AIAA SPACE 2012 Conference & Exposition
11 - 13 September 2012, Pasadena, California

AIAA 2012-5317

Towards Guaranteeing Safe and Efficient Human-Robot Collaboration Using Human Intent Prediction

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

This paper describes an autonomous framework for determining a robotic manipulator's optimal actions in real-time when interacting in close physical proximity to a human in a shared workspace environment. This framework allows the robot to purposefully choose to avoid physical and mental conflicts with a human companion while each agent performs tasks to complete their respective, separately-assigned goals. We pose scenarios in which the human does not need to divert attention to internally model the robot's behavior, or track or acknowledge the robot's actions during operations. The robot is meant to unobtrusively 'work around' the human rather than directly collaborate on task completion. The distinction of this work is in its use of human intent prediction (HIP) as a key factor in robot action selection for task-level planning. We choose to model HIP with a Markov Decision Process (MDP). Human state data is input into the HIP MDP policy that then outputs the predicted human intent, which we define as the best-matched or most-likely in-progress and future action-choice(s) that the human is or will be pursuing to complete mission goals. Predicted human intent is then used by a second MDP to determine the optimal policy with respect to the robot's action-choice. We present an autonomous framework that integrates the HIP MDP and robot action-choice (RAC) MDP to support autonomous close-proximity operations and propose offline and online (scaled) formulations of the two MDPs. During real-time policy execution, once the optimal action for the robot to take is determined, it is passed to the robot's path planner to be translated from a task-level command to a trajectory and motion primitives, which are then given to a low-level controller to enact. We evaluate our HIP MDP in simulation, and find that the policy output from our system is consistent and smooth across small changes in parameter values.

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
$\vec{E}$ = environmental data, $\vec{E} = \{E, L\}$
E = object *k* status in environment, $E = \{{}^{E}x_1, ..., {}^{E}x_{n_l}\}$, where n_l is the number of tracked objects
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
L = list of object locations, $L = \{{}^{obj}\vec{x}_1, ..., {}^{obj}\vec{x}_{n_l}\}$, where n_l is the number of objects being tracked
M = discretized safety metric, obtained via calculations involving danger metric and spatial zones Z

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

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

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

${}^h\vec{x}, {}^h\dot{\vec{x}}$ = position and velocity of the human agent's wrist, where ${}^h\vec{x} = [{}^hx \quad {}^hy]$, ${}^h\dot{\vec{x}} = [{}^h\dot{x} \quad {}^h\dot{y}]$

${}^{obj}\vec{x}_k$ = location of object k, ${}^{obj}\vec{x}_k = [{}^{obj}x_k \quad {}^{obj}y_k \quad {}^{obj}z_k]$

X^i = subset of current and/or future-predicted action-choices

Z = set of spatial zones for the MDP, $Z = \{1, 2, ..., n_z\}$

Z^i = history of recently visited zones in state i, $Z^i = \{z_1^i, z_2^i\}$, $z_k^i \in Z$

I. Introduction

Human-robot collaboration scenarios often practically assume the agents' workspaces have little to no overlap. However, this is an extremely conservative measure that can constrain a mission compared to what would be possible if workspaces could be safely and effectively shared. Modern sensor systems can now enable robots to reliably sense nearby humans or, more generally, moving objects in real-time with enough accuracy to support safe close-proximity operations. Further, this information can be integrated into the robot's decision-making processes to allow the robot to customize its reactions based on its companion's activities. Specifically, we propose a two-step robot decision-making process for proximity operations. First, the robot predicts the intent of its human companion based on a priori knowledge and real-time observations. Second, the robot uses this intent prediction along with its own task-level goals to select the optimal next action or action sequence. Selected actions are then executed and the cycle repeats.

Previous work shows that human action recognition is possible through the use of Markov Decision Processes (MDPs)[1-9], while human intent can be determined with partially-observable MDPs (POMDPs)[10-12]. In this paper we utilize a MDP to simplify computational overhead, and go one step further to *predicting* future human intent for robot decision-making, as well. There has been similar work recently in robotic action-choice done in a collaborative setting, using MDPs to learn and help deconflict collaborative activities by learning and agreeing to a common task assignment distribution between a human and robot via agreement on a shared mental model (SMM), when both of the agents are capable of performing all actions in a collaborative task in an overlapping workspace, and would like to share the work.[13] Our research explores an alternative direction – a more simplified task model that does not require direct human-robot collaboration for task accomplishment, but a more complex constraint to minimize overhead for the human by eliminating all robot supervision and communication demands, with a focus on safety. In this paper we describe a general approach for translating a specific proximity operations scenario – environment, goals, constraints, agents, actions, and sensory input – into a domain model that a robot can understand and use. Our representations below correspond to a simple example scenario that is discussed in more detail in our previous work.[14]

We propose a two-step decision process that allows a robot to determine the locally-optimal action-choice for overall human-robot team efficiency and productivity with constraints imposed to maintain a minimum level of safety, where safety translates to collision avoidance in this work. We describe the information transfer between architectural modules and specifics of the MDP models required for human intent prediction (HIP) and action choice. We describe an approach to transform sensor data to a form useful for robot decision-making and then briefly discuss how the action-choice output is handled once calculated. We present results from a HIP MDP for a simulation-based case study of a space-based human robot interaction (HRI) scenario in which the robot and human have distinct but physically-overlapping tasks to complete.

II. Problem Statement, Assumptions, Definitions, and Simplifications

To enable locally-efficient team operations that meet a guaranteed minimum level of safety (avoiding conflict and collision between agents), the robot must sense its human companion, process sensed information to extract the human's current state, predict the human's intent, and then use that prediction to inform the robot's action choice.

In order to do this, we propose the following hypotheses, and explain their supporting basis in our application space. Assumptions required to simplify the problem in this paper are also stated.

1. A robot can predict companion intent by identifying actions based on sensor observations without relying on explicit communication, then recognizing those observed actions as part of a sequence.

In a space environment, the most-likely structured action sequences would be known in-advance for EVA operations. For IVA collaboration, action sequences are less certain as the human is less restricted in the environment – no EVA suit limiting motion, a wider visual field and tactile cues, and so forth. However, long-term observation of a human companion's behavior could inform the prediction of action sequences.

We choose to restrict our human state data to observable (sensed) positions and motions to focus on interaction cases without explicit communication (e.g. verbal communication, physical gestures). We also simplify our human model by not including the human's model of the robot state, as such state features would not be directly observable thus would require our MDP to become partially-observable. Machine learning can be used to determine bundles of motion-trajectories – or discrete zones in physical space – that correspond to each observable and modeled action sequence a human companion can execute.

In addition to assuming a closed world and full observability, we assume that the robot has sufficient memory to store an *n*-action history, for a finite but potentially large *n*. This state history allows the robot to best estimate the specific goal-directed action sequence its human companion is executing. We assume that human subject data exists for specifying the relevant human model parameters, or that a process exists for observing the human to iteratively improve parameter estimates, and that we can calculate viable models and procedures or policies offline prior to online use. We assume that this offline-calculated information can be stored in an online-searchable database so that updates – changes in what previously-calculated information is selected for use – can be made in real-time when necessary, and that a mechanism exists for performing these updates in a timely fashion. We discuss an update process for this in Ref. 15.

2. The use of predicted companion intent results in improved real-time robot action-choices over those made without it, when the relative worth of the intent data is known and both are supplied to a procedure derived from a 'good-enough' domain model.

We define 'improved robot action-choice' as the optimal choice for the goal completion needs of the entire human-robot team, rather than only the robot's own goal completion needs, assuming that robot tasks do not have higher-priority than human tasks. In our HRI scenario, the robot is meant to 'work around' the human to minimize the human's overhead, so this assumption is valid. From previous human subject testing, we have determined that, in our ground-based scenario, the inclusion of the robot in shared-workspace operations without explicit communications did not significantly reduce human productivity, even when only minimal conflict-avoidance algorithms were used for 'intelligent' task-selection.[14,16] This implies that so long as the robot causes only minimal interference or conflict with the human, the human would be expected to have similar productivity as if the robot was not there, so any additional goals accomplished by the robot would improve overall team productivity.

