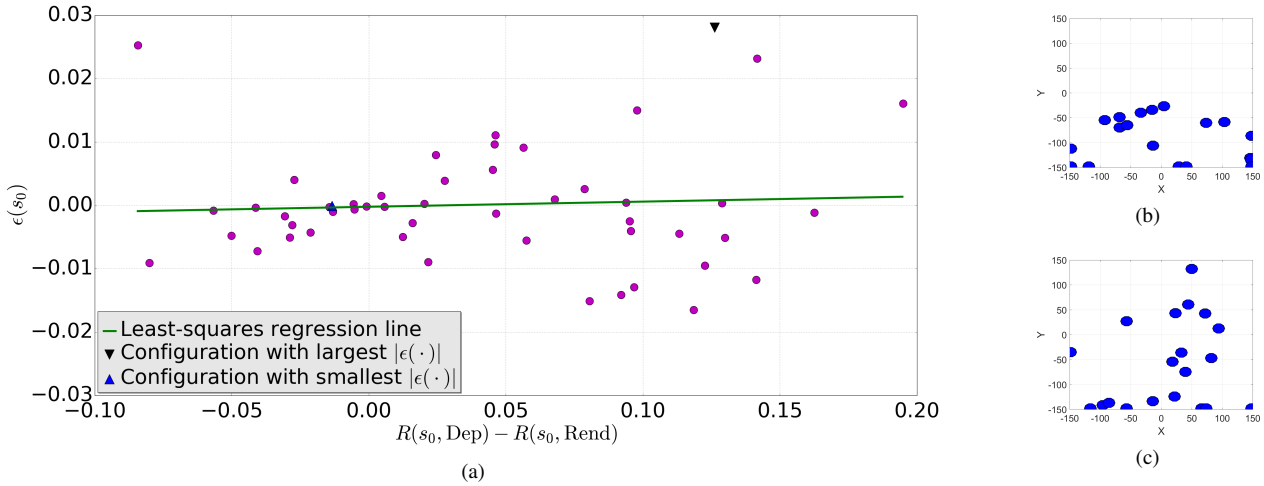


Fig. 4: (a) No significant correlation exists between prediction error $\epsilon(\cdot)$ (25) and the difference in reward under both strategies (R-squared value is 0.0034). (b) Configuration with largest $|\epsilon(\cdot)|$. (c) Configuration with smallest $|\epsilon(\cdot)|$.

and infrequent actions, which are modeled as a series of K-choice games.

In future work, we plan to consider reward functions requiring safety (viability) and safety with guarantees (reach-avoid), and to use the reachable set calculations to predict conflicts in collaborative human-automation systems. We also plan to identify additional features that could further reduce the prediction error.

References

[1] M. Richtel and C. Dougherty, "Google's driverless cars run into problem: Cars with drivers," September 2015. [Online]. Available: http://www.nytimes.com/2015/09/02/technology/personaltech/google-says-its-not-the-driverless-cars-fault-its-other-drivers.html?_r=0

[2] P. Fitts, *Human engineering for an effective air navigation and traffic control system*. Columbus, OH: Ohio State University Research Foundation, 1951.

[3] D. McRuer, "Human dynamics in man-machine systems," *Automatica*, vol. 16, no. 3, pp. 237–253, 1980.

[4] R. Hess, "Human-in-the-loop control," in *The Control Handbook*, W. Levine, Ed. CRC Press, Inc., 1996, pp. 1497–1505.

[5] J. Bailleiul, N. E. Leonard, and K. Morgansen, "Interaction dynamics: The interface of humans and smart machines," *Proceedings of the IEEE*, vol. 100, no. 3, pp. 567–570, 2012.

[6] M. Oishi, D. Tilbury, and C. J. Tomlin, "Guest editorial: Special section on human-centered automation," *IEEE Transactions on Automation Science and Engineering*, vol. 13, no. 1, pp. 4–6, January 2016.

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

[8] A. Stewart, M. Cao, A. Nedic, and N. E. Leonard, "Towards human-robot teams: Model based analysis of human decision making in two-alternative choice tasks with social feedback," *Proceedings of the IEEE*, vol. 100, no. 3, pp. 751–775, 2012.

