Social Feedback For Robotic Collaboration

Emily Wu, Nakul Gopalan, James MacGlashan, Stefanie Tellex, and Lawson L.S. Wong
Brown University

Abstract—Interacting with a robot is a challenging task: difficulties with robotic perception and in knowing what the robot is currently thinking prevent human collaborators from providing the information the robot needs to complete the task quickly and accurately. In order to address these problems, robotic collaboration must include not just communication from the human to the robot, but also from the robot to the human. We call this robot-to-human communication *social feedback*. In order to flexibly and intelligently generate social feedback, we present a model based off a Partially Observable Markov Decision Processs (POMDP) called a β-POMDP model that incorporates the human's belief about the robot's intent. Doing so incentivizes the robot to communicate its current state using language and gesture in order to allow the human participant to make informed decisions about their next action. Initial evaluation in simulation shows decreased interaction time in a simulated domain while maintaining near-perfect accuracy.

I. INTRODUCTION

Collaboration is a process that relies heavily on communication for its success. When humans collaborate, this communication is both obvious and implicit—we not only instruct and request aid from each other, but we also tacitly monitor our partners for signs of approval and understanding while producing these signals ourselves. However, in robotics, many of these vital components of successful communication are missing or lost. The absence of these signals likely account for numerous failures in human-robot collaborative tasks. In order to provide robots these missing communicative skills, we employ *social feedback* signals to provide human-like communication to both inform the human participant of the robot's state and request information from the participant. In this paper, we describe a framework that describes how robotic agent should likewise generate social feedback for its human partner in human-collaborative tasks, and show that it improves the speed and accuracy of human-robot interaction.

Many existing works focus on the task of interpreting human communications[20, 16, 21, 17], but provide no response if the interpretation fails. As a consequence, it becomes difficult to tell whether the robot is still making progress in interpreting the request, or has failed. Some works do allow for robotic responses to ambiguous situations, but the manner of their responses are hard coded, often limited to generic requests, such as "please repeat the question." In this work, we provide a means to generate these requests in the form of language and gesture that can flexibly represent the robot's state as well as inform the human in such a way so that allows the human to help the robot as much as possible.

Fig. 1. The experimental setup from the participant's perspective: a Baxter robot has six objects within reach. The participant stands in front of the table to provide speech and gesture requests

We will solve this problem by formulating it as a Partially Observable Markov Decision Process (POMDP)[10], which will allow us to dynamically and flexibly determine how to choose social feedback actions. First, introduce a two-agent POMDP model called a β-POMDP. We can then reduce this model to a POMDP that is improved over existing robot interaction models by the inclusion of a state variable that tracks the human's interpretations of the actions the robot takes.

Specifically, we address the object delivery task. In this task, a set of objects are laid out on a table within a Baxter robot's reach. A human participant requests an object from the robot using speech and gesture (pointing). The robot must interpret the human's speech and gesture and deliver the requested object to the human. The task then repeats with the remaining objects on the table. This task is achievable without social feedback: the robot need only wait until enough information is given and then deliver the correct object. However, we will show that by adding social feedback actions, such as asking questions, looking at objects, and pointing at objects, we will achieve better accuracy and speed as well as improved user experience.

As an initial evaluation, we run our model in simulation to showcase improved speed while maintaining accuracy in the object delivery task.

