Infotech@Aerospace 2012
19 - 21 June 2012, Garden Grove, California

AIAA 2012-2445

Human Intent Prediction Using Markov Decision Processes

Catharine L. R. McGhan[1], Ali Nasir[2], and Ella M. Atkins[3]
Autonomous Aerospace Systems Lab, University of Michigan, Ann Arbor, MI, 48109

This paper describes a system for modeling human task-level intent through the use of Markov Decision Processes (MDPs). To maintain safety and efficiency during physically-proximal human-robot collaboration, it is necessary for both human and robot to communicate or otherwise deconflict physical actions. Human-state aware robot intelligence is necessary to facilitate this. However, physical action deconfliction without explicit communication requires a robot to estimate a human (or robotic) companion's current action(s) and goal priorities, and then use this information to predict their intended future action sequence. Models tailored to a particular human can also enable online human intent prediction. We call the former a 'simulated human' model – one that is non-specific and generalized to statistical norms of human reaction obtained from human subject testing. The latter we call a 'human matching' model – one that attempts to produce the same output as a particular human subject, requiring online learning for improved accuracy. We propose the creation of 'simulated human' and 'human matching' models in this manuscript as a means for a robot to intelligently predict a human companion's intended future actions. We develop a Human Intent Prediction (HIP) system, which can model human choice, to satisfy these needs. This system, when given a time history of previous actions as input, predicts the most likely action a human agent will next make to a robot's task scheduling system. Our HIP system is applied to an intra-vehicle activity (IVA) space robotics application. We use data from preliminary human subject testing to formulate and populate our models in an offline learning process that illustrates how the models can adapt to better predict intent as new training data is incorporated.

Nomenclature

γ_i, φ_i, ϕ_i, α_i^j, β_i^j	=	known constants for all i and j
A	=	set of actions for the MDP, $A = \{1, 2, ..., n_a\}$
A^i	=	history of recently executed actions in state i, $A^i = \{a_1^i, a_2^i, ..., a_{n_h}^i\}, a_k^i \in A$.
$B_{z,k}$	=	$p(g_z^j = 1 \mid s^i, F^i, a_k)$, the probability of a goal objective g_z^j completing by execution of the action a_k and having high-priority interruptive goal flags F^i in state s^i
F^i	=	set of binary flags for high-priority interruptive goal states (on/off) in state i, $F^i = \{f_1^i, f_2^i, ..., f_{n_f}^i\}$
G^i	=	set of binary flags indicating goal status (complete/incomplete) in state i, $G^i = \{g_1^i, g_2^i, ..., g_{n_g}^i\}$
$H_{t,\, t+T}$	=	recorded time history of human actions from the last change in model policy
P_z	=	$p(g_z^j = 1 \mid A^i)$, the probability of a goal objective g_z^j completing given an action history A^i
$P_{z,x}$	=	$p(g_z^j = 1 \mid A^i, a_x)$, the probability of goal objective g_z^j being or becoming 1 (completed) due to occurrences of action a_x in action history A^i of state s^i
$R(s^i)$	=	reward function for state s^i

[1] Ph.D. Candidate, Aerospace Engineering Dept., University of Michigan, Ann Arbor, MI 48109, Student Member.
[2] Ph.D. Candidate, Aerospace Engineering Dept., University of Michigan, Ann Arbor, MI 48109, Student Member.
[3] Associate Professor, Aerospace Engineering Dept., University of Michigan, Ann Arbor, MI, Associate Fellow.

S = set of MDP states $S = \{s^1, s^2, ..., s^{n_s}\}$, where $s^i = \{G^i, A^i, F^i\}$

$T(s^i, a_k, s^j)$ = $p(s^j \mid s^i, a_k)$, probability of transitioning from state s^i to s^j by executing action a_k

I. Introduction

HUMAN subject testing has always been the gold standard for determining the operational characteristics of any system that requires human participation or interaction. However, computer simulation prior to extensive human subject testing is useful for running a high volume of repetitive tests without bias, exercising boundary cases without online risk to human actors, and various other advantages, at a greater rate in a shorter amount of time and for less monetary cost than human subject testing. Unfortunately, computer models generally only capture what we believe to be the most relevant features of the problem space we are attempting to define – in this case, human-robot interaction in space – and, as we cannot capture all elements of the real system, this incompleteness means that simulation is not a complete replacement for human subject testing. Yet, if we can create models populated with numerical parameters that have a basis in reality, then those simulations could be refined to be "good enough" to capture the most salient performance results and further refine our algorithms and procedures. Thus, a simulated system can be a useful evaluation tool on the road to system validation, especially to identify trouble areas and boundary cases that should be evaluated during later human subject testing in a physical environment, so long as one is careful to make those models as realistic as necessary to capture those problematic and failure cases. Use of preliminary test data to initially structure and populate offline-computed models is one method to do so.

There are many challenges that must be surmounted before intelligent robotic systems capable of supporting semi- and fully-autonomous human-robot interaction (HRI) mission operations are ready for on-orbit deployment. However, extensive testing tends to be long, costly, and does not cover the entire possible operational range of human interaction, especially for space environments, which are difficult to test on the ground even with carefully-constructed underwater testing scenarios. We believe that ground-based human subject testing can help advance the theory of HRI even in experimental settings; simple tests using common, familiar tasks such as eating food, drinking liquids, and simple computation are equally compelling across differing-gravity environments. Simulation studies using a computer model of human actions in an HRI context, which we call human intent prediction (HIP), could also help in the validation of the robotic decision-making algorithms used in HRI systems. Being a computer model, for HIP to be able to be useful tool requires that the space environment be well-characterized. The interaction scenario must also allow the specification of a "closed" (complete) action set as well as the specification of factors that lead the human to his/her choice of actions in this closed action set. Luckily, some space environments meet these constraints. Human subject testing on Earth is a valuable means to obtain model parameters for an HIP system for characterizing human behaviors in space, and learning will allow a decision-making system to adapt to a new space environment.

In order for a robot to be able to reason about a human and then make informed decisions about its own motion, however, it must first identify the human in the workspace, most commonly from visual data such as time-sequenced video imagery. Human action recognition – the process of sensing and determining a human's current state from observing their motion (in other words, the action they are taking) – is a difficult but solvable problem. Prior work has shown that this is possible using methods such as template matching and state-space models, the latter of which use a Hidden Markov Model (HMM) to identify gestures in real-time, a process also sometimes called "intention recognition".[1-7] Newer work in this area defines real-time techniques for 3D decomposition and reconstruction of explicit geometric poses and motion using regression methods,[8] as well as classification methods[9] for action recognition,[9] for which modified or expanded HMM methods may still be applied. With speed and computational increases in computing technology, HMM's use 3D pose information directly as their input.[9] Looking at these techniques more broadly, they supply not just the current state of the human, but also the known action state-space for the human. This information can be utilized by other decision-making algorithms.

For HRI, humans are generally treated as having random, unpredictable behaviors due to the uncertainty inherent in the human-sensing problem. However, humans generally act as rational agents, a precept that can be exploited given sufficient models of the human decision-making process. The action-state space at a cognitive level also tends to be reduced, when compared to the vast number of slightly varying pose trajectories that could be bundled together and considered the same motion, with only minimal changes in joint space, let alone larger deviations in motion that may be necessary for obstacle avoidance. Interestingly, though there has been a great deal of research done in human motion classification, there has not been much work done yet at a higher level that uses the output of these processes to predict the *next* most likely human action. There are two recent works that use Partially-Observable Markov Decision Processes (POMDP's) for this purpose. The first paper computes the current task

from observations of the human's actions given no explicit communication, but only shows results for a real physical case where the human has only two possible mission tasks to choose between.[10] The second paper also talks about a robot with a shared mission but without explicit communication and differing goals, and tests the accuracy of using extended MDP's to predict a human in simulation with two different simulated human policies, random-choice and closest-first. Results were encouraging when the simulated human's policy was not completely random and help motivate our choice of an MDP process for our work.[11] Neither paper addresses both using human subject experiment data to populate the models and multiple mission tasks, as we do.