We assume that the inclusion of predicted intent, when it is consistent and trustworthy enough to be usable, will not make the action-choice *less* optimal than the use of current state data alone. Generally, the addition of more and better data to a model or process that can include a measure of data trustworthiness will improve the results.

3. If the human's actions can be classified as rational with divergence within a known bounded uncertainty, a model can be found with parameters that will give a 'good-enough' fit, and a minimum level of safety can be assured in-advance of robot operations.

From basic control theory, we know that imperfect models can still be useful so long as the uncertainty (error) is characterizable and bounded below a certain threshold. Further, if the exerted control can keep the system stable about the equilibrium set point at which the model parameters were identified, and the stable region is large enough, an adaptive controller can be used, and the controller can transition to follow the progression of system state.

In our chosen space scenario, humans have been trained to make highly rational choices in expected ways. Because the human's choices are rational, it follows that there must be a set point about which a well-structured model of that human's behavior can be fitted, even if that equilibrium progresses over time. Also, the uncertainty inherent in any less-than-perfect model of such an astronaut's rational choices should be characterizable within a bounded error.

Safety is defined as "the condition of being protected against physical… or other types or consequences of failure, damage, error, accidents, harm or any other event which could be considered non-desirable, [or otherwise] the control of recognized hazards to achieve an acceptable level of risk."[17] We categorize three different types of safety in our robotics research: mechanical system, software system, and external real-time. The first two are 'internal' faults, the third external. Mechanical system safety failures include physical device failures and consequences of wear-and-tear on the joints, linkages, and so forth. Software system safety failures include communication issues between devices, sensor dropout, unaccounted-for data signal lag, electrical component failure, loss of power, and bugs in the operational code. We assume the first and second types of safety are assured.

We define 'external real-time safety' faults as occurring during interactions of the robot with its surroundings, with other agents, or with itself. This includes physical collisions with agents or the environment, occlusions, to being occluded. We are interested in scheduling process difficulties resulting from aspects of this type of 'external' safety, which are generally best mitigated by prior contingencies or replanning on-the-fly, via a cognitive process.

We consider safety a priority and guarantees can be made for real-time operations, but this is dependent on our models. Unfortunately, no efficiency guarantees can be made when the human is never explicitly communicated with, and therefore cannot be 'forced' to act (or not act) in a certain manner.

In guaranteeing a minimum level of safety, the idea of bounded-input bounded-output stability is useful. With the closed-world assumption, the set of all possible input and output values are known and characterized as finite sets for discrete quantities, bounded and mapped to a finite set of intervals for continuous quantities. If we can also characterize the update rate and the noise (measurement error) for the sensor data, and if the level of noise is not comparable to the level of rationality exhibited by the human, then not all modeled state possibilities are equally likely. We can determine what the most likely human state and state outcomes are. With human intent prediction, we do so by determining the statistics for consistency and uncertainty in each rational human action-choice, using a solution method that supports uncertain reasoning. We can then plan the robot's actions to take full advantage of this and choose balanced safety-efficiency tradeoffs by injecting the groundwork for it into the robot decision process formulation.

With sufficient lookahead, the robot can choose actions that are either always within an acceptable risk level, or determine offline and in-advance all cases of unacceptable risk and then plan outcomes to avoid those bad states. In this work we presume sufficient lookahead is possible through offline MDP policy optimization; in future work online adaptation of model parameters requiring online optimization will require further analysis of lookahead constraints.

III. Problem Formulation

Under the assumptions and simplifications above, we discuss two MDP systems that allow a robot to exploit knowledge of the human's state (obtained without explicit communication) to determine a companion's current state and predict their next action. The robot then uses this information, along with self-knowledge of its own state and knowledge of the traversable environment, to intelligently choose its own action. The context of this deliberative process is discussed below, before we focus on the main MDP formulations.

A. Solution Architecture and Use of Markov Decision Processes

We want the robot to act as autonomously as possible. This requires that the robot sense and understand its environment and the human agent in it, predict human intent and use that intent to inform its action-choice procedure, and then send the proper command signals to enact the action it chose in a reasonable, safe, and timely manner. To support this level of autonomous control of the robotic system, we propose a three-tier architecture (3T).[18-20] At the highest level is the decision-making process, where deliberative cognition takes place; the next level holds the task-selecting reactive executor where optimal policies are utilized to make the robot's action-choice. The lowest level includes a local path-planner and a reactive feedback controller that interface with the robot hardware. We focus on the decision-making process that creates the policies used by the reactive executor with consideration of the form and content of information passed between internal modules – what can and cannot be calculated offline for later use. Figure 1 shows the detailed architecture.

We use an MDP framework to capture and model uncertainty. Although we assume fixed models in this work, more accurate prediction of the next human action may be achieved by accounting for specific user preferences and preference shifts over time. This can be done by updating the choice of model parameters used through learning logic procedures that evaluate observed behaviors using implicitly communicated human action (state) data.[15] We divide the problem into two separate MDPs to address the curse of dimensionality that occurs when both intent prediction and action selection are integrated into a single MDP. Instead, we assume that we can break down the formulation into a serial chain of subproblems, with each module independent of all previous others.

In this problem, we break our deliberation processes into three basic components:

- A translator ${}^{h}T$ that maps raw sensor data of the human's physical location and dynamic pose to zones in continuous physical space; we simplify our problem further by assuming that these zones are unique and disparate, thus additionally map one-to-one to the set of (discrete, symbolic) action-states of the human, the translator's output.
- A MDP that creates a policy we call human intent prediction (HIP), which takes as input the human's current action-state, the (assumed fully-observable) human's current goal, and the current environmental state. The computed policy specifies the human's most likely next action-choice.
- A MDP that creates a policy we call the robot action-choice (RAC), which takes as input the human's current and predicted action-state, the robot's current state, and additional environmental state data. The computed policy gives the most locally-optimal action-choice for the robot to take that will satisfy the minimum safety constraints while maximizing system-level (team) efficiency.

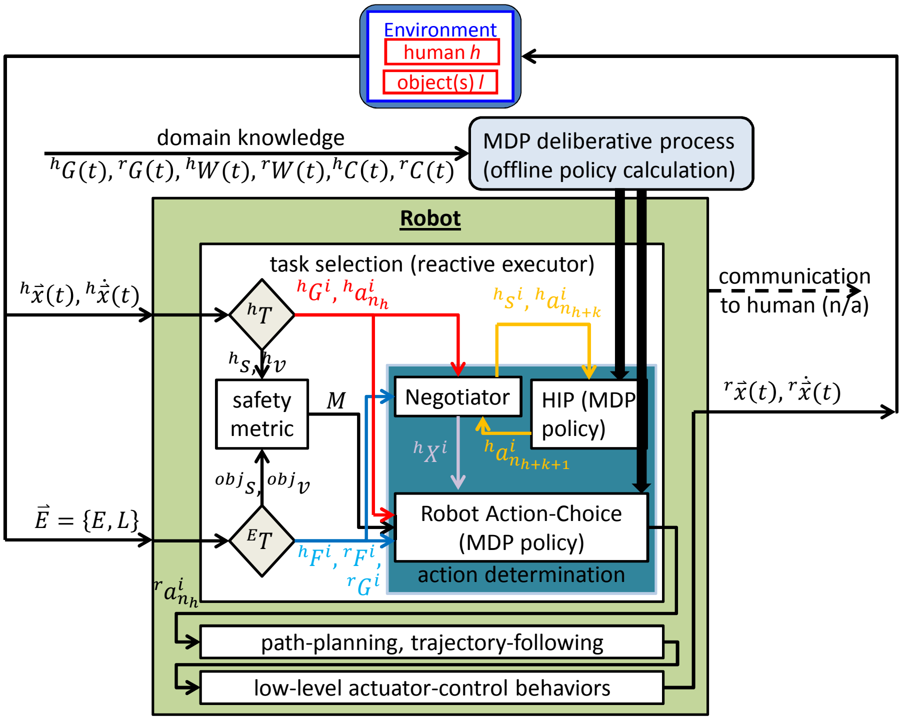

Environment
human h
object(s) l
domain knowledge
${}^{h}G(t), {}^{r}G(t), {}^{h}W(t), {}^{r}W(t), {}^{h}C(t), {}^{r}C(t)$
MDP deliberative process (offline policy calculation)
Robot
task selection (reactive executor)
${}^{h}\vec{x}(t), {}^{h}\dot{\vec{x}}(t)$
${}^{h}T$
${}^{h}G^{i}, {}^{h}a^{i}_{n_h}$
${}^{h}s^{i}, {}^{h}a^{i}_{n_{h+k}}$
communication to human (n/a)
${}^{h}s, {}^{h}v$
safety metric
M
Negotiator
HIP (MDP policy)
${}^{r}\vec{x}(t), {}^{r}\dot{\vec{x}}(t)$
${}^{h}X^{i}$
${}^{h}a^{i}_{n_{h+k+1}}$
obj_s, obj_v
$\vec{E} = \{E, L\}$
${}^{E}T$
Robot Action-Choice (MDP policy)
${}^{h}F^{i}, {}^{r}F^{i}, {}^{r}G^{i}$
action determination
${}^{r}a^{i}_{n_h}$
path-planning, trajectory-following
low-level actuator-control behaviors