[9] P. Reverdy and N. E. Leonard, "Parameter estimation in softmax decision-making models with linear objective functions," *IEEE Transactions on Automation Science and Engineering*, vol. 13, no. 1, pp. 54–67, January 2016.

[10] V. Srivastava, A. Surana, and F. Bullo, "Adaptive attention allocation in human-robot systems," in *Proceedings of the American Control Conference*, Montreal, Canada, June 2012, pp. 2767–2774.

[11] M. Forghani, J. M. McNew, D. Hoehener, and D. del Vecchio, "Design of driver-assist systems under probabilistic safety specifications near stop signs," *IEEE Transactions on Automation Science and Engineering*, vol. 13, no. 1, pp. 43–53, January 2016.

[12] X. Ding, M. Powers, M. Egerstedt, R. Young, and T. Balch, "Executive decision support," *IEEE Robotics and Automation Magazine*, vol. 16, no. 2, pp. 73–81, June 2009.

[13] J. R. Anderson and C. Lebiere, *The Atomic Components of Thought*. Lawrence Erlbaum Associates, Mahwah, NJ, 1998.

[14] J. R. Anderson, D. Bothell, M. D. Byrne, S. Douglass, C. Lebiere, and Y. Qin, "An integrated theory of the mind." *Psychological review*, vol. 111, no. 4, p. 1036, 2004.

[15] A. Abate, M. Prandini, J. Lygeros, and S. Sastry, "Probabilistic reachability and safety for controlled discrete time stochastic hybrid systems," *Automatica*, vol. 44, no. 11, pp. 2724–2734, 2008.

[16] Amazon Inc., "Amazon Mechanical Turk," 2016. [Online]. Available: https://www.mturk.com/mturk/welcome

[17] Y. S. Chow and H. Teicher, *Probability theory*. Springer, 1997.

[18] J. Lygeros, K. H. Johansson, S. Sastry, and M. Egerstedt, "On the existence of executions of hybrid automata," *IEEE Int'l Conference on Decision and Control, Phoenix, AZ*, pp. 2249–2254, 1999.

[19] R. Parasuraman, T. B. Sheridan, and C. D. Wickens, "A model for types and levels of human interaction with automation," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 30, no. 3, pp. 286–297, 2000.

[20] K. Sycara, C. Lebiere, Y. Pei, D. Morrison, Y. Tang, and M. Lewis, "Abstraction of analytical models from cognitive models of human control of robotic swarms," in *International Conference on Cognitive Modeling*, Groningen, The Netherlands, 2015.

[21] C. Lebiere, "The dynamics of cognitive arithmetic," *Kognitionswissenschaft*, vol. 8, no. 1, pp. 5–19, 1999.

[22] M. Althoff, O. Stursberg, and M. Buss, "Computing reachable sets of hybrid systems using a combination of zonotopes and polytopes," *Nonlinear Analysis: Hybrid Systems*, vol. 4, no. 2, pp. 233–249, 2010.

[23] M. Kvasnica, P. Grieder, and M. Baotić, "Multi-Parametric Toolbox 2.6.3 (MPT)," 2004. [Online]. Available: http://control.ee.ethz.ch/ mpt/

[24] A. B. Kurzhanski and P. Varaiya, "Ellipsoidal techniques for reachability analysis," *Hybrid Systems: Computation and Control*, pp. 202–214, 2000.

[25] C. Le Guernic and A. Girard, "Reachability analysis of linear systems using support functions," *Nonlinear Analysis: Hybrid Systems*, vol. 4, no. 2, pp. 250–262, 2010.

[26] I. M. Mitchell, "The flexible, extensible and efficient toolbox of level set methods," *Journal of Scientific Computing*, vol. 35, no. 2-3, pp. 300–329, 2008.

[27] ——, "Comparing forward and backward reachability as tools for safety analysis," *Hybrid Systems: Computation and Control*, pp. 428–443, 2007.

[28] F. Bullo, J. Cortés, and S. Martínez, *Distributed Control of Robotic Networks*, ser. Applied Mathematics Series. Princeton University Press, 2009.