A third paper[12] discusses the use of a POMDP by a slaved robotic system using explicit communication with a human to determine the next task it should accomplish; this system uses the POMDP to determine the task that the robot should perform next with strongest-belief of the human's preferences, and identifies when its own model of human preferences does not seem to match the observed human actions and it must explicitly request new information from the human. In our setup we allow the human to explicitly communicate their own task and state information to the robot when they like, but without prompting from the robot such that implicit communication is the default. However, we also assume that we do not need explicit communication from the human, and that the accuracy and reliability of such information, as well as the data conversion to a similar MDP state form, would be determined outside of our MDP framework as part of the sensory process – we only need to update the current MDP-determined policy with a new policy when the human preferences seem to have deviated far enough to require an update. Another paper[13] uses HMM's in a POMDP used to filter human dialog and human activity. Their paper focuses more on continued human direction of the slaved robot than collaboration where both robot and human are performing tasks with sporadic communication. Their setup uses agent actions as states and actions in what they call a "human activity recognition filter," but it takes human gestural actions as the state set, with the output being the robot's next explicitly-directed action, rather than acting as a pure human prediction module in the course of the robot making its own decisions as we intend.

In this paper, we define an "action" as a subtask (or subgoal) that may require multiple motions and manipulations to complete, and is short enough that we can assume that it will complete without interruption, but that would be considered a complete task at the cognitive layer. Mapping a sequence of actions to a larger task (or goal) requires knowledge of the sequence of events, which we assume to be more loose to allow for less constraints in the next sequence. For example, removing a panel would be considered a 'larger task' left to the scheduler to identify, while actions (subtasks) would include retrieving a screwdriver (for panel operation), removing a screw (one of four) (at a panel location), replacing a screwdriver back in a toolbox, moving a panel to the side, and so forth. Many of these actions could be considered more general, and further context would be determined by the scheduling process. However, the act of retrieving a screwdriver followed by the act of removing a screw at a panel location would indicate a higher probability that the next action would be to remove another screw at a panel location, rather than replacement of the screwdriver in the toolbox. As time-sequencing is an important part of this, we use a stochastic process – explicitly, an MDP – in order to compute this next most likely action.

The goal of this paper is build a decision-making scheme to predict the next intended action of a human, given a known history of actions, with a high degree of accuracy. The framework presented here could ultimately be used by a robot in for automated scheduling of activities in a physically-proximal human-robot collaboration environment. We assume that the human's actions and their decision-making process leading to the selection of actions (their goal-objectives) are fully-observable, which allows us to use an MDP rather than a POMDP for computational tractability. As our domain is a space environment, the assumption of full observability is valid because the human astronaut is operating in a highly-rehearsed HRI scenario such as satellite servicing. This is a good first step towards being able to deal with a more complicated partially-observable situation where we relax that assumption. For now, we are more interested in being able to use this model to evaluate HRI robot task-scheduling methods. This would necessarily assume known inputs, either through environmental sensing or through the explicit selection of simulation test case studies. The inclusion of partially-observable states under these circumstances would be counter-productive. Previous ground-based testing has provided evidence that in cases where the closed-world assumption holds, human-robot collaboration will be more efficient when explicit communication is not used. Explicit communication, via speech, gestures, or similar methods, requires a mandatory minimum workload for humans[14] while implicit communication, via passive visual and audible observation of physical motions, can be exploited for useful information with the benefit of requiring no additional overhead from a collaborating human. For a 'human matching' model attempting to predict a particular human's next action(s), we use implicit communication – passive visual sensing of physical motion and gaze – as the primary source of sensory information to estimate and determine a human's ongoing action responses. Explicit communication is not required, and not exploited nearly so much as the data extracted from implicit communication, but for completeness we include a

mechanism that allows infrequent, sporadic verbal informational updates from the human to be taken into account in the learning and update process after evaluation and filtering.

We explain our motivating example – the prediction of a human's motion in a workspace shared with a robotic agent in a space environment: an astronaut on IVA inside a pressurized spacecraft, sitting in front of a computer console with food and drink nearby so they may keep up their energy level as they work on computer tasks and keep an eye on a nearby experiment that may require some upkeep. This environment lends itself well to human intent prediction (HIP), as the motion prediction element complexity is reduced due to physical constraints imposed by the seating device to secure the human in space in zero-g. This is not as restrictive as an on-orbit EVA where a spacesuit is required, nor would tasks require as much time or energy expenditure for the human to complete. However, we still believe it a good assumption that random or otherwise unmodeled motions and actions that do not contribute to task completion will not occur in the IVA case, similar to the EVA case. We then discuss an analogue ground-based environment: a laboratory experimental setting that captures similar related motion-primitives to those one would expect in the primary example case. Note that we assume the robot using the MDP takes no actions that would influence the human directly; hence we assume that there is no feedback where the human might be reasoning about the robot's actions as part of their own goal-scheduling or task-based planning process. This assumption simplifies our model but later must be relaxed to include the impact of robot actions on human intent. Below, we formulate the human intent prediction (HIP) problem, describe the solution approach using an MDP formulation, and present a series of case studies.

II. Problem Statement

Efficient operation of a collaborative human-robot team requires that the robot be capable of predicting a human companion's intent. We hypothesize that a robot can predict companion intent by first identifying actions based on observations rather than relying on explicit communication, then recognizing those observed actions as part of a sequence. Structured action sequences would be expected to be known for applications such as human-robot collaboration on EVA. Intent prediction is also possible for IVA collaboration, where action sequences are less certain but long-term observation of a human companion's behavior can inform prediction of action sequences.

We make the closed-world assumption, and assume that it is valid. We assume perfect observation of human action, enabling us to assume full observability, and we assume the robot has sufficient memory to store an n-action history where n is finite but potentially large. We assume that human subject data exists for learning model parameters, or that a process exists for observing the human to iteratively improve parameter estimates. We assume for the 'human matching' model that, prior to the start of HRI, we have a pre-existing initial baseline model of human preferences and uncertainties in performing the most preferred action and an online database of similar MDP policies to search.

Under these assumptions, we present a framework to exploit the human action models in a manner that enables the robot to predict a companion's next action without explicit communication. For this purpose, we use the MDP to generate an optimal policy for human intent prediction with respect to the current known human preferences and activities, which is also optimal (having a high level of accuracy in its output) under the constraint of real-time operation, given a known history of human actions and goal-objectives as input. To accurately predict the next action of a specific human, we update the choice of model parameters used through learning logic procedures that evaluate observed behaviors using implicitly communicated human action (state) data.

III. Solution Architecture

Figure 1 shows our architecture for human intent prediction (HIP). A Markov Decision Process (MDP) is the central decision-making algorithm with collaboration-based parameter learning used to improve the MDP model.

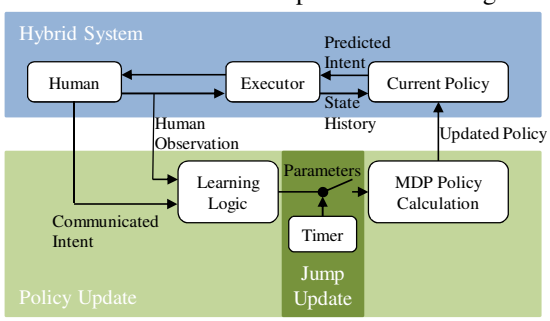

Figure 1: Markov Decision Process for this implementation

This architecture is proposed to predict human intent in the context of a human-robot collaboration application.