Figure 1. Multi-layer Control Architecture for Physically-Proximal HRI with Human Intent Prediction

Before operations begin, the interaction scenario is evaluated and the domain knowledge is determined to set up the state space used in the MDP deliberative process. This includes the system goals and tasks, ${}^{h}G(t)$, ${}^{r}G(t)$, the system priorities and breakdown of work, ${}^{h}W(t)$, ${}^{r}W(t)$, and system constraints and conflicts, ${}^{h}C(t)$, ${}^{r}C(t)$. In all notation, left superscript *h* represents the human, *r* represents the robot, and *E* represents the environment. These sets of information inform the internal setup of the translator modules, ${}^{E}T$ and ${}^{h}T$, shown as grey diamonds in Figure 1, which convert the sensor data to a discrete-valued state variable form. The translators are also informed by human subject data, and the 3D workspace is segmented into a finite set of zones that become discrete state feature values for the MDP. The deliberative process is informed by human subject data to determine the model parameters for the human intent prediction (HIP) policy calculation, but the robot action-choice (RAC) policy is informed by translating the superset of robot and human goals, constraints, and conflicts into optimal policies. Both HIP and RAC policies are calculated offline, a viable approach until online model adaptation is activated.

During real-time operations, the system functions as follows: the robot's sensors read the continuous human state ${}^{h}\vec{x}(t)$, ${}^{h}\dot{\vec{x}}(t)$ and environmental data $\vec{E} = \{E, L\}$ from the environment. This data is sent to the translator modules which convert the sensor information into the values for the goal states ${}^{h}G^{i}$, ${}^{r}G^{i}$ and high-priority interruptive goal states ${}^{h}F^{i}$, ${}^{r}F^{i}$ of the human and robot, respectively. The human translator module also determines the current action the human is performing, ${}^{h}a^{i}_{n_h}$. These are sent to the executing MDP policies. The translator boxes also include physical dynamic models and supply the safety metric box with data to compute a danger index *DI* where *s* is the distance from the critical point to the nearest point on the person or obstacle object, respectively, and *v* is the magnitude of velocity component along the line between each of those points (called the approach velocity).[21,22] The discretized safety metric *M* generally helps to delineate 'closeness' between the physical human and robot appendages in same or different zones. The Negotiator box updates its internal action history of previously-known human action-choices, ${}^{h}A^{i}$, and passes this to the HIP policy along with ${}^{h}G^{i}$, ${}^{h}a^{i}_{n_h}$, ${}^{h}F^{i}$ as input; the policy outputs the predicted future intent, ${}^{h}a^{i}_{n_h+1}$. (The Negotiator could repeat this process to build up a vector of predicted future intent states farther into the future, if we want to know more predicted states past ${}^{h}a^{i}_{n_h+1}$. Alternately, the HIP could be expanded, but this would add further complexity. For now, we take k=0 for n_h+k and n_h+k+1, or as below, ${}^{h}n_x$=2.) The RAC policy takes as input from the Negotiator ${}^{h}a^{i}_{n_h}$, ${}^{h}a^{i}_{n_h+1}$, ${}^{h}F^{i}$, ${}^{r}F^{i}$, ${}^{r}G^{i}$ and outputs the most locally-optimal robot action-choice, ${}^{r}a^{i}_{n_h}$. The lower layers take that action-choice and pick the associated pre-scripted path for the robot to follow.

B. Deliberative Layer – Markov Decision Process Formulations

A general discrete time stochastic dynamic programming (SDP) problem – also known as a Markov Decision Process (MDP) – can be described as:[23,24]

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

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

where

$$\begin{gathered} p\left(s^j \mid s^i, a_k\right) = T\left(s^i, a_k, s^j\right), s^i \in S, s^j \in S, a_k \in A \\ satisfying \sum_{j\in\{1,\dots,n_s\}} T\left(s^i, a_k, s^j\right) = 1, \forall i \in \{1,\dots,n_s\}, \forall k \in \{1,\dots,n_a\} \end{gathered} \tag{3}$$

$$\begin{gathered} \forall g^i_k, g^i_k \rightarrow \left\{\left\langle a^1_k,\dots,a^{p_k}_k \right\rangle, \left\langle a^1_k, a^1_l, a^2_k \dots, a^{p_k}_k \right\rangle, \dots\right\}, a^x_k \in A, a^x_l \in A, p_k \le n_h \\ \forall f^i_k, f^i_k \rightarrow \left\{\left\langle a^1_k,\dots,a^{p_k}_k \right\rangle, \left\langle a^1_k, a^1_l, a^2_k \dots, a^{p_k}_k \right\rangle, \dots\right\}, a^x_k \in A, a^x_l \in A, p_k \le n_h \end{gathered} \tag{4}$$

This general form of the MDP transition probability function or tensor in Eq. (3) 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.

An action-sequence, as shown above in Eq. (4), is a particular action or ordered set of actions that the human must perform to satisfy a goal objective g_k^i or f_k^i. This ordered set is an n-tuple (action) sequence, where n=p_k is the number of actions corresponding to completing a goal. Note that 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. The value of n_h is consistent for each MDP model and chosen or otherwise optimized offline.

The Human Intent Prediction (HIP) MDP formulation, as given in Ref. 15, is as follows:

$$\begin{aligned} &S = \left\{s^1, s^2, \dots, s^{{}^{h}n_s}\right\} \\ &s^i = \left\{{}^{h}s^i\right\}, {}^{h}s^i = \left\{{}^{h}G^i, {}^{h}A^i, {}^{h}F^i\right\} \\ &{}^{h}G^i = \left\{{}^{h}g^i_1, {}^{h}g^i_2, \dots, {}^{h}g^i_{{}^{h}n_g}\right\}, {}^{h}g^i_k \in \{0,1\}, k = \left\{1,\dots,{}^{h}n_g\right\} \\ &{}^{h}F^i = \left\{{}^{h}f^i_1, {}^{h}f^i_2, \dots, {}^{h}f^i_{{}^{h}n_f}\right\}, {}^{h}f^i_k \in \{0,1\}, k = \left\{1,\dots,{}^{h}n_f\right\} \\ &{}^{h}A^i = \left\{{}^{h}a^i_1, {}^{h}a^i_2, \dots, {}^{h}a^i_{{}^{h}n_h}\right\}, {}^{h}a^i_k \in {}^{h}A, {}^{h}A = \left\{1,2,\dots,{}^{h}n_a\right\}, k = \left\{1,\dots,{}^{h}n_h\right\} \end{aligned} \tag{5}$$

$${}^{h}R(s^i) = \sum_{j=1}^{{}^{h}n_g} {}^{h}\gamma_j \, {}^{h}r_1\left({}^{h}g^i_j, {}^{h}A^i\right) + \sum_{j=1}^{{}^{h}n_f} {}^{h}\phi_j \, {}^{h}r_2\left({}^{h}f^i_j\right) - \sum_{j=1}^{{}^{h}n_f} {}^{h}\varphi_j \, {}^{h}r_3\left({}^{h}f^i_j\right) \tag{6}$$

Above, ${}^{h}A^{i}$ is the human's abbreviated action-choice history of n_h actions, ${}^{h}g_k^i$ and ${}^{h}f_k^i$ are normal and high-priority goal objectives, respectively, and $R({}^{h}s^{i})$ is the reward function.