In the object-delivery task, our social-feedback framework allows the task to be completed more quickly by allowing the robot to be an active participant in the collaboration. For example, if the human asks the robot for a bowl, but there are many bowls of different colors available, if the robot communicates that it knows the human wants a bowl, this lets the human know it must next specify the color of the bowl. An in-depth explanation is provided in figure 2.
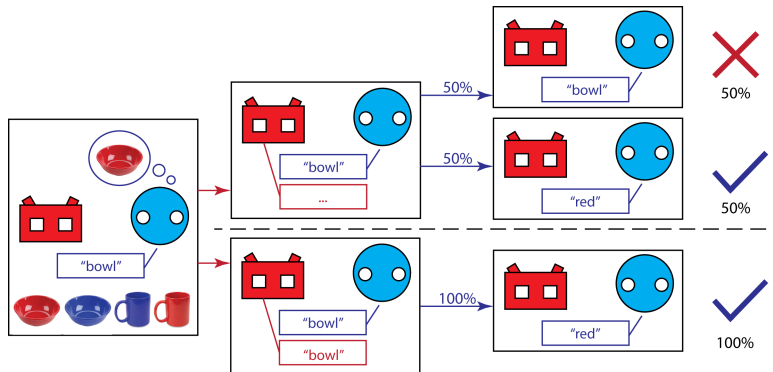
Fig. 2. The robot has four objects within reach: a red bowl, a blue bowl, a red cup, and a blue cup. Of the four objects, the human desires the red bowl. They tell the robot they would like a bowl. In response, the robot can choose to take an action or wait. If the robot chooses to wait, the human's belief about what the robot knows remains uniform. According to the human, it is equally helpful for them to tell the robot the color of the bowl is red or to repeat themselves and say they would like a bowl. However, if the robot chooses to inform the human that the robot knows they would like a bowl, the human now knows that the best observation they can take is to inform the robot that the object they would like is red.

II. RELATED WORK

This work is built off of Whitney et al. [23], which describes a system for incremental speech and gesture recognition in an object-delivery domain. The model from Whitney et al. [23] is expanded from a Bayesian inference model into a POMDP model that can ask yes-no questions in Wu et al. [25]. In this work, we generalize the idea of the affect of robot's actions on humans state within a POMDP framework.

Work demonstrating the importance of social feedback in human-human communication has been done in the field of psycholinguistics. In Clark and Krych [5], one human (labeled the builder) builds a Lego model according to instructions given by another human (labeled the director). In the feedback-free trials, the director's instructions were prerecorded, and the resulting models were very inaccurate (in fact no model was completely correct). In the feedback trials, errors were reduced by a factor of eight. Our goal is to enable a robot to collaborate with a human in this way.

Other work with collaborative robots exists, for example, Foster et al. [6] which describes research done with a bartending robot and Bohus and Horvitz [2], where a robotic building guide directs guest to specific rooms. We focus on collaboration within the domain of object delivery, with a focus on avoiding rule-based methods in favor of balancing information collection with exploitation via a POMDP.

Many other works have addressed the object delivery domain. Some approaches have no social feedback and will either deliver the wrong item or do nothing if given a request it does not understand [20, 16, 21, 17]. Language only feedback models also exist [3, 14, 20, 16, 8, 9, 17], and several gesture only models [22, 15]. We supplement gesture and language interpretation with social feedback, allowing the robot to respond.

In the field of computational linguistics, Chai et al. [4] provides work towards resolving referring expressions in a different domain, but does not address the task of acting on the results of these referring expressions. In Kruijff

et al. [12], they propose a system for planning to ask for clarifications, which covers a wide scope of knowledge failures. In this work, we are interested only in a small subset of these clarifications, and address the problem of how and when these clarifications should be used in a concrete human-robot collaboration task.

POMDP approaches to dialog [26] are quite common, but treat dialog as a discrete, turn-taking interaction. The Dialog State Tracking Challenge [24] a notable driving force for computer dialog understanding, treats dialog in this turn-based way. Our model is also based off a turn-based interaction, but is designed such that it would be suitable for a more incremental approach given the appropriate framework.

Interactive POMDPs (I-POMDPs) [7] describe a multi-agent POMDP that allows agents to anticipate each other's actions by forming beliefs over action and observation histories. This work attempts to accomplish the same, but makes more specialized assumptions that allow for a more tractable state space.

Alternative approaches to POMDPs include cognitive architectures such as SOAR [13] or DIARC [19]. By taking a probabilistic approach, we can seamlessly fuse information from multiple sources and explicitly reason about the robot's uncertainty when choosing actions.

III. TECHNICAL APPROACH

We begin by describing our two-agent β-POMDP, and then construct a typical POMDP motivated by the structure of our β-POMDP. We describe the observation function and transition function of the POMDP with respect to the β-POMDP.