Intent prediction is based on a dynamic model of human preferences and observations of recent human action sequences. The policy executes to determine the human's intent, cast as the next action the human is most likely to execute. To predict the next action, an updated human state is sensed from the environment, and that human's known goal-objectives and previous actions are given to the MDP model policy. Use of the MDP requires the Markov Assumption, that is, independence of future state from past state. However, an abbreviated state history captured within the current state can be used when some past history is required. In our case, as each objective may require multiple actions for completion, knowledge of the past n_a actions can help clarify which objective is currently active and which action in the objective-completion sequence might be next. n_a is considered a critical parameter to select for a given domain. Environmental information is included in the form of sensed events that are folded into the human's goal/objectives prior to inclusion in the MDP state vector. This information is necessary for the human to make decisions about motivations that have external and internal consequences. For now, we assume the human senses all events as soon as they occur and their goal objectives will immediately change accordingly; this means that we currently do not include models of distraction, nor probabilities of sufficient attention and correct identification of environmental events given occlusion or fatigue.

We first seed the MDP with a model of human preferences based on parameters obtained from the offline processing of human subject data – this is what we call a 'simulated human' model. For real-time use we must account for possible drift in human preference over time because of the simplicity of the model used. We account for inaccuracies in the initial model by periodically improving our model of human preferences in real-time. The iterative, online adaptation of the MDP process occurs through a learning process similar to the one described in Ref. 15 that tracks a history of the human's action-task state. This learning process can also allow the utilization of filtered human-intent information obtained through sporadic unprompted explicit communication from the human collaborator.[12] However, we assume that any such information is preprocessed into the sensed input human state. This is what we call a 'human matching' model.

Because of the complexity and variability of human cognition, we do not try to create a single "most efficient" or "most complete" model. Rather, we attempt to find a model that will best match an observed set of human actions under differing circumstances – a range of environmental and human states within a partially-restricted setting. We also assume in our current and past work that any robot sharing the human's workspace can act intelligently enough to be unobtrusive: that the human can ignore the robot while they work cooperatively, rather than collaboratively, so there is no need to include robot state in the human's model. In the best case, the observed time history of human goal-objectives and actions that we use to learn the model parameters will span a sufficiently large cross-section of the full set of possible interactions such that we can obtain the 'best fitting' policy.

We use a Markov Decision Process to generate a human intent prediction policy because it can handle uncertainty in models, so long as data is observable, and the robot can use a pre-determined policy for choosing the best action based on observation of the current human activity as well as prediction of the human's next action.

We define a discrete time stochastic dynamic programming (SDP) problem – also known as a Markov Decision Process (MDP) – as:[16-17]

$$MDP = \{T, S, A_s, p_t(s^j|s^i,a_k), r_t(s^i, a_k)\} \rightarrow \pi_t(s^i) \tag{1}$$

$$MDP = \{S, A, T(s^i,a_k,s^j), R(s^i)\} \rightarrow \pi(s^i) \tag{2}$$

where T is the set of decision epochs, times t at which decisions are made; S denotes the set of states s^i; A (and A_s) denotes the set of available (or allowable) actions a_k; $R(s^i)$ is the set of state dependent rewards, with $r_t(s^i,a_k)$ denoting the specific reward for performing an action a_k at a state s^i for epoch t; and $T(s^i,a_k,s^j)$ is the set of all possible combinations of transition probabilities $p_t(s^j|s^i,a_k)$ that indicate the probability of a state s^j occurring as an outcome, given an action a_k when executed from a state s^i. MDP's do not have a separate cost function C, only a reward function. We can pose costs as 'rewards with negative utility' without losing functionality. We assume that the optimal policy $\pi(s^i)$, of the optimal action set a_k for each potentially reachable state in S, is consistent across all time epochs T, such that the MDP policy is time-invariant.

As this is an infinite horizon case, for policy calculation we use the standard Gauss-Seidel value iteration algorithm to evaluate a set of reward values for a given MDP model, for a particular convergence rate. A timer input is used to control when an updated policy is calculated and transmitted to the MDP block.

IV. MDP Formulation for Human-Intent Prediction

We formulate the HIP problem as follows:

$$
\begin{aligned}
&S = \left\{s^1, s^2, ..., s^{n_s}\right\} \\
&s^i = \left\{G^i, A^i, F^i\right\} \\
&G^i = \left\{g_1^i, g_2^i, ..., g_{n_g}^i\right\}, g_k^i \in \{0,1\}, k = \{1,..,n_g\} \\
&F^i = \left\{f_1^i, f_2^i, ..., f_{n_f}^i\right\}, f_k^i \in \{0,1\}, k = \{1,...,n_f\} \\
&A^i = \left\{a_1^i, a_2^i, ..., a_{n_h}^i\right\}, a_k^i \in A, A = \{1,2,...,n_a\}, k = \{1,..,n_h\}
\end{aligned}
\tag{3}
$$

Every state s^i given to the MDP is assumed to be observable. Each state s^i in S includes variables denoting: the human's low-priority mission goal states G^i, which can be inferred from sensed human actions); actions A^i, the abbreviated past history of n_h actions that are in the set of possible human actions A; and the high-priority interruptive goal states F^i, changes in the human's workspace that are activated by sensed events and override mission goals. We discretize values to reduce model complexity: goals states are either false (0) or true (1), and the set of actions A has cardinality n_a with symbolic actions mapped to natural numbers [1 n_a]. The human's internalized goal objectives g_k^i and f_k^i – the goals that the human attempts to satisfy via action-choice at any given moment – are assumed to be independent from each other: they cannot be further simplified or combined, and the human can only attempt to fulfill one high-priority goal-objective at a time. Low priority goals, however, may be satisfied in groups, depending on the impact of the action chosen.

$$
\begin{aligned}
&\forall g_k^i, g_k^i \rightarrow \left\{\left\langle a_k^1, ..., a_k^{p_k}\right\rangle, \left\langle a_k^1, a_l^1, a_k^2 ..., a_k^{p_k}\right\rangle, ...\right\}, a_k^x \in A, a_l^x \in A, p_k \le n_h \\
&\forall f_k^i, f_k^i \rightarrow \left\{\left\langle a_k^1, ..., a_k^{p_k}\right\rangle, \left\langle a_k^1, a_l^1, a_k^2 ..., a_k^{p_k}\right\rangle, ...\right\}, a_k^x \in A, a_l^x \in A, p_k \le n_h
\end{aligned}
\tag{4}
$$

As shown in Eq. (3), we assume that each goal-objective can be completed by a particular action or ordered set of actions that the human must perform to satisfy that goal objective. This ordered set is an n-tuple (action) sequence, where $n=p_k$ is the number of actions corresponding to completing a goal. Some goals may have many satisficing action-sequences, if the actions do not need to be completed in a strict order with no interruptions in sequence. Goal objective activation is assumed to occur external to the MDP model. The value of n_h is consistent for each model and chosen or otherwise optimized offline. We utilize information about goal satisficing action sequences to simplify the transition probability tensor, by setting impossible state transitions to zero probability.

$$
R(s_i) = \sum_{j=1}^{n_g} \gamma_j r_1\left(g_j^i, A^i\right) + \sum_{j=1}^{n_f} \phi_j r_2\left(f_j^i\right) - \sum_{j=1}^{n_f} \varphi_j r_3\left(f_j^i\right) \tag{5}
$$

The reward function $R(s_i)$ for each state given in Eq. (4) is based on fulfillment of recurrent objectives of the human. We assume that the weighting variables – rewards γ_j and ϕ_j, and the costs φ_j – are constant for a given MDP policy. We test weights in the range of [0 1] with a change in weight of $\Delta = 0.25$. We generally assume that the set A is a superset of human actions that allow for the possibility of fulfilling each modeled goal objective, when the proper action-sequence is performed. We also assume that the human tries to respond rationally based on known mission objectives to be completed. We define $r_1, r_2, r_3 \in \{0,1\}$, such that the value of completion is transparent and directly proportional to the weights. Also, from Eq. (3), g_k^i and f_1^i are discrete-valued variables, where a value of 1 implies the goal-objective has been satisfied, while 0 indicates that the goal-objective has not yet been achieved or the high-priority interruptive goal state is in a non-optimal state. This means that reward is only given once an objective is met; there is no reward for partial success.