The Robot Action-Choice (RAC) MDP formulation is given by:

$$
\begin{aligned}
&S = \left\{s^1, s^2, \ldots, s^{{}^r n_s}\right\} \\
&s^i = \left\{{}^r s^i, {}^h s^i\right\} \\
&{}^r s^i = \left\{{}^r G^i, {}^r \mathrm{H}^i, {}^r X^i, {}^r M^i\right\}, {}^h s^i = \left\{{}^h X^i, {}^r M^i\right\} \\
&{}^r G^i = \left\{{}^r g_1^i, {}^r g_2^i, \ldots, {}^r g_{n_g}^i\right\}, {}^r g_k^i \in \{0,1\}, k = \left\{1,..,{}^r n_g\right\} \\
&{}^r \mathrm{H}^i = \left\{{}^r \gamma_1^i, {}^r \gamma_2^i, \ldots, {}^r \gamma_{n_g}^i\right\}, {}^r \gamma_k^i \in \left\{0, \ldots, {}^r n_g\right\}, k = \left\{1,..,{}^r n_g\right\} \\
&{}^r X^i = \left\{{}^r a_1^i, \ldots, {}^r a_{{}^r n_x}^i\right\}, {}^r a_k^i \in {}^r A, {}^r A = \left\{1, \ldots, {}^r n_a\right\}, k = \left\{1, \ldots, {}^r n_x\right\} \\
&{}^h X^i = \left\{{}^h a_{{}^h n_h}^i, \ldots, {}^h a_{{}^h n_h + {}^h n_x - 1}^i\right\}, {}^h a_k^i \in {}^h A, k = \left\{{}^h n_h, {}^h n_h + {}^h n_x - 1\right\} \\
&{}^r M^i = \left\{{}^r z_1^i, \ldots {}^r z_{{}^r n_x}^i\right\}, {}^r z_k^i \in Z, k = \left\{1,..,{}^r n_x\right\} \\
&{}^h M^i = \left\{{}^h z_{{}^h n_h}^i, \ldots, {}^h z_{{}^h n_h + {}^h n_x - 1}^i\right\}, {}^h z_k^i \in Z, k = \left\{{}^h n_h, {}^h n_h + {}^h n_x - 1\right\} \\
&Z = \left\{1, \ldots, n_z\right\}
\end{aligned}
\tag{7}
$$

$$
{}^r R(s^i) = \left(\sum_{j=1}^{{}^r n_g} {}^r \gamma_j \, {}^r r_1\left({}^r g_j^i, {}^h X^i, {}^r X^i\right)\right) * noconflict({}^h X^i, {}^r X^i, {}^h M^i, {}^r M^i)
\tag{8}
$$

The RAC process is solved similarly to the HIP MDP, but for RAC the robot must model both itself and the human to select the optimal action for each state. There is no 'action history' required for RAC, the robot instead uses only the current set of ongoing and future-predicted actions to make its decision. Above, ${}^{h}X^{i}$ holds the current and future-predicted action-choices of the human obtained from the translator module and HIP policy, respectively, and are assumed to be correct. ${}^{r}X^{i}$ holds the current action that the robot is completing; ${}^{r}X^{i}$ is presumed known with certainty. ${}^{r}\mathrm{H}^{i}$ encapsulates the relative goal priority of each of the robot's goals. The reward function r_1, however, is more complex than in the HIP case. The safety-efficiency tradeoff occurs in RAC, as RAC must calculate the utility of a robot action occurring and whether a particular robot action would conflict with the current and future actions of the human. We use a *noconflict* parameter to weight the risk accordingly, or disallow the action completely.

IV. Case Study: Space HRI Domain Representation

In this paper, as in Ref. 15, we use a simple domain model with concentration and pick-and-place tasks that would be required for astronauts performing intravehicular activity (IVA) as well as on Earth. Specifically, we model an environment in which the human is engaged in problem-solving and interacting with a control panel (pressing buttons) but is also able to select nutrient consumption activities, with all tasks conducted at different reachable worksite locations. Our previous experiments in which a seated human executes these tasks in an environment shared by a fixed-based robot manipulator confirm HRI is feasible for this scenario.[16] We hypothesize that our basic simulation and experimental results will translate to models of humans performing similar activities in IVA in space. In our HRI scenario, the human is asked to type solutions to simple arithmetic problems as quickly and efficiently as possible while not overly concerning themselves with the robot's motion. The human is also asked to press buttons in response to sporadic events, as well as inserting actions to eat [chips] and drink [soda]. A robotic manipulator arm, operating in the same workspace, completes tasks at a fixed set of prespecified locations within the workspace.

The Human Intent Prediction (HIP) MDP representation is also as given in Ref. 15. Briefly summarized here, the state space is defined to be:

$$\begin{aligned}
&{}^{h}n_{g}=3,{}^{h}n_{f}=1,{}^{h}n_{h}\in\{1,2,3,4,5\},{}^{h}n_{a}=5\\
&s^{i}={}^{h}s^{i}=\left\{{}^{h}g_{1}^{i},{}^{h}g_{2}^{i},{}^{h}g_{3}^{i},{}^{h}f_{1}^{i},{}^{h}a_{1}^{i},\ldots,{}^{h}a_{n_{h}}^{i}\right\}\\
&{}^{h}g_{1}^{i}\in\{0,1\},{}^{h}g_{2}^{i}\in\{0,1\},{}^{h}g_{3}^{i}=\in\{0,1\}\\
&{}^{h}f_{1}^{i}\in\{0,1\}\\
&{}^{h}a_{k}^{i}\in\{1,2,3,4,5\}\\
&{}^{h}R(s^{i})={}^{h}\gamma_{1}{}^{h}g_{1}^{i}+{}^{h}\gamma_{2}{}^{h}g_{2}^{i}+{}^{h}\gamma_{3}{}^{h}g_{3}^{i}+{}^{h}\phi_{1}(1-{}^{h}f_{1}^{i})-{}^{h}\varphi_{1}{}^{h}f_{1}^{i}
\end{aligned}\tag{9}$$

Table 1. Domain Representation of actions ${}^{h}a_{k}^{i}$

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	
${}^{h}g_{1}^{i}$	0	1	?hunger? (sated)
${}^{h}g_{2}^{i}$	0	1	?thirst? (sated)
${}^{h}g_{3}^{i}$	0	1	?work_motivation? (sated)
${}^{h}f_{1}^{i}$	0	1	?button_1_active?

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

For reference, the equations that make up the domain-specific transition probabilities are given in Eq. (10) and Eq. (11):

$$\begin{aligned}
&{}^{h}\alpha_{x}=\left\{{}^{h}\alpha_{x}^{1},\ldots,{}^{h}\alpha_{x}^{{}^{h}n_{h}}\right\},x=\{1,\ldots,{}^{h}n_{g}\}\\
&{}^{h}P_{z,x}=p({}^{h}g_{z}^{j}=1|{}^{h}A^{i},{}^{h}a_{x})=\frac{\sum_{y=1}^{{}^{h}n_{h}}{}^{h}\alpha_{x}^{y}*({}^{h}a_{y}^{i}=={}^{h}a_{x})}{\sum_{y=1}^{{}^{h}n_{h}}{}^{h}\alpha_{x}^{y}}\\
&generally,{}^{h}P_{z}=p({}^{h}g_{z}^{j}=1|{}^{h}A^{i})=\frac{\sum_{x=1}^{{}^{h}n_{x}^{z}}p({}^{h}g_{z}^{j}=1|{}^{h}A^{i},{}^{h}a_{x})}{{}^{h}n_{x}^{z}}
\end{aligned}\tag{10}$$

$$\begin{aligned}
&{}^{h}\beta_{q}=\left\{{}^{h}\beta_{q}^{1},\ldots,{}^{h}\beta_{q}^{{}^{h}n_{f}}\right\},q=\{1,\ldots,{}^{h}n_{b}\}\\
&{}^{h}B_{z,k}=p({}^{h}g_{z}^{j}=1|{}^{h}s^{i},{}^{h}F^{i},{}^{h}a_{k})=\sum_{m=1}^{{}^{h}n_{f}}{}^{h}\beta_{q}^{m}*(1-{}^{h}f_{m}^{i}),q=f({}^{h}s^{i},{}^{h}g_{z}^{j})\\
&p({}^{h}g_{z}^{j}=1|{}^{h}s^{i},{}^{h}a_{k})=\frac{{}^{h}P_{z}+{}^{h}B_{z,k}}{1+{}^{h}n_{f}}\\
&generally:T({}^{h}s^{i},{}^{h}a_{k},{}^{h}s^{j})=\frac{\sum_{z=1}^{{}^{h}n_{T}^{j}}p({}^{h}g_{z}^{j}=1|{}^{h}s^{i},{}^{h}a_{k})}{{}^{h}n_{T}^{k}-1},{}^{h}s^{i}\to{}^{h}s^{j}\Rightarrow{}^{h}g_{z}^{j}\to 1
\end{aligned}\tag{11}$$