A. POMDP Overview

Markov Decision Processes (MDPs)[1] are models that describe how an agent can take actions to transition between states, receiving rewards for its actions. Partially Observable Markov Decision Processes (POMDPs) are used to model

MDPs where the true state is not known. Instead, the agent receives observations that are generated by the true state, and must infer what the true state is from the observations. Thus, the agent maintains a belief over true states which is updated as it receives new observations. In subsequent sections, we will refer to this belief over hidden states as b. The agent must then use this belief state to determine which action to take to maximize its expected value of the reward over time.

B. β-POMDP

In order to model human-robot interaction, we make the following assumptions.

- Both agents model uncertainty over the other agent's intentions and choose actions according to a POMDP.
- Actions the human takes are interpreted as observations by the robot. Similarly, actions taken by the robot are interpreted as observations by the human.

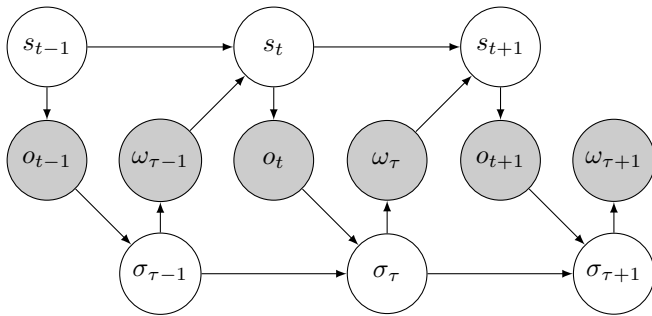This allows us to construct the graphical model depicted in figure 3.

Fig. 3.   The two-agent β-POMDP model.

We represent human states s_t, robot states σ_τ, and observations generated by the robot ω_τ and by the human o_t. The upper POMDP models the interaction from the robot's perspective: the human's state s_t is hidden from the robot, and the robot must infer it from observations o_t. The lower POMDP is the POMDP from the human's perspective: the robot has some hidden states σ_τ, which the human observes by means of ω_t. The human takes actions o_t to influence the robot's state σ_t. When the robot takes action $\omega_{\tau-1}$, it affects the human's state s_t. Because each agent treats the other agent's action as an observation, when the robot takes an action, it will influence the human's belief about the robot's hidden state, and vice versa. Thus, the human's actions affect the robot's belief about the human's state, which is what we call b above. Importantly, the reverse is also true: the robot's actions affect the human's belief about what the robot's hidden state is. We will call the human's belief over the robot's hidden state β.

In the following section we will use this dual structure to inform the construction of our POMDP as applied to our object delivery domain.

C. POMDP Definition

We define our object-delivery POMDP as a tuple $\{S, A, T, R, \Omega, O\}$:

- Each $s \in S$ is a tuple of $\langle \iota, \beta, \mathcal{I} \rangle$
  - $\mathcal{I}$ is the set of all objects that the robot can deliver. Each object is parameterized by a name, a unigram vocabulary, and a position; for example: a red bowl would be represented $\langle$redBowl, [red, red, bowl, bowl, plastic], (1.0, 2.0, 0.0)$\rangle$. We assume the set of all objects is known.
  - $\iota \in \mathcal{I}$ is the object the human desires. This is a hidden variable.
  - β is a distribution over the robot's hidden states, as defined above. In this domain, the robot's hidden state contains which object the robot will hand the human, or which object the robot believes the human wants. We assume the human estimates the distribution over the hidden state by observing the actions the robot has taken. Specifically, $\beta_t(\iota_t) = p(\iota_t | a_{1:t-1})$. While this variable is technically unknown, we make some assumptions about the transition functions and initialization so β_t's value is known at every time step.

  Since our state is composed of both hidden and known states, it resembles a Mixed Observability MDP [18], and we leverage the computational benefits of having both hidden and known states.

- The set of actions A consists of both social and non-social actions. Non-social actions are pick(i) (picking up and delivering an object i) and wait. Social actions are point(x), pointing at a location x; look(x), looking at a location x; and say(p), informing the human that the robot believes the desired object has property p, where p is an element of some object's vocabulary.