For the HIP formulation, function r_1 calculates how close a human is to achieving an objective based on the last n_h actions. Because we assume g_k^i and f_1^i are known sensed states, r_1 is independent of A^i in our formulation. It is implied that when the model transitions to a new state s^j, the goals and the action history in s^i are also updated inside the transition equation. The weighting of r_1 determines the importance of completing that goal-objective. The

weighting of function r_2 expresses a reward associated with the high-priority event being handled, while the weighting of function r_3 expresses a cost associated with not completing or dealing with a high-priority event flag as soon as possible. This is our way of allowing the formulation to treat some or all of the higher-priority interruptive goals as overriding goals that must be dealt with – if the cost weight is high enough relative to the values of γ_j, it can further stratify the goal-objectives. This does however effectively create an offset in the reward calculation. The reward functions are based on the normal or expected behavior of the human –a baseline of average behavior over a large number of experiments.

$$p\left(s^j \mid s^i, a_k\right) = T\left(s^i, a_k, s^j\right), s^i \in S, s^j \in S, a_k \in A$$
$$satisfying \sum_{j \in \{1,\ldots,n_s\}} T\left(s^i, a_k, s^j\right) = 1, \forall i \in \{1,\ldots,n_s\}, \forall k \in \{1,\ldots,n_a\} \quad (6)$$

The HIP MDP transition probability function or tensor in Eq. (6) represents the probability that a human will transition to a state s^j, when performing an action a_k in a particular state s^i. From the formulation of states, actions, and reward function, an expected action for a human can be calculated, but as the human is not wholly-rational, they are not guaranteed to always stick to its expected behavior. The selection of internal structure must then implicitly allow for a possibility of deviation of the human behavior from an otherwise-expected action – the optimal behavior that would give the "most efficient outcome". In the above equation, $a^i_{n_h+1} = a^j_{n_h} = a^*_k$ would be the expected optimal behavior. We generally handle the unmodeled vectors that could lead to this deviant behavior by allowing a "non-optimal" choice of parameters that maximizes matching the human's sensed response – the subjective choice they made – rather than an objective "best" choice of action behavior given a state s^i, considered optimal within the simpler computational model. The function T does not have to be analytic, though ours is, and that n_s is the total number of possible state combinations, $n_s = 2^{n_g} * n_h^{n_a} * 2^{n_f}$ with $n_h = 4$ history states for our chosen baseline scenario.

As mentioned earlier, we solve for an MDP policy using the standard Gauss-Seidel value iteration algorithm to evaluate a set of reward values for the given model, with discount factor 0.95 and an error bound of $1*10^{-5}$.

V. Case Study: Space HRI Domain Formulation

In this paper, we use a simple domain model that will be familiar to most readers: eating, drinking, interacting with a computer, augmented by pushing buttons in time-critical circumstances. These tasks would be required for astronauts in space as well as for humans on Earth, and we have conducted previous human subject experiments that confirm HRI is feasible for this scenario.[18] We hypothesize that our basic simulation and experimental results will translate to models of humans performing similar activities in IVA in space. In our previous work,[18] we constructed experiments in which the human was seated in a chair next to a robotic manipulator arm, and could eat chips, drink soda, solve math problems, or press buttons. The human was asked to type solutions to simple arithmetic problems as quickly and efficiently as possible while not overly concerning themselves with the robot's motion, and the human test subject was also asked to eat or drink at times, with food on their left and drink on their right. High-priority overriding goals –button-pressing tasks – were indicated to be necessary via messages displayed on the same monitor that displayed the math-problem solution input, and the human subject could not continue to input solutions to the math work until the indicated button was 'pushed off'. No more than one overriding goal was given at a time.

In our first-cut model of the human, we define the baseline problem to include a sporadic need to press only one button. This initially simplifies the human choice preference to be between just hunger/thirst/math versus button-pushing, with the latter objective able to override all other operations, depending on the parameters used in our reward and transition probability function formulation.

We therefore choose the following domain representation as given in Eq. (7):

$$
\begin{aligned}
&n_g = 3, n_a = 5, n_f = 1, n_h \in \{1,2,3,4,5\} \\
&s^i = \left\{g_1^i, g_2^i, g_3^i, a_1^i, ..., a_{n_h}^i, f_1^i\right\} \\
&a_k^i \in \{1,2,3,4,5\} \\
&g_1^i \in \{0,1\}, g_2^i \in \{0,1\}, g_3^i = \in \{0,1\} \\
&f_1^i \in \{0,1\} \\
&R(s_i) = \gamma_1 g_1^i + \gamma_2 g_2^i + \gamma_3 g_3^i + \phi_1 f_1^i - \varphi_1 (1 - f_1^i)
\end{aligned} \tag{7}
$$

Table 1. Domain Representation of actions a_k^i

Discrete Value	Corresponding Action
1	eat_chips
2	drink_soda
3	computer_work
4	push_button
5	no_op

Table 2. Domain Representation of goal-objectives

Goal Obj.	Discrete Value		Corresponding Action
	false	true	
g_1^i	0	1	?hunger?
g_2^i	0	1	?thirst?
g_3^i	0	1	?work_motivation?
f_1^i	0	1	?button_1_inactive?

Tables 1 and 2 describe the human's actions, goals, and fault status used for our domain. We do not explicitly differentiate between physical and mental tasks in our MDP representation, mixing action such as computer work (math) with eating, drinking, and button pushing.

The equations that make up the domain-specific transition probabilities are given in Eq. (8) and Eq. (9):

$$
\begin{aligned}
&\alpha_x = \left\{\alpha_x^1, ..., \alpha_x^{n_h}\right\}, x = \{1, ..., n_g\} \\
&P_{z,x} = p(g_z^j = 1 \mid A^i, a_x) = \frac{\sum_{y=1}^{n_h} \alpha_x^y * (a_y^i == a_x)}{\sum_{y=1}^{n_h} \alpha_x^y} \\
&generally, P_z = p(g_z^j = 1 \mid A^i) = \frac{\sum_{x=1}^{n_x^z} p(g_j^i = 1 \mid A^i, a_x)}{n_x^z}
\end{aligned} \tag{8}
$$