${}^{h}\alpha_{x}$ is a vector of weights that define the impact of an action ${}^{h}a_{x}$ in the action history on a probabilistic change in state, given how far back in time it occurred. ${}^{h}P_{z,x}=p({}^{h}g_{z}^{j}=1|{}^{h}A^{i},{}^{h}a_{x})$ is defined as the probability of goal objective ${}^{h}g_{z}^{j}$ being or becoming 1 (completed) due to occurrences of action ${}^{h}a_{x}$ in action history ${}^{h}A^{i}$ of state ${}^{h}s^{i}$. The variable ${}^{h}\beta_{q}$ is a group of weights that define the probability that the human choosing action ${}^{h}a_{k}$ for their next action will result in a goal-objective ${}^{h}g_{z}^{j}$ being/becoming 1 (completed) due to that action. If a high-priority interrupt flag is not set, then the probability ${}^{h}B_{z,k}$ of a choice of action ${}^{h}a_{k}$ completing a low-priority goal is added to the transition probability; otherwise, it is not included because the probability of completing a low-priority goal when a high-priority goal exists is 0.

For the RAC policy evaluation, we first created a baseline for comparison. For this, we considered our human-subject experiments discussed in Ref. 16 and elaborated upon in Ref. 14. In these preliminary human-robot shared workspace experiments, a simple algorithm was used by the robot for it action-choice – a first-in-first-out (FIFO) queue. Goals on the queue were removed once completed, and goals were *temporarily* skipped if they were 'blocked' due to a physical or mental conflict with the human (e.g., the robot physically blocks the human from reaching a target, or visually distracts within or occludes an essential viewing area). If a goal that has been previously postponed is no longer blocked, the robot immediately stops attempting to complete the lower-priority task set and instead executes the task set associated with completing the higher-priority goal. If no 'nonblocked' goal is found on the queue, the manipulator arm moves to a neutral unstowed position and waits there (no-op) until a nonconflicting goal-seeking task activates.

We specify a Robot Action-Choice (RAC) MDP representation for this work that parallels the conditional action-choice algorithm used in our original human subject experiments:

$$
\begin{aligned}
&{}^{r}n_{g}=3,{}^{r}n_{a}=4,{}^{r}n_{x}=1,{}^{h}n_{h}=4,{}^{h}n_{a}=5,{}^{h}n_{x}=1\\
&{}^{r}s^{i}=\left\{{}^{r}g_{1}^{i},{}^{r}g_{2}^{i},{}^{r}g_{3}^{i},{}^{r}\gamma_{1},{}^{r}\gamma_{2},{}^{r}\gamma_{3},{}^{r}a_{1}^{i}\right\},{}^{h}s^{i}=\left\{{}^{h}a_{4}^{i}\right\}\\
&{}^{r}g_{1}^{i}\in\{0,1\},{}^{r}g_{2}^{i}\in\{0,1\},{}^{r}g_{3}^{i}\in\{0,1\}\\
&{}^{r}\gamma_{k}\in\{0,1,2,3\}\\
&{}^{r}a_{k}^{i}\in\{1,2,3,4\},{}^{h}a_{k}^{i}\in\{1,2,3,4,5\}\\
&{}^{r}R({}^{r}s_{i})=\left(\sum_{j=1}^{3}{}^{r}\gamma_{j}\,{}^{r}r_{1}\left({}^{r}g_{j}^{i},{}^{h}X^{i},{}^{r}X^{i}\right)\right)*noconflict({}^{h}X^{i},{}^{r}X^{i})\\
&{}^{r}r_{1}\left({}^{r}g_{k}^{i},{}^{h}X^{i},{}^{r}X^{i}\right)={}^{r}g_{k}^{i}+(1-{}^{r}g_{k}^{i})*gettingthere({}^{r}g_{k}^{i},{}^{r}X^{i})\\
&noconflict({}^{h}X^{i},{}^{r}X^{i})=\begin{Bmatrix}if\,({}^{r}a_{1}^{i}==1),0,else,1,\forall{}^{h}a_{4}^{i}=3\\ if\,({}^{r}a_{1}^{i}==2),0,else,1,\forall{}^{h}a_{4}^{i}=1\\ if\,({}^{r}a_{1}^{i}==3),1,else,1,\forall{}^{h}a_{4}^{i}=2\\ 1,\forall{}^{h}a_{4}^{i}=4\end{Bmatrix}\\
&gettingthere({}^{r}g_{k}^{i},{}^{r}X^{i})=\begin{Bmatrix}if\,({}^{r}a_{1}^{i}==4),0.3,\\ elseif\,({}^{r}a_{1}^{i}==k),1,\forall k\in\{1,2,3\}\\ else,0\end{Bmatrix}
\end{aligned}
\tag{12}
$$

Table 3. Domain Representation of actions ${}^{r}a_{k}^{i}$

Discrete Value	Corresponding Action	Conflicts With
1	press_b1	math, ${}^{h}a_{k}^{i}=3$
2	press_b2	eat chip, ${}^{h}a_{k}^{i}=1$
3	press_b3	n/a (near ${}^{h}a_{k}^{i}=2$)
4	return_to_unstow	n/a

Table 4. Domain Representation of ${}^{r}g_{k}^{i}$ goal-objectives

Goal Obj.	Discrete Value		Corresponding Action
	false	true	
${}^{r}g_{1}^{i}$	0	1	?b1_inactive?
${}^{r}g_{2}^{i}$	0	1	?b2_inactive?
${}^{r}g_{3}^{i}$	0	1	?b3_inactive?

Tables 3 and 4 describe the robot's actions and goals used for our domain; the human's actions and goals are the same as in Tables 1 and 2 above. In Eq. (12), the ${}^{r}\gamma_k$ encapsulate the activation time of each goal, which is a direct mapping of the relative goal priority, i.e., how long the goal has been active on the FIFO goal queue.

Given our assumption that robot will always successfully complete an action that is executed to completion, the transition probability for robot action choice (RAC) MDP is 1, as shown in Eq. (13). Further, in our RAC MDP formulation, we do not interrupt action. We only set this to zero for goal states that cannot be reached by a given action taking place or a simple transition, but these are rare.