- $T(s, a, s') = p(s'|s, a)$. We make the assumption that the human's desired object ι does not change unless the robot picks up object ι. The set of all objects $\mathcal{I}$ transitions to $\mathcal{I} \setminus \{i\}$ when the robot chooses action pick(i). For each action in A, we define a "reverse observation function" that describes our assumptions about how the human imagines the robot generates actions (which the human sees as observations) given a hidden robot state. This allows us to define a transition function for β. The transition dynamics are described in greater detail in section III-C2.

- $R(s, a, s')$ returns a numeric reward from transitioning from s to s' by taking action a. We incentivize our robot to pick the correct object by giving it a $+10$ reward if it delivers the correct object and a -50 reward for picking the incorrect object. We also give negative rewards for taking various actions: wait receives a -1 reward (to incentivize the robot to finish the task quickly); look(x) receives a reward of -2; say(p) receives a reward of -3; point(x) receives a reward of -4. These additional penalties for social actions reflect the

penalty for "bothering the user", as well as the time it takes to execute these actions. These reward values were chosen experimentally to result in a high rate of pick accuracy.

- Observation $o \in \Omega$ represents an observation generated by the human. These are tuples of language and gesture: $\langle l, g \rangle$. Language utterance l is represented by a string of any number of words, obtained by transcribing microphone input using webkit's speech recognition API through Google Chrome. A gesture g is represented by a vector from the participant's shoulder to their wrist, and all gestures are interpreted as a straight-armed point. This vector is obtained using the Kinect's tracking software.

- $O(o, s, a) = p(o|s, a)$ describes the probability of of seeing an observation o from the human given their state s and the robot's last action a (though assume all observations are independent of a). We choose an observation function that reflects that the human is an agent attempting to communicate which object they desire to the robot, and thus chooses to generate observations that are more likely to result in the robot delivering the correct object. This is gone into more detail in the next section, section III-C1.

For quick reference, a table of variables is provided in figure 4.

1) Observation Function: According to our double-agent model, the human emits observations as though it were an agent interacting with our robotic agent. Thus, we choose an observation model that depends on the human's belief about the robot's state, β. Specifically, the human will choose an action according to its estimate that the robot will hand them their desired object. In order to define this function, we will first have to define a base-level observation function.

a) Base-Level Observation Function: The base level observation function describes the probability of an observation conditioned only on the object: $p(o|\iota)$. For our object delivery domain, we will define two base-level observations, one for language and one for gesture. These observation functions are the ones defined in Whitney et al. [23].

Speech Model: Language is interpreted according to a smoothed unigram speech model. An utterance l is broken down into individual words, $w \in l$:

$$p(l|\iota) = \prod_{w \in l} p(w|\iota) = \prod_{w \in l} \frac{\texttt{count}(w, \iota.\texttt{vocab}) + \alpha}{|\iota.\texttt{vocab}| + \alpha |\texttt{words}|}$$

where $\texttt{count}(w, \iota.\texttt{vocab})$ is the number of times word w appears in ι's vocabulary. This allows for repeated words to have greater probability of being spoken.[1] α is a smoothing value chosen to be 1. $|\texttt{words}|$ is the total number of words in all object vocabularies.

---

[1]In the future, object vocabularies will be collected from data; repeated words are therefore meaningful as ways that are commonly used to describe objects.

Gesture Model: All gestures are interpreted as a straight armed point. These pointing gestures are selected from a normal distribution centered at the object's location.

Define the angle between the vector defined by the pointing gesture and the vector from the human's shoulder to the object ι to be θ_ι. The probability of a particular gesture is then

$$p(g|\iota) = \mathcal{N}(\theta_\iota | 0, v)$$

where v is a hand-tuned variance. We choose $v = 0.4$.

b) Posterior Observation Function: We will use the base-level observation function defined above to define a posterior observation function that considers how the base-level observation function will affect the robot's belief. Specifically, because we assume the human is solving a POMDP the same way the robot is, they will pick the most optimal action. Instead of calculating what the most optimal action is, we will use a heuristic: the best action for the human to take (or the best observation for them to generate) is the action that results in the greatest probability of the robot handing them their desired object ι.

$$p(o|\iota, \beta) = \begin{cases} 1 & \text{if } o = \operatorname{argmax}_o p(\iota|o) \\ 0 & \text{otherwise} \end{cases}$$

To calculate $p(\iota|o)$, we can use Bayes rule:

$$p(\iota|o) = \eta \frac{p(o|\iota)p(\iota)}{\sum_{\iota'} p(o|\iota')p(\iota')}$$

where η is a normalization factor.