α_x is a vector of weights that define the impact of an action a_x in the action history on a probabilistic change in state, given how far back in time it occurred. For instance, α_x = [0 0 0 1] would give weight to only the most recent action, while α_x = [4 3 2 1] would associate more distant actions in the history as having greater impact than more current actions. Also, when all the terms of α_x are nonzero, a history with many actions a_x actions would imply that it is more likely that the associated goal objective will transition if the choice of next action a_k is the same as a_x. For instance, it is more likely that the person will transition to a state of no-hunger if they have eaten many times before and they choose to eat again, than if they had not eaten multiple times in the past. We restrict values for α_x to be within the range of whole numbers [0 n_h], allowing for a straight-ranking of history terms, or for a relative weighting if the weights are repeated. $P_{z,x} = p(g_z^j = 1 \mid A^i, a_x)$ is defined as the probability of goal objective g_z^j being or becoming 1 (completed) due to occurrences of action a_x in action history A^i of state s^i. We define P_z, the probability of a goal objective g_z^j completing given an action history A^i, explicitly because some goals may be influenced by n_x^z multiple actions. For instance, sometimes people feel hunger when they are actually thirsty, so under these circumstances, both a hunger-fulfilling action and a thirst-fulfilling action may have an impact on reducing a feeling of hunger (completing the hunger objective). n_x^z may vary for different goal objectives *z*, and P_z may differ depending on dependencies, e.g., problem-specific multiplicative terms.

$$\beta_q = \left\{\beta_q^1,...,\beta_q^{n_f}\right\}, q = \{1,...,n_b\}$$
$$B_{z,k} = p(g_z^j = 1 \mid s^i, F^i, a_k) = \sum_{m=1}^{n_f} \beta_q^m * (1 - f_m^i), q = f(s^i, g_z^j)$$
$$p(g_z^j = 1 \mid s^i, a_k) = \frac{P_z + B_{z,k}}{1 + n_f} \tag{9}$$
$$generally: T(s^i, a_k, s^j) = \frac{\sum_{z=1}^{n_f^j} p(g_z^j = 1 \mid s^i, a_k)}{n_T^k - 1}, s^i \to s^j \Rightarrow g_z^j \to 1$$

The variable β_q is a group of weights that define the probability that the human choosing action a_k for their next action will result in a goal-objective g_z^j being/becoming 1 (completed) due to that action. Being a probability, we set each β_q to be in the range [0 1] with a change in weight of 0.25. q is dependent upon which possibility of transition is being tested (see Table 3 below). To simplify our model, we reduce the number of weighting terms to n_b per high-priority event, with n_T^k is the number of possibilities of transition for a particular action a_k. We then define n_b to be the number of possibilities of transition minus the number of actions; we can subtract n_a because for every action a_k we can calculate the n_T^k 'the probabilistic outcome for that action from the other n_T^k -1 terms, as all probabilistic outcomes for an action must sum to 1. The impact of $B_{z,k}$ is dependent on the high-priority interruptive goal states. If a high-priority interrupt flag is not set, then the probability $B_{z,k}$ of a choice of action a_k completing a low-priority goal is added to the transition probability; otherwise, it is not included because he probability of completing a low-priority goal when a high-priority goal exists is 0. The divisions performed in Eq. (8) and Eq. (9) normalize terms to ensure probability values between zero and one.

As described above, the transition probability is dependent upon the past action history A^i and the predicted next action choice of the human a_k. There are no explicit mappings between a goal state and an action-choice, but the probabilities are constrained to the known valid action-choices for goal-completion to simplify the number of parameters to optimize. By leveraging this information, we do not need to specify the full tensor, thereby reducing the computational complexity. For our domain, we have n_b =10 possibilities:

Table 3. Possible Outcomes of Action-Choices a_k

Action choice	Possible changes in state when action a_k is applied (0=false, 1=true; goal states 'and'ed together across columns of a row; blank cell="no change")				
	g_1^i, $s^i \to s^j$?hunger?	g_2^i, $s^i \to s^j$?thirst?	g_3^i, $s^i \to s^j$?work_motivation?	g_4^i, $s^i \to s^j$?button_1_inactive?	influencing β_q^1 (q=?)
eat chip (a_k=1, n_T^k=2)	0→1				1
	0→0 or 1→1				1 (indirect)
drink soda (a_k=2, n_T^k=4)	0→1	(0→0 or 1→1)			2
	(0→0 or 1→1)	0→1			3
	0→1	0→1			4
	0→0 or 1→1	0→0 or 1→1			2,3,4 (indirect)
solve math (a_k=3, n_T^k=2)			0→1		5
			0→0 or 1→1		5 (indirect)
push button (a_k=4)				0→1	n/a ($p(f_1^j=1 \mid a_4)$=1)
no-op (a_k=5)					n/a ($p(s^j=s^i \mid a_5)$=1)

Note that we only have five β_q^1 instead of seven because the push_button and no_op actions have only one transition possibility each – if a person intends to perform either of these actions, they will perform that action with probability 1.

As an example, we define some of our probabilities in this formulation as:

- Probability of eat=1 given chips action: $p(g_1^j = 1 \mid s^i, a_1) = \dfrac{\dfrac{(P_{1,1} + P_{1,2})}{2} + \beta_1^1 (1 - f_1^i)}{2}$
- Probability of eat=1 & drink=1 given drink action: $p(g_1^j = 1 \,\&\, g_2^j = 1 \mid s^i, a_2) = \dfrac{\dfrac{P_{2,2} * (P_{1,1} + P_{1,2})}{2} + \beta_4^1 (1 - f_1^i)}{2}$

The "drink" probability is slightly more complex than other probability computations because we have four possible changes in state due to the minor coupling of the drink action influence on the combined state (on both the hunger and thirst objectives), as discussed earlier. The $P_{2,2}$ term is multiplicative over the additive term ($P_{1,1}$ + $P_{1,2}$)/2 because we assume the objectives thirst and hunger are conditionally independent from each other.

As shown, our state-space model includes an action history as well as attributes describing sensed events. By reduce the mapping of objectives directly to actions as shown in Eq. (4), we are effectively tracking action-completion as our state. This also reduces the complexity of our representation.

While this Earth-based experiment may be simple, it has direct analogs to a possible IVA environment. Many internal on-orbit operations are done at a computer workstation, and when deadlines are short an astronaut may eat or drink at the same time as they complete other work. The button-pushing task is an interruptive task that needs minimal upkeep time intermittently – such as checking up on a science experiment that sets off an alarm on a schedule the astronaut is not necessarily tracking, being concerned with their own main work. This also has parallels with on-orbit EVA space-repair at a satellite electronics panel: 'chip eating' is a retrieval action, such as consumables that may need to be used to fix internal electronics; 'soda drinking' is a pick-and-place action, such as the retrieval, use, and stowing of a screwdriver; 'math problems' are a highly cognitive-stressing action, such as troubleshooting problems inside the panel mainly by visual inspection; and 'button pushing' is a task of overriding importance, such as a time-critical task like noticing and grabbing a toolkit before it floats away.

A. Learning Logic

For the 'human-matching' model, our policy must be updated to best represent the human. This need could be due to our original Earth-based model not quite matching the human's reaction in a space-based case, or because the human's preferences shift over time as they operate, due to unmodeled changes in their own mental state or the environment. Instead of reformulating the entire problem, however, we choose to update the model as divergences are recorded. In this manner, we view the problem as one of determining the "best matching" policy over multiple epochs, without knowing the timespan each policy will cover. Although this introduces lag, we assume in this work that the system will notice lag-induced divergences before they become problematic. To create and update our 'human matching' model, we include a policy update procedure (see Fig. 1) that may learn new or evolving parameters that describe the human, or otherwise replace the model policy being used for the real-time human prediction with a better policy. For this purpose, we use what we call a "Multiple-Model Learning Method" similar to what the dynamic systems literature calls "Multiple-Model Fault Diagnosis."[19] In fault detection, multiple dynamics models are used and compared to detect which of the models matches best with the observed dynamics of the system. Similarly, we compare the output of multiple MDP policies to the evolution of states and action-sequences observed. To evaluate which policy is the best match, we use a reward function with weighted terms to score each candidate MDP policy, and pick the highest-scoring candidate for our new "best matching" policy, akin to a maximum likelihood approach. We assume the Markov property, in that we can discard previous state data in the time history that we perceive as no longer useful.