$$T(s^{i}, {}^{r}a_{k}, s^{j}) = 1 \qquad (13)$$

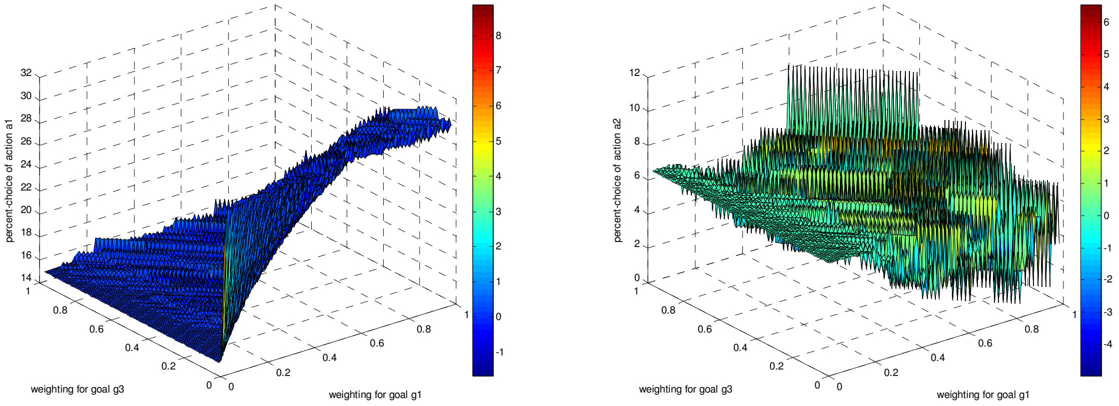
V. Simulation Results

The results presented in this section focus on evaluation of the HIP MDP, reserving evaluation of the RAC and integrated HIP/RAC MDP system for future work. We solved for the policies using value iteration over an infinite horizon with a discount factor of 0.95. We evaluated the HIP MDP formulation by varying parameter values for the weightings of two goals at a time and looking at the action-choices output by the process, in order to gain a better understanding of the impact of the reward function weightings on the optimal policy output. Our main evaluation metric is 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. First, we discuss the 'smoothness' of the HIP output in Table 5, which is a refinement of the results in Ref. 15 at finer variable parameter resolution, with Fig. 2 and 3 giving visual examples. We remove the boundary case 'edges' where only one term is being rewarded from the analysis, as the data is unrealistic to use for weights, and unhelpful in the context of tradeoffs. Next, we compare smoothness while looking at subsets of similar states and draw conclusions on the differences between this analysis and the full policy percent action-choice. Finally, we compare consistency between policy action-choices, analyzing how quickly the individual action-choices changes over the state-subsets.

A. HIP MDP Smoothness of Action-Choice Output – All States

We first looked at the 'smoothness' of the action-choice output at a much smaller delta variation of parameter values than Ref. 15: 0.01 instead of 0.25 on the range of [0 1].

Figure 2 below is indicative of what we see across multiple MDPs: there are local minima and maxima that look like 'rough' spikes, but overall trends emerge. We observe continuous and smooth trends in action-choice percentage across changes in the parameters. Referring back to Tables 1 and 2, note that in the Figure 2 case, ${}^{h}\gamma_1$ and ${}^{h}\gamma_3$ are the reward function weights for goal 1 and goal 3: sating hunger and work motivation, respectively. Looking at the parameters, action-percentages in the neighborhood of any particular point are actually very close to each other, despite the gradient jumps. In Figure 2, the trend follows what we would expect to see. The tradeoff is between action 1 to compete goal objective 1 (sating hunger) and action 3 to complete goal objective 3 (work motivation). Comparing the curved surfaces in the lower left and upper left plots, we see that action a_3 (computer work) use increases dramatically as the goal 3 weight increases, and similar for action a_1 (eating chips) and goal 1 weight. We also note the tradeoff between actions, where a_3 and a_1 have policy preference almost directly proportional to the tradeoff of reward weightings, an intuitive result given that a_3 has an equal chance of g_3 completion as a_1 has of g_1 completion ($\beta_5 = 0.25$ versus $\beta_1 = 0.25$).



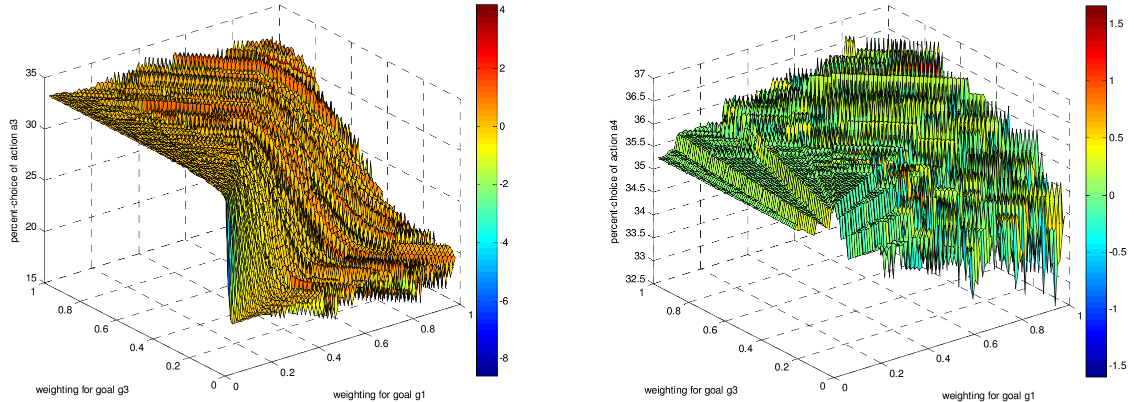


Figure 2. Percentage of action-choice across all states, MDP1 policy, g_1-g_3 tradeoff (right axis ${}^h\gamma_1$, left axis ${}^h\gamma_3$) (from upper left, clockwise: percent-choice a_1, percent-choice a_2, percent choice a_4, percent choice a_3)

Table 5 gives the summarized results.

Table 5. Parameter values for HIP MDP Evaluation (single number constant and/or ranges)

${}^hn_h$=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, $R(s_i)={}^h\gamma_1\,{}^hg_1^i+{}^h\gamma_2\,{}^hg_2^i+{}^h\gamma_3\,{}^hg_3^i+{}^h\phi_1(1-{}^hf_1^i)-{}^h\varphi_1\,{}^hf_1^i$					Comments on percentage of action choice for each action, per policy
	${}^h\gamma_1$ eat reward	${}^h\gamma_2$ drink reward	${}^h\gamma_3$ math reward	${}^h\phi_1$ button reward	${}^h\varphi_1$ button cost	Δ value variance	
MDP1	[0 1]	0	[0 1]	0	0	0.01	roughest looking of these; mostly smooth with <~4-6% gradients of for actions over the entire policy; long gradients with the tradeoff slopes changing +/- across ${}^h\gamma_3$ ~= ${}^h\gamma_1$
MDP2	[0 1]	[0 1]	0	0	0	0.01	looks slightly less rough than MDP1, but somewhat smooth / continuous; gradients and range similar to MDP1, tradeoff slopes changing +/- across ${}^h\gamma_2$ ~= ${}^h\gamma_1$ /5
MDP3	[0 1]	0	0	[0 1]	0	0.01	all look very smooth except in the range of ${}^h\phi_1$ =(0.01 0.1]
MDP4	[0 1]	0	0	0	[0 1]	0.01	similar smoothness and range as MDP3
MDP6	[0 1]	[0 1]	0.75	1	1	0.01	a_1 and a_2 trade off at decreasing ${}^h\gamma_1$ similar to MDP2; a_3 falls off nicely as ${}^h\gamma_1$ and ${}^h\gamma_2$ increase; a_4 steady at ~51.25%, a_5 steady at ~2.5%; all are smooth

As the action-percentages are calculated across all states, including those MDP representations that have no goal flags set for the rewarded cases, we suspected that some of the local variations were due to a lack of any driving force on those other states towards taking a particular action. When we look at the other cases – MDP2 through MDP4 – we see this trend confirmed with similar local maxima/minima spikes across all cases, as shown in Table 5 above. So, we look at MDP6 next – a case with fully-defined non-zero rewards and costs for every goal parameter.

Looking at Fig. 3, the MDP6 case, all plots look much smoother across the board than the previous MDP outputs, likely because the problem was underfitted before – having no weights on the other three goal terms generally means that those terms will oscillate between possible actions so long as there are no goal interdependencies. Note the bottom right image, which shows the rising-a_1/lowering-a_2 tradeoff quite well. Recall that the drinking action is set up as satisfying both goals because sometimes people feel hungry and want to eat (${}^{h}g_1$=0) when they are actually thirsty, so the drink action sates the hungry state. When comparing just the tradeoff between actions a_1 and a_2, a_2 is overall more preferable than a_1. This is because a_2 has an equal chance of only g_1 completion as a_1 (β_2 = 0.25 versus β_1 = 0.25), a high chance of only g_2 completion (β_3 = 0.75), and the same chance of completing both g_1 and g_2 together (β_4 = 0.25) as a_1 has of completing g_1 alone (β_1 = 0.25). (For transition function equation details see Ref. 15.) This is also shown in the policy: because of the interrelationship between the drinking action a_2 and both the hunger and thirst goals, when ${}^{h}\gamma_1$ and ${}^{h}\gamma_2$ are both high, making both goals considered rather important, it becomes a more efficient policy to drink (with a greater probability of satisfying both objectives with the given values of β) than to eat. Similarly, when the thirst goal is not as important relative to hunger, the number of eating actions a_1 tends to rise, but not quickly unless the thirst goal importance is ~1/5 or less that of hunger. This is supporting evidence that the MDP policy is encapsulating the probabilistic meaning of the β choices used in the model. Also, in the bottom-left image, as ${}^{h}\gamma_3$ is 0.75, a_3-choice is fairly high overall for other not-high weightings, but starts to decrease dramatically as the other weights pass that threshold of importance (0.75).