Next, we will set $p(\iota) = \beta(\iota)$. $p(\iota)$ describes the robot's belief in ι. This value is represented in the robot's belief-state vector, b. But since the human does not know b, it uses its estimate of b, β. Our new expression is

$$p(i|o) = \eta \frac{p(o|\iota)\beta(\iota)}{\sum_{\iota'} p(o|\iota')\beta(\iota')}$$

To account for suboptimal behavior and avoid irregularities in our observation function, we soften our optimality assumptions such that $1 - \epsilon$ fraction of the time, the human chooses the optimal action. With probability ϵ they pick according to the base-level observation function described above.

$$p(o|\iota, \beta) = \begin{cases} (1 - \epsilon) + \epsilon p(o|\iota) & \text{if } o = \operatorname{argmax}_o p(\iota|o) \\ \epsilon p(o|\iota) & \text{otherwise} \end{cases}$$

Using β as a prior for our observation function incentivizes the robot to take actions that modify β in such a way that it increases the probability of useful observations, while in turn decreasing the probability of getting observations that tell the robot things it already knows. This result falls out of our assumption that the human is more likely to provide observations that communicate its intentions to the robot well. An example is available in figure 2.

Variable	Explanation
$s = \langle \iota, \beta, \mathcal{I} \rangle$	A single state, defined by a tuple
$\mathcal{I}$	The set of all objects
$\iota \in \mathcal{I}$	The object desired by the human
β	A distribution over $\mathcal{I}$. $\beta_t(i_t) = p(i_t \vert a_{1:t-1})$
$i = \langle \text{name, vocab, location} \rangle$	An object, defined by a name, vocabulary, and location
a	A robot action, `pick(i)`, `look(x)`, `point(x)`, or `say(x)`, observed by human
$o = \langle l, g \rangle$	An observation received by the robot, generated by the human.
l	A string of language.
g	A pointing gesture.
ϵ	The probability of picking a non-optimal observation.
L	An indicator variable that is 1 if language is observed.
G	An indicator variable that is 1 if gesture is observed.
c_L	A constant that describes the probability of observing language.
c_G	A constant that describes the probability of observing gesture.
α	A smoothing parameter for the unigram model.
$p(a_t \vert i_{t+1})$	A reverse observation function
A_t	A diagonal matrix describing the reverse observation function $p(a_t \vert i_{t+1})$ for all values of i_{t+1}
T	A matrix describing all transition probabilities $p(\iota_{t+1} \vert \iota_t)$

Fig. 4.   Table of Variables

2) Transition Function: We make the assumption that every component of our state transitions conditionally independent of the other components given the previous state. In addition, each state variable depends only on its previous value and the action taken.

$$p(\iota_{t+1}, \beta_{t+1}, \mathcal{I}_{t+1} \vert \iota_t, \beta_t, \mathcal{I}_t, a_t)$$
$$= p(\iota_{t+1} \vert \iota_t, a_t) p(\beta_{t+1} \vert \beta_t, a) p(\mathcal{I}_{t+1} \vert \mathcal{I}_t, a_t)$$

a) Object Transition Function: As previously stated, we assume the object the human desires does not change unless its desired object is picked up. If the desired object is picked, it transitions uniformly at random between the remaining objects.