First, human subject testing is performed to obtain the human reaction data necessary to populate the baseline 'simulated human' model. The test environment would generally be an Earth-based setup, which would have an analog to the space IVA case. Then, during real-time operations, a time history of human states $S_{t,t+T}$ and actions $H_{t,t+T}$ is recorded from the last change in model policy used, and the accuracy of each of the current and available model policies is evaluated through comparison of the policy output to the actual human state and action-sequences being observed. Models are rated given a weighted average of the accuracy of results calculated using Eq. (10), which takes into account the number of matching transitions in a state-action chain and the number of times an output action from the model matched the observed action. The final best model is chosen from the highest scoring of these. This model updating procedure is our 'human matching' model. Because human subject experiments to validate this process were not possible before this paper was published, the results presented below use simulations of human behaviors only.

$$S_{t,t+T} = \{s_t, s_{t+1}, ..., s_{t+T-1}, s_{t+T}\}, H_{t,t+T} = \{a_t, a_{t+1}, ..., a_{t+T-1}, a_{t+T}\}$$
$$score(H_{t,t+T}, S_{t,t+T}, MDP_k) = \sum_{i=t}^{t+T-1} w_1 * (\arg\max_{s_{t+1} \in S}(T_k(s_t, a_t, s_{t+1})) == s_{t+1}) + \sum_{i=t}^{t+T} w_2 * (\pi_k(s_t) == a_t) \quad (10)$$
$$\pi^* \leftarrow MDP^* = \arg\max_{MDP_k \in MDP}(score(H_{t,t+T}, S_{t,t+T}, MDP_k))$$

VI. Simulation Results

A. Evaluation of MDP Formulation – Impact of Reward Function Weights on Policy

We evaluate the MDP formulation by varying parameter values for the weightings of two goals at a time in a preliminary analysis of the resulting action-choices codified in the policy output before conducting a full Pareto frontier analysis.[20-22] We target these analyses to explicitly compare the tradeoff between different types of goal interdependencies in the equations: two independent goals (chips and math), two partially-dependent goals (chips and soda), and low-priority versus high-priority goals (chips and button-push). We then revisit the first two cases using 'common sense' parameter choices for weights not being directly compared, to see if the use of nonzero terms for the weights of lesser-focus has any great effect. We give a summary of our conclusions in Table 4. The alpha and beta parameter values chosen for the analysis were 'common sense' values given a lack of statistically-significant human subject experiment data.

Table 4. Parameter values for MDP Evaluation (single number constant and/or ranges)

n_h=4	α_1 = [1 2 3 4], α_2 = [1 2 3 4], α_3 = [1 2 3 4], β_1 = 0.25, β_2 = 0.25, β_3 = 0.75, β_4 = 0.25, β_5 = 0.75, $R(s_i) = \gamma_1 g_1^i + \gamma_2 g_2^i + \gamma_3 g_3^i + \phi_1 f_1^i - \varphi_1(1 - f_1^i)$						Comments on percentage of action choice for each action, per policy
	γ_1 eat reward	γ_2 drink reward	γ_3 math reward	ϕ_1 button reward	φ_1 button cost	Δ value variance	
MDP1	[0 1]	0	[0 1]	0	0	0.25	a_3 and a_4 picked about as often, then a_1 and a_5, then a_2
MDP2	[0 1]	[0 1]	0	0	0	0.25	a_2 and a_4 picked about as often, then a_1 and a_5, then a_3
MDP3	[0 1]	0	0	[0 1]	0	0.25	high percentage of a_4 action choice, then a_1, then others
MDP4	[0 1]	0	0	0	[0 1]	0.25	high percentage of a_4 action choice, slightly lower at boundary where γ_1=0 than MDP3
MDP5	[0 1]	γ_1	[0 1]	1	1	0.25	a_2 choice generally less often than in MDP6
MDP6	[0 1]	[0 1]	0.75	1	1	0.25	more even balance between percentage of a_1 and a_2 action choices, a_3 and a_4 choices consistently high due to reward

Generally, when looking at the individual actions without taking into account state correspondence, when comparing policies between MDP's with similar parameter sets when there are tradeoffs in weightings between the two reward objectives, there are few, if any, times where the percentage of an action choice a_k – the absolute number of times a_k is chosen by the policy divided by the number of all possible states – has global maxima and minima. We usually consistently see the highest or lowest percentage of action choice for the actions at the boundary cases in Table 4 – the 'edges' where the reward terms are rewarding only one of the two objectives. We see similar when comparing the ratio of action choices for the policies for actions which have an impact on those objective terms whose weights were subject to a tradeoff. Overall, the drink action a_2 tends to have middling action-choice percentage even when the drink reward is 0, as in MDP2, and is a higher percentage choice than eating when γ_1 is not 0, due to the impact of the drink act a_2 on the eat goal g_1^i. The choice of no-op action a_5 is significantly lower for the MDP5 and MDP6 cases, due to all other actions having a nonzero reward, also as expected. It is notable that the action choices seem consistent and do not tend to jump largely when comparing policy outcomes to other policies with 'similar' weightings (within the delta value variance), so that it is feasible for these policies to have

their terms updated without drastic changes to the parameters, or that a policy could be refined by searching over a 'nearby' range of parameter solutions using a finer mesh (smaller delta) for a closer-fitting solution.

B. Evaluation of MDP Formulation – Impact of Length of History on Policy

As discussed above, the number of parameters in the time history n_h must go back in time sufficiently far that we can assume the Markov property for our formulation. In this context, we define 'instability' to be a lack of sufficient history states, and 'stability' to be the opposite. This can be thought to be similar to the idea of stability in dynamics and controls, in that when the MDP is 'stable', the policy does not have sudden odd fluctuations in action-choice when compared to the state and action history we are trying to match. An indication of 'instability' could be a policy whose action outputs are seen to diverge quickly and frequently from the observed history when used in the learning logic process, even when the internal parameters for alpha and beta and the reward weights are known to be good. (The other possibility could be that the structure of the MDP may not match, but we will assume this not to be an issue in this work.)

In simulation, however, we do not attempt to compare policies using an evaluatory approach similar to Eq. (10), because attempting to examine how policies with differing values of n_h explicitly match their action choice outputs against each other is nontrivial when the number of states differs between policies. There is no good way to decide whether the action given for a state with a certain G^i and F^i with A^i=[1] for a n_h=1 policy matches the action for a policy with differing n_h, since there will be multiple similar states with the same G^i and F^i but with different histories – for instance, with n_h=5, A^i=[1] could potentially match to A^i=[1 1 1 1 1], or [2 1 1 1 1], or [5 2 3 5 1], and so forth. Instead, we evaluate 'instability' in comparisons similar to the analysis performed above to evaluate the reward weightings. Table 5 gives an overview of the cases tested. We test these histories over the ranges of weightings given in Table 4.