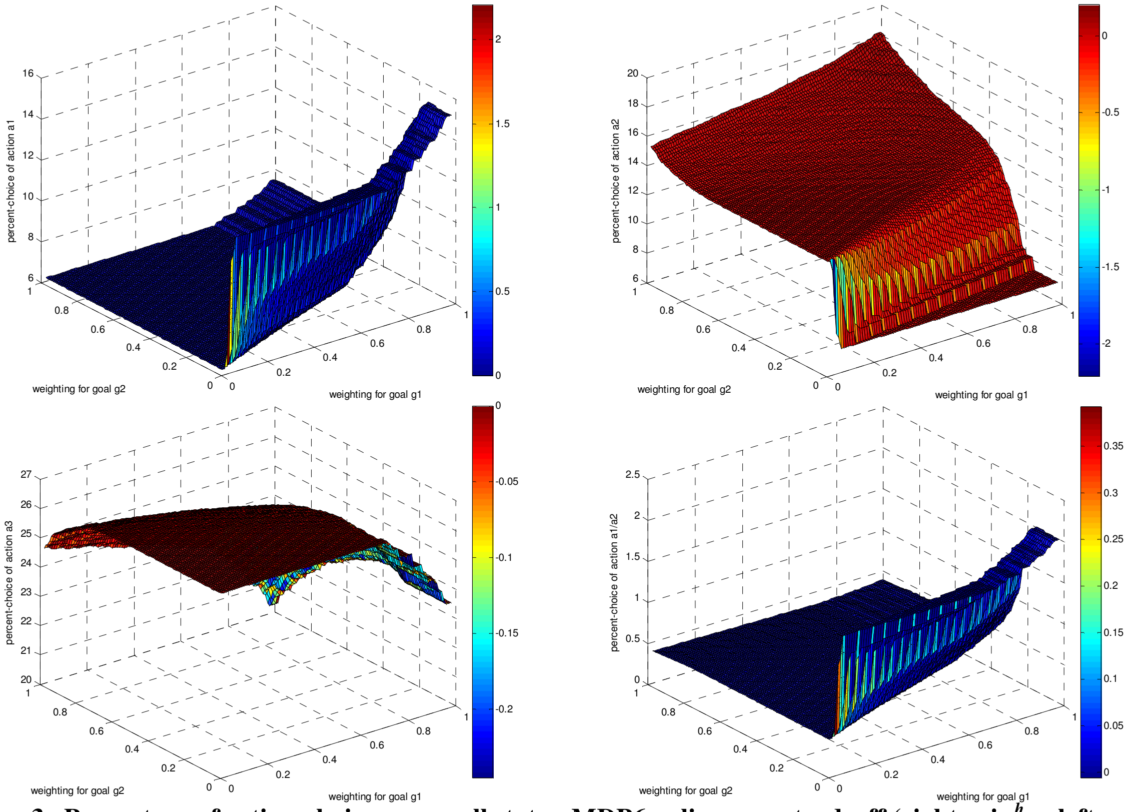


Figure 3. Percentage of action-choice across all states, MDP6 policy, g_1-g_2 tradeoff (right axis ${}^{h}\gamma_1$, left axis ${}^{h}\gamma_2$) (from upper left, clockwise: percent-choice a_1, percent-choice a_2, percent choice ratio a_1/a_2, percent choice a_3)

B. HIP MDP Smoothness of Action-Choice Output – Reward-Tradeoff Groups of Selected-States

In order to confirm or refute our suspicion regarding the action-choices and rewarded states, we decided to rerun the same sort of calculations as above, but instead split the calculation of action-percentages across groups of those states where the goal state for both goals in the rewards tradeoff are zero (no rewarded goals met) or one (both rewarded goals met), or only one rewarded goal is equal to zero (one rewarded goal satisfied), to refine the scope of our comparison. In those states, if the decision process has non-zero rewards, one would expect to see a majority of satisficing action-choices picked by the policy, and all local variations confined to those states that are given no guiding impact by the reward function. Additionally, for states where both the rewarded goals are both satisfied, we would expect a much more significant amount of variation across the board.

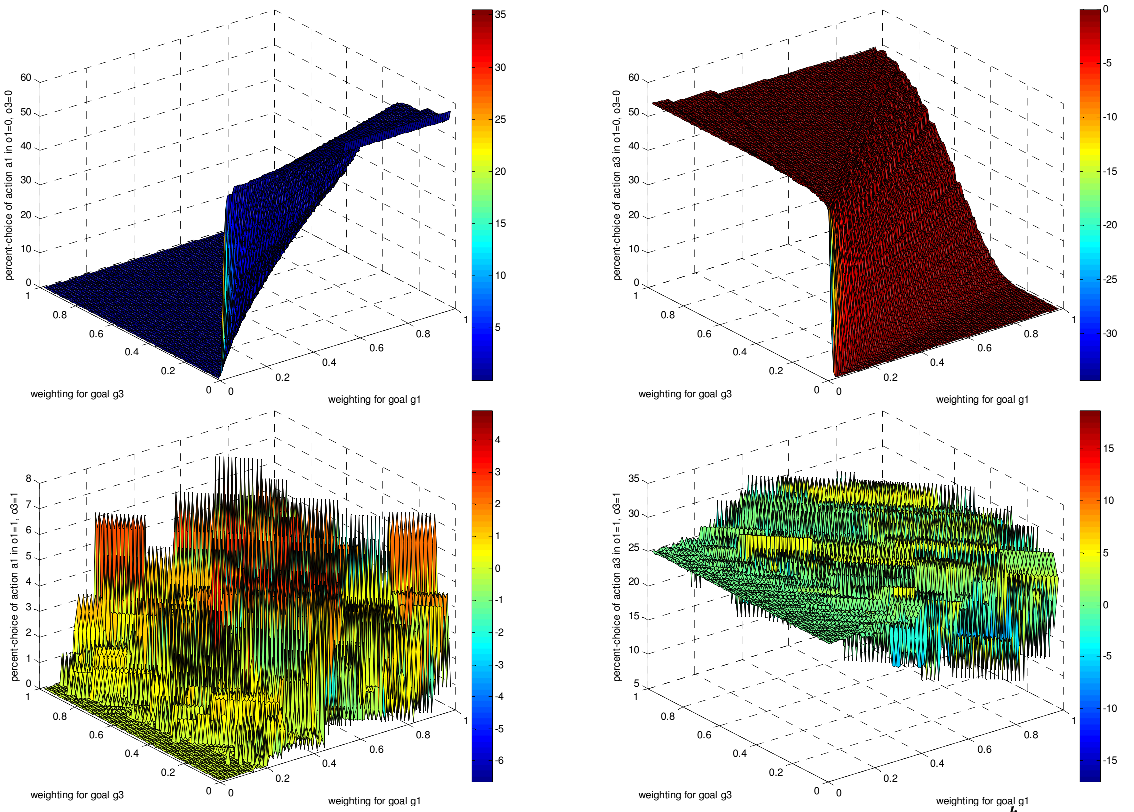


Figure 4. Percentage of action-choice, selected states, MDP1 policy, g_1-g_3 tradeoff (right axis ${}^h\gamma_1$, left axis ${}^h\gamma_3$) (from upper left, clockwise: percent-choice a_1 for {g_1=0,g_3=0}, percent-choice a_3 for {g_1=0,g_3=0}, percent-choice a_3 for {g_1=1,g_3=1}, percent-choice a_1, for {g_1=1,g_3=1})