If a_t is `pick(`ι_t`)`,

$$p(\iota_{t+1} \vert \iota_t, a_t) = \begin{cases} 1/(\vert \mathcal{I} \vert - 1) & \text{if } \iota_{t+1} \in \mathcal{I} \text{ and } \iota_{t+1} \neq \iota \\ 0 & \text{otherwise} \end{cases}$$

If a_t is any other action:

$$p(\iota_{t+1} \vert \iota_t, a_t) = \begin{cases} 1 & \text{if } \iota_{t+1} = \iota_t \\ 0 & \text{otherwise} \end{cases}$$

b) Object Set Transition Function: The set of objects $\mathcal{I}$ changes only when an object is picked up. The picked object is removed from the set.

If a_t is a `pick(i)` action:

$$p(\mathcal{I}_{t+1} \vert \mathcal{I}_t, a_t) = \begin{cases} 1 & \text{if } \mathcal{I}_{t+1} = \mathcal{I}_t \setminus \{i\} \\ 0 & \text{otherwise} \end{cases}$$

If a_t is any other action:

$$p(\mathcal{I}_{t+1} \vert \mathcal{I}_t, a_t) = \begin{cases} 1 & \text{if } \mathcal{I}_t = \mathcal{I}_{t+1} \\ 0 & \text{otherwise} \end{cases}$$

c) Belief Transition Function: The human's belief about which object the robot will hand over, β_t, is informed by the actions the robot takes. In the same way that the robot's belief state b is a summary of all the observations the robot has made [10], β_t is a summary of all observations the human has made of the robot's actions, and reflects the human's estimate about the robot's true state. We therefore update β_t according to Bayesian probabilities in the same way that b is updated.

In a standard Markovian update, for a particular state ι_t, its probability after t observations is given as follows:

$$p(\iota_{t+1} \vert o_{1:t+1}) = p(o_{t+1} \vert \iota_{t+1}) \sum_{\iota_t} p(\iota_{t+1} \vert \iota_t) p(\iota_t \vert o_{1:t})$$

Recall that β is a distribution over states (objects) ι_t, so we can use this rule to write an update for each entry of β.

$$\begin{aligned} \beta_{t+1}(\iota_{t+1}) &= p(\iota_{t+1} \vert a_{1:t}) \\ &= p(a_t \vert \iota_{t+1}) \sum_{\iota_t} p(\iota_{t+1} \vert \iota_t) p(\iota_t \vert a_{1:t-1}) \\ &= p(a_t \vert \iota_{t+1}) \sum_{\iota_t} p(\iota_{t+1} \vert \iota_t) \beta_t(\iota_t) \end{aligned}$$

This requires us to specify $p(a_t \vert \iota_{t+1})$. Recall that actions are operating as observations from the perspective of the human, making this expression a *reverse observation function*. We must provide a reverse observation function for each action. These will often be very similar to the base-level observations described in Section III-C1, though adapted slightly to suit being performed by a Baxter robot. See section III-C2d for details.

We must also specify a transition function, $p(\iota_{t+1} \vert \iota_t)$. We will use the object transition function described earlier.

We can rewrite our element-wise update for the whole β vector using a matrix multiplication:

$$\beta_{t+1} = A_t T \beta_t$$

where A_t is a matrix representing the reverse observation function $p(a_t|\iota_{t+1})$ and T is a matrix representing the transition function $p(\iota_{t+1}|\iota_t)$. If the set of items $\iota = \{\iota_1, \iota_2, \ldots, \iota_n\}$, A is a diagonal matrix such that

$$a_{ij} = \begin{cases} 0 & \text{if } i \neq j \\ p(a_t|\iota_{t+1}) = \iota_i & \text{otherwise} \end{cases}$$

and T is a matrix with entries

$$t_{ij} = p(\iota_{t+1} = \iota_i | \iota_t = \iota_j)$$

This gives us a deterministic transition function for β:

$$p(\beta_{t+1}|\beta_t, a_t) = \begin{cases} 1 & \text{if } \beta_{t+1} = A_t T \beta_t \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

If we assume that β_0 begins as the uniform distribution over objects, we can now calculate β_t for any given time step deterministically. However, there are many assumptions that were made, and there is additional expressiveness that can be used by allowing for non-deterministic updates to β. However, we lead the exploration of this space for future work.

d) Reverse Observation Functions: We define a reverse observation function for each action the robot can take. These define how β is updated when a robot takes an action.

For `look(x)` actions, we define an angle ϕ_l to be the angle between a vector from the robot's face to the location x and the vector from the robot's face to the location of object ι_{t+1}. We assume this angle is chosen from a normal distribution centered at 0 with a hand tuned variance, v_l.

$$p(a_t = \texttt{look(x)}|\iota_{t+1}) = \mathcal{N}(\phi_l|0, v_l)$$

For `point(x)` actions, we define an angle ϕ_p to be the angle between the vector from Baxter's wrist to location x and the vector from Baxter's wrist joint to the object ι_{t+1}. This gesture is chosen according to a normal distribution with hand-tuned variance v_p.

$$p(a_t = \texttt{point(x)}|\iota_{t+1}) = \mathcal{N}(\phi_p|0, v_p)$$

Actions `say(p)` take the form of a statement "I believe that the object you want is p", where p is some property from some object's vocabulary. The probability of this action given some ι is identical to the speech model for the human, given in section III-C1a.

IV. Evaluation

We use Belief Sparse Sampling [11] (BSS) to solve our domain to evaluate the efficacy of our model in simulation. We run 25 trials, results for which are presented in figure 5. We see a reduction in the number of actions needed until the robot can successfully pick the correct object, while accuracy remains the same. With only six objects, it is easy to identify the correct object with enough observations, accounting for the high accuracy in both trials. However, taking social feedback actions allows the agent to request

the observations that it needs from the human to complete the interaction as quickly as possible. Notice also that the variance in the number of actions the robot takes until the first correct pick is very small in the social trials, while it is very large in the non-social trials. Social feedback actions allow the agent to control the interaction so that it is as brief as possible.
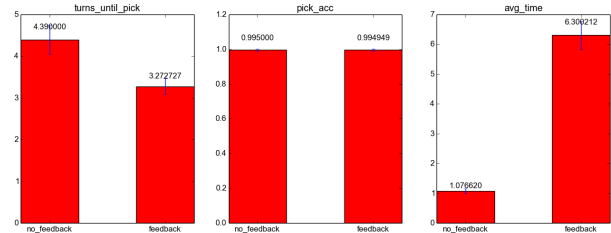

Fig. 5. Results in simulation. `turns_until_pick` is the number of timesteps until first correct pick, `pick_acc` is the ratio of correct picks to total picks, `avg_time` is the time to compute a single action

V. Future Work

The next task will be to run user-studies with human participants to compare the performance of the system with and without social feedback. This will require the use of approximate solvers to allow us to solve the POMDP in real time, or to precompute policies.

In the future, we will investigate more sophisticated observation functions, allowing for language interpretation that can understand prepositional phrases and other ways of composing referring expressions.

Other areas for further exploration can be found by removing many of the assumptions made in this paper. For example, the set of objects on the table was assumed to be known. Attempting to learn from the human more about the set of objects, such as their vocabularies and location, would be an interesting expansion.

Use of the β variable can be expanded to include uncertainty over the value of β_t, allowing for ambiguity in how the human interprets the robot's action. In the same vein, additional modes of interaction from both the robot and human can be explored.

The object delivery domain can also be abstracted to a domain where the goal is not to determine which object the human wants, but the human's intent in general. This system of tracking the human's belief about the robot's communicated intent can be widely applied to other human-robot interaction tasks.

VI. Acknowledgements

References

[1] Richard Bellman. A markovian decision process. *Indiana Univ. Math. J.*, 6:679–684, 1957. ISSN 0022-2518.

[2] Dan Bohus and Eric Horvitz. Managing human-robot engagement with forecasts and... um... hesitations. In *Proceedings of the 16th International Conference on Multimodal Interaction*, ICMI '14, pages 2–9, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2885-2. doi: 10.1145/2663204.2663241. URL http://doi.acm.org/10.1145/2663204.2663241.

[3] Joyce Y Chai, Lanbo She, Rui Fang, Spencer Ottarson, Cody Littley, Changsong Liu, and Kenneth Hanson. Collaborative effort towards common ground in situated human-robot dialogue. In *Proceedings of the 2014 ACM/IEEE international conference on Human-robot interaction*, pages 33–40. ACM, 2014.

[4] Joyce Yue Chai, Zahar Prasov, and Shaolin Qu. Cognitive principles in robust multimodal interpretation. *CoRR*, abs/1109.6361, 2011. URL http://arxiv.org/abs/1109.6361.

[5] Herbert H Clark and Meredyth A Krych. Speaking while monitoring addressees for understanding. *Journal of Memory and Language*, 50(1):62–81, 2004.

[6] Mary Ellen Foster, Andre Gaschler, Manuel Giuliani, Amy Isard, Maria Pateraki, and Ronald P. A. Petrick. Two people walk into a bar: dynamic multi-party social interaction with a robot agent. In *International Conference on Multimodal Interaction, ICMI '12, Santa Monica, CA, USA, October 22-26, 2012*, pages 3–10, 2012. doi: 10.1145/2388676.2388680. URL http://doi.acm.org/10.1145/2388676.2388680.

[7] Piotr J Gmytrasiewicz and Prashant Doshi. A framework for sequential planning in multi-agent settings. *Journal of Artificial Intelligence Research*, pages 49–79, 2005.

[8] Sergio Guadarrama, Erik Rodner, Kate Saenko, Ning Zhang, Ryan Farrell, Jeff Donahue, and Trevor Darrell. Open-vocabulary object retrieval. In *Robotics: Science and Systems*, 2014.

[9] Daniel Hewlett, Thomas J Walsh, and Paul Cohen. Teaching and executing verb phrases. In *Development and Learning (ICDL), 2011 IEEE International Conference on*, volume 2, pages 1–6. IEEE, 2011.

[10] Leslie P. Kaelbling, Michael L. Littman, and Anthony R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101(1-2), 1998.

[11] Michael Kearns, Yishay Mansour, and Andrew Y Ng. A sparse sampling algorithm for near-optimal planning in large markov decision processes. *Machine Learning*, 49(2-3):193–208, 2002.

[12] G.M. Kruijff, M. Brenner, and N. Hawes. Continual planning for cross-modal situated clarification in human-robot interaction. In *Robot and Human Interactive Communication, 2008. RO-MAN 2008. The 17th IEEE International Symposium on*, pages 592–597, Aug 2008. doi: 10.1109/ROMAN.2008.4600731.

[13] John Laird. *The Soar cognitive architecture*. MIT Press, 2012.

[14] Matthew MacMahon, Brian Stankiewicz, and Benjamin Kuipers. Walk the talk: connecting language, knowledge, and action in route instructions. In *Proceedings of the 21st National Conference on Artificial Intelligence*, pages 1475–1482. AAAI Press, 2006.

[15] Matthew Marge, Aaron Powers, Jonathan Brookshire, Trevor Jay, Odest C Jenkins, and Christopher Geyer. Comparing heads-up, hands-free operation of ground robots to teleoperation. *Robotics: Science and Systems VII*, 2011.

[16] C. Matuszek, E. Herbst, L. Zettlemoyer, and D. Fox. Learning to parse natural language commands to a robot control system. In *Proc. of the 13th Intl Symposium on Experimental Robotics (ISER)*, 2012.

[17] DK Misra, Jaeyong Sung, Kevin Lee, Ashutosh Saxena, Jaeyong Sung, Bart Selman, Ashutosh Saxena, Jaeyong Sung, Colin Ponce, Bart Selman, et al. Tell me dave: Context-sensitive grounding of natural language to mobile manipulation instructions. In *Robotics: Science and Systems, RSS*, 2014.

[18] Sylvie C. W. Ong, Shao Wei Png, David Hsu, and Wee Sun Lee. Planning under uncertainty for robotic tasks with mixed observability. *Int. J. Rob. Res.*, 29 (8):1053–1068, July 2010. ISSN 0278-3649. doi: 10.1177/0278364910369861. URL http://dx.doi.org/10.1177/0278364910369861.

[19] Paul W Schermerhorn