Table 5. Parameter values for History Length on Performance Comparison

β's and weights for R are both used as (MDP#) given in Table 4	Values for $\alpha_1 = \alpha_2 = \alpha_3$ (see Eqn. 8 for α_x usage) with a_1 oldest in history, a_{n_h} newest					Comments on percentage of action choice for each action, per policy
	n_h=1	n_h=2	n_h=3	n_h=4	n_h=5	
MDP1 (eat, math) newest actions most important (case 1)	[4]	[3 4]	[2 3 4]	[1 2 3 4]	[0 1 2 3 4]	a_2 choice has a large jump from never being chosen with no history to being chosen frequently with a history; a_5 does the reverse, from being chosen >50% of the time, to usually ~20% or less; a_2 and a_5 change is minimal otherwise a_1, a_3, a_4 change minimal fairly stable for n_h=2 to n_h=5
MDP1 (eat, math) oldest actions most important (case 2)	[1]	[2 1]	[3 2 1]	[4 3 2 1]	[5 4 3 2 1]	a_2 choice has similar jumps as above; similar large dropoff for a_5; a_1, a_3, a_4 change minimal fairly stable for n_h=2 to n_h=5
MDP1 (eat, math) all actions given equal importance (case 3)	[2]	[2 2]	[2 2 2]	[2 2 2 2]	[2 2 2 2 2]	a_2 choice has similar jumps as above; similar large dropoff for a_5; a_1, a_3, a_4 change minimal fairly stable for n_h=2 to n_h=5

To look for instability, we first compute the percentage of an action choice a_k for each policy, then compare the differences between each series of MDP# policies for $n_h \in \{1,2,3,4,5\}$, when the β parameters and the set of weights for R are kept constant and only the α_x and n_h differ. Doing this, we see that when comparing MDP's with a history of n_h=1 – with only the current action known at any point in time, effectively no history – to other n_h choices, we see significant jumps in the percentage of action choice per policy for every action and weighting across the board, especially at the 'edges' where one of the γ=0. We also see similar large jumps for the ratios of action choice, but only at the 'edges' where one of the γ=0. For the n_h=2 to n_h=5 comparisons, the action choices and ratios are generally far more stable, with no standing out trends. It should be noted that even for the more 'stable'

MDP policies, for MDP formulations with the same parameters excepting α_x and n_h, changes in percentage of action choice are not always consistent in direction; for instance, from n_h=2 to n_h=3 the number of action choices or ratio of choices might be increasing, while from n_h=3 to n_h=4 it may decrease. However, generally there are larger jumps between the n_h=1 versus n_h=5 and n_h=4 versus n_h=5 comparisons for case 2 than the other neighboring n_h comparisons; this is not seen with case 1 or case 3, which instead see similar large jumps between n_h=1 and all other n_h values.

The main tradeoff that will result in a marked difference in effect, as seen here, is a choice of action history or no action history: if an action history is needed but only the most recent action is remembered, the model will not match well, and vice-versa. If an action history is needed (n_h>1) but more or less terms are necessary (with n_h>1 still) then in this formulation the MDP policies look to be more forgiving.

C. 'Human Matching' Model – Learning Logic Update Process

For the 'simulated human' model, we used general observations obtained during the human subject testing in order to choose the initial parameters for a guess at an (unverified through experimentation) 'best model' for our MDP formulation. To extend this to the 'human matching' model, we create our database from a range of delta-offsets from the chosen 'simulated human' model parameters, solve for the optimal policies for those parameter sets, and then use the learning logic selection process to choose the parameters for our updated model. We use a process inspired by Ref. 15 to do so, but because our application differs – we predict a human's state, while they are creating an adaptive controller to drive a system – we cannot make the same assumptions – in part due to our smaller dataset size, as the observed human may never transition to all available states necessary for model validation, among other factors – thus we cannot claim that this process is 'truly optimal' or that it will definitely converge. Mathematical proofs that explore the possible level of optimality and accuracy (performance) of a solution using our modified method will be explored in future work. Initial simulation results comparing the scores of various MDPs inside the learning logic with the observed actions of the human are positive. In our procedure, a "score" is the measure of agreement of human actions in our learning data set with an MDP, as given in Eq. (10). For simplicity, we choose w_1=0 and w_2=1, so that an MDP with a particular set of parameters gets an increment of 1 in its score every time its policy-determined action choice from an input state matches with the action in the dataset. Below, our dataset for comparison is obtained by running a 'simulated human' model, as we did not have a statistically-significant set of human subject experiment data for use at the time this paper was published.

To start, we carried out three phases of simulation where each phase consists of 250 actions. We seeded the process with an 'ideal' MDP, which we call MDP2, to supply initial states and output actions for which the process attempted to converge to a best solution, given a set of MDP's to score. In our first phase, the robot predicted actions of the human with an incorrect policy from MDP1. In the second phase, the robot policy was updated by the learning logic, selected based on the maximum score supplied using Eq. 10. In the third phase, MDP of the robot was once again checked by the learning logic process, but the policy was not changed, as the best fitting policy had already been selected. Table 7 shows parameter values for MDP's used in the learning logic. Table 8 shows accuracy of prediction of human actions by the robot in three phases of operation. Figure 2 shows how the scores evolved during the three phases of simulation.

Table 7. Parameter values for MDPs in Learning Logic

MDP1 (initial MDP of the robot)	α_1 = [1 2 3 4], α_2 = [1 2 2 4], α_3 = [1 3 2 4] β_1 = 0.3, β_2 = 0.2, β_3 = 0.7, β_4 = 0.1, β_5 = 0.8
MDP2	α_1 = [1 2 3 4], α_2 = [4 2 2 4], α_3 = [1 2 3 4] β_1 = 0.2, β_2 = 0.5, β_3 = 0.1, β_4 = 0.6, β_5 = 0.95
MDP3	α_1 = [1 1 2 4], α_2 = [4 2 1 4], α_3 = [1 2 2 4] β_1 = 0.4, β_2 = 0.3, β_3 = 0.1, β_4 = 0.5, β_5 = 0.9

Table 8. Prediction Results

Phase	# of actions performed	# of correct predictions	%age of correct predictions	MDP of the robot
Phase1	250	235	94%	MDP1
Phase2	250	250	100%	MDP2
Phase3	250	250	100%	MDP2

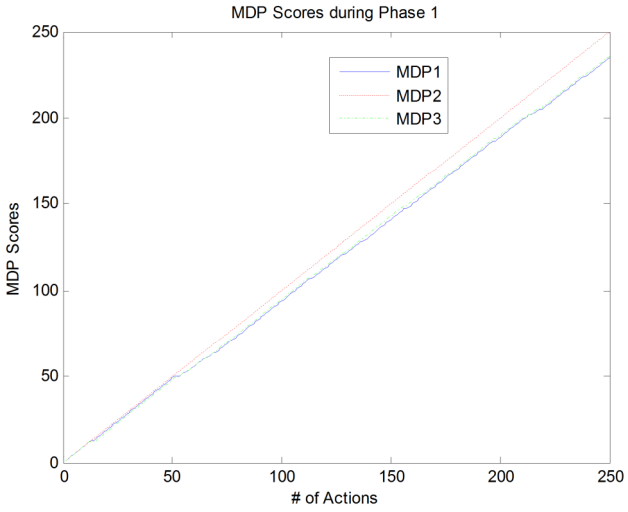

Figure 2. MDP Score2 during Phase 1

For this 'human matching' model, we did not learn the initial 'simulated human' model or 'human matching' model against statistically significant human subject data. Extracting the model parameters directly from human subject data is a nontrivial process, and not in the scope of this paper. Extracting the data for 'human matching' is not trivial, but could be done by conducting human subject experiments, and then action-matching trajectories in the video to the discrete actions in the MDP model and discerning the human's recorded goal-objective states during offline processing of the experimental data.

We believe that this initial work can give a 'realistic' simulated human for general testing of HRI robotic scheduling algorithms. Whether this system will be able to learn and predict the actions of a specific human in real-time well enough to give "accurate-enough" useful predictions is unclear at this time, as the level of accuracy necessary for those predictions is dependent on the robustness and stability of the scheduling process in which it is being used, as well as the reaction time necessary for HRI operational safety. This will be explored in future work.