As seen from the MDP1 policies in Figure 4, the majority of ‘noise’ in the percentage action-choice numbers seems to come arises from the already-satisfied goal states. Note that the action-choice percentages in Figure 4 are higher than in previous results because each of these subsets is ¼ of the total number of 10,000 states. Not shown above are the single-active-goal subset cases where the action-choice percentages were constant: the {g_1=0,g_3=1} case had a_1=59.92% and a_3=0%, and the {g_1=1,g_3=0} case had a_1=0% and a_3=53.92%. We also saw similar results for the other underfitted MDP formulations: the single-active-goal subset cases associated with the weight-tradeoffs tended to have flat action-choice percentages because there was a no-contest tradeoff in the given reward. For the MDP3 and MDP4 single-active-goal cases with {g_1=1,f_1=0}, a_1 vs. a_4 tradeoff (note that this state pair implies that the human is not hungry and the button does not need pressing), when the high-priority button-pressing goal f_1 is weighted less than 0.1 (positive for reward, negative for cost) and the sate-hunger goal g_1 is given more than 0.4 weight, the amount of lookahead required is sufficient that the MDP policy decides the person will start to choose eat and button actions preemptively. In other words, the action history seems to have a more significant impact here – recall from Eqn. 10 that the inclusion of more eat actions in the action history increases the probability of the hunger goal transitioning from active to inactive, and similar for button-pressing and its satisficing goal. Also, for cases where both these goals are active, {g_1=0,f_1=1}, when f_1 is weighted less than 0.1 (positive for reward, negative for cost) and the sate-hunger goal g_1 is given more than 0.1 weight, the mild increase in eat action-choice is directly proportional to the decrease in button-pressing. This implies that with a low reward weighting f_1’s edge of having a transition probability of 1 to secure that reward begins to be lost – as the likelihood of g_1’s transition to sated state increases, that state would be encountered more often and that weight would be added within the reward function more often. We were also able to see this interesting near-boundary behavior with the broader delta steps.

C. Policy Consistency – Direct Comparisons of State to Action-Choice

Finally, we examined whether the policy action choices were consistent for individual states. After all, comparing the number of times a particular action was recommended by two different policies is very different than being able to say that a certain percentage of the same states recommended the same action across policies. Quantitatively, we can compare the actions in each calculated action-policy to each other and track the changes in action-choice per state (row) – which ones, when, and to what. Visually, we can show this by plotting 2D 'slices' of the action-policy column vectors side-by-side and see when the actions shift to new integer values. Figure 5 is indicative of a common result, where the policy action-choice changes seem to 'creep' across the columns as the values of the reward weights shift incrementally.

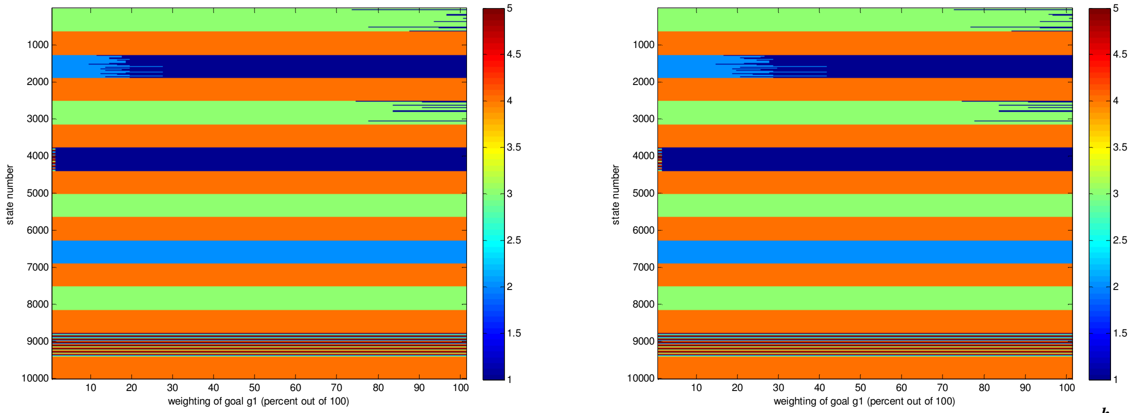


Figure 5. Policy action-choices for all states, a_k=[1 5], MDP6 policy, g_1-g_2 tradeoff (state number vs. $^h\gamma_1$*100) (Left: slice at $^h\gamma_1$=[0 1] at delta=0.01, $^h\gamma_2$=0.04; Right: slice at $^h\gamma_1$=[0 1] at delta=0.01, $^h\gamma_2$=0.06)

Figure 5 confirms that the action choices are consistent when comparing policy outcomes to other policies with 'similar' weightings within the delta value variance. It is feasible that these policies could update terms without drastic changes to the parameters. We could also search in the neighborhood of a nearby range of parameter solutions using a finer mesh for a closer-fitting solution when policy refinement is necessary. For these HIP models in particular, a delta of 0.1 appears to be a good starting point about which to fine a coarse match, and then further refine.

VI. Conclusions and Future Work

We have presented a framework for supporting autonomous human-robot interaction in a close-quarters collaborative setting, where maintaining safety is key. Such an environment would be present in an environment such as the space station. We have discussed why we think it is a viable approach to attempt to model human motion, recognize the human action, and convert this knowledge into an understanding of human intent for more intelligent future planning and task scheduling. We have evaluated human intent prediction (HIP) models in simulation and plan to integrate them into a robot's intelligent system framework for real-time use. We have discussed HIP simulation results in the context of policy consistency and sensitivity to varied reward function weightings.

Future work that builds on this simulation study will involve real-world human subject testing of this approach applied to a comparable laboratory-based human subject experiment using a safe robotic manipulator, where we will evaluate the system quantitatively and qualitatively through participant performance and feedback as well as with the briefly-discussed safety and efficiency metrics. We will iteratively improve the models with real human-subject experimental data and further test the HIP system around that parametric benchmark. We will also be exploring the challenges of converting the HIP MDP formulation to a POMDP formulation, as many of the goal-state flags for the human are considered internal, and could not be directly sensed without invasive measures. We will also examine differences that using a finite-horizon solver may have on the policy output and general human-robot operations. We will also conduct similar analyses of the RAC MDP alone and when integrated with the HIP MDP.

Acknowledgments

The authors would like to thank the undergraduate research students who helped develop our safe robot manipulator platform and who helped conduct human subject experiments. We also thank the graduate and undergraduate student volunteers who participated in the human subject testing that gave us the reference data that helped formulate and populate our MDP models. We would also like to thank Ali Nasir, whose help has been invaluable both in the creation of the original MDP software for policy calculation, and also in the determination of the MDP HIP formulation and representation.

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] Nikolaidis, S., and Shah, J., "Human-Robot Interactive Planning using Cross-Training: A Human Team Training Approach," *Proc. of Infotech@Aerospace*, Garden Grove, CA, June 2012.

[14] McGhan, C. L. R., and Atkins, E. M., "Human Productivity in a Workspace Shared with a Safe Robotic Manipulator," *Journal of Aerospace Computing, Information, and Communication*, accepted July 18, 2012 (to be published).

[15] McGhan, C. L. R., and Atkins, E. M., "Human Intent Prediction Using Markov Decision Processes," *Proc. Infotech@Aerospace Conference*, Garden Grove, CA, June 2012.

[16] 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.

[17] Wikipedia contributors, "Safety," URL: http://en.wikipedia.org/w/index.php?title=Safety&oldid=507198516 [cited 17 August 2012].

[18] Bonasso, R. P. *et al.*, "Experiences with an architecture for intelligent, reactive agents," *Journal of Experimental & Theoretical Artificial Intelligence, 9*, Vol. 2, No. 3, 1997, pp. 237-256.

[19] Gat, E., "On Three-Layer Architectures," in *Artificial Intelligence and Mobile Robots*, David Kortenkamp, R. P. B. a. R. M. ed., AAAI Press, 1997, pp. 195-210.

[20] Montemerlo, M., Roy, N., and Thrun, S., "Perspectives on standardization in mobile robot programming: The Carnegie Mellon navigation (CARMEN) toolkit," *Proceedings of 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2003. (IROS 2003).*, Vol. 3, 2003, pp. 2436-2441.

[21] Kulic, D., and Croft, E. A., "Real-time safety for human-robot interaction," *Robotics and Autonomous Systems*, Vol. 54, No. 1, 2006, pp. 1-12.

[22] Kulic, D., and Croft, E., "Pre-collision safety strategies for human-robot interaction," *Autonomous Robots*, Vol. 22, No. 2, 2007, pp. 149-164.

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

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