VII. Conclusions and Future Work

We have presented a Markov Decision Process (MDP) based framework for predicting human (astronaut) intent during a space mission. This framework is intended to be used by a robotic manipulator arm that is trying to perform its tasks without disturbing or endangering the human astronaut. Our framework contains a learning logic procedure that tries to learn the 'best fit' model of the human in terms of their fully-observable goal-objectives and action-choices. Learning has been based on the comparison of the observed actions of the human with a built-in offline-calculated set of model policies and controlled, timed updates to the MDP model parameters and policy. We have presented some initial simulation results where we evaluate policies developed from the model formulation, in order to better understand the characteristics of the 'simulated human' model. We have also seen that the 'human matching' model prediction is shown to be quite accurate on the considered models, but we are still working to establish guarantees about the success of prediction with human subject data. We expect that the accuracy of the HIP system's predictions will be "good enough" to be useful for real-time task scheduling for a robot. Future work will involve the incorporation of this system into a larger scheduling implementation for a manipulator in a real-time physically-proximal human-robot collaboration setting. A ground-based example related to IVA is explored in this paper, but it is assumed that, with the inclusion of a total HRI system including this component, once astronauts become comfortable working side-by-side with robots (e.g., on space station), they will be much more ready to also accept them as companions on EVA. This experimental setting may necessitate the addition of state information about the robot(s) sharing the physical workspace to the goal-objective states of the human in the HIP model, if we wish to allow not just interaction, but also collaboration.

Future work could also involve extension of the MDP to a POMDP formulation. We avoided much model complexity here by assuming that the human goal-objectives could be directly sensed; if were using internal, hidden states of the human (such as hunger, thirst, work-motivation), we would have uncertainty in the input state of the human as part of the MDP process. In a more complex formulation, we would also want to include 'reduction of fatigue' as a goal-objective and "rest"/no-op would have a more intricate effect by reducing fatigue; the fatigue state could also have an impact on the transition probabilities for other actions, due to loss of concentration and

other associated effects. We also decoupled the time history of actions from the transition and rewards processes somewhat – if there was a partial or full match between an action-chain describing an objective and the memory of actions in the input state, it would seem more likely for a particular state change in that objective to occur. Taking this into account would argue for making the objectives not Boolean values, but rather estimates of percent-completion; the impact of an addition of this level of complexity could be explored, along with a comparison of the relative accuracy of the new more-generalized model versus this basic MDP formulation. We also did not discuss how these discrete, generalized actions are identified; we assumed it would be via another process from a trajectory of sensed dynamic state estimations, or, less-abstractly, a time-history of bundled physical poses, but the exact details of integrating such a process, and the accuracy of the results, would be substantial work.

Acknowledgments

The authors would like to thank the UROP students who helped with the MM-Arm platform development and human subject testing, and for those volunteers who participated in the human subject testing that gave us the data used to formulate and populate our MDP models.

References

[1]Yang, J., Xu, Y., and Chen, C. S., "Human action learning via hidden Markov model," *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, Vol. 27, No. 1, 1997, pp. 34-44.

[2]Aggarwal, J. K., and Cai, Q., "Human motion analysis: A review," *Computer Vision and Image Understanding*, Vol. 73, No. 3, March 1999, pp. 428-440.

[3]Yamato, J., Ohya, J., and Ishii, K., "Recognizing human action in time-sequential images using hidden Markov model," *Proceedings of the CVPR'92, IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1992, pp. 379-385.

[4]Bregler, C., "Learning and recognizing human dynamics in video sequences," *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1997, pp. 568-574.

[5]Brand, M., Oliver, N., and Pentland, A., "Coupled hidden Markov models for complex action recognition," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1997, pp. 994-999.

[6]Lee, C., and Xu, Y., "Online, interactive learning of gestures for human/robot interfaces," *Proceedings of IEEE International Conference on Robotics and Automation*, 1996, pp. 2982-2987.

[7]Bobick, A. F., "Movement, activity and action: The role of knowledge in the perception of motion," *Philosophical Transactions of the Royal Society B: Biological Sciences*, Vol. 352, No. 1358, August 29 1997, pp. 1257-1265.

[8]Poppe, R., "Vision-based human motion analysis: an overview," *Computer Vision and Image Understanding (CVIU)*, Vol. 108, No. 1-2, 2007, pp. 4-18.

[9]Poppe, R., "A survey on vision-based human action recognition," *Image and Vision Computing*, Vol. 28, No. 6, 2010, pp. 976-990.

[10]Karami, A.-B., Jeanpierre, L., and Mouaddib, A.-I., "Human-robot collaboration for a shared mission," *Proceeding of the 5th ACM/IEEE international conference on Human-robot interaction (HRI'10)*, 2010, pp. 155-156.

[11]Karami, A.-B., Jeanpierre, L., and Mouaddib, A.-I., "Partially Observable Markov Decision Process for Managing Robot," *21st International Conference on Tools with Artificial Intelligence (ICTAI '09)*, 2009, pp. 518-521.

[12]Matignon, L., Karami, A. B., and Mouaddib, A. I., "A Model for Verbal and Non-Verbal Human-Robot Collaboration," *2010 AAAI Fall Symposium Series*, 2010.

[13]Schmidt-Rohr, S. R., Knoop, S., Losch, M., and Dillmann, R., "Bridging the gap of abstraction for probabilistic decision making," *Robotics: Science and Systems, Zurich*, 2008.

[14]Kaupp, T., Makarenko, A., and Durrant-Whyte, H., "Human–robot communication for collaborative decision making — A probabilistic approach," *Robotics and Autonomous Systems*, Vol. 58, No. 5, May 2010, pp. 444-456.

[15]Kumar, P., and Becker, A., "A New Family of Optimal Adaptive Controllers for Markov Chains," *IEEE Transactions on Automatic Control*, Vol. 27, No. 1, Feb 1982, pp. 137-146.

[16]Puterman, M. L., *Markov Decision Processes: Discrete Stochastic Dynamic Programming*, 1st ed., John Wiley & Sons, New York, USA, 1994.

[17]Russell, S. J., and Norvig, P., *Artificial intelligence: A modern approach*, 2nd ed., Prentice Hall/Pearson Education, Upper Saddle River, N.J, 2003.

[18]McGhan, C. L. R., and Atkins, E. M., "Physically-Proximal Human-Robot Collaboration: Enhancing Safety and Efficiency Through Intent Prediction," *Proc. Infotech@Aerospace Conference*, Seattle, WA, Apr. 2009.

[19]Rago, C., Prasanth, R., Mehra, R. K., and Fortenbaugh, R., "Failure Detection and Identification and Fault Tolerant Control

using the IMM-KF with applictions to the Eagle-Eye UAV," *Proceedings of the 37th IEEE Conference on Decision and Control*, Tampa, Florida, December 1998.

[20]Dellnitz, M., and Junge, O., "An adaptive subdivision technique for the approximation of attractors and invariant measures," *Computing and Visualization in Science*, Vol. 1, No. 2, 1997, pp. 63-68.

[21]Fonseca, C. M., and Fleming, P. J., "An Overview of Evolutionary Algorithms in Multiobjective Optimization," *Evolutionary Computation*, Vol. 3, No. 1, Spring 1995, pp. 1-16.

[22]Teich, J., "Pareto-Front Exploration with Uncertain Objectives," in *Evolutionary Multi-Criterion Optimization, Lecture Notes in Computer Science, 2001*, Zitzler, E., Thiele, L., Deb, K., Coello Coello, C. & Corne, D. eds., Springer Berlin / Heidelberg, 2001, pp. 314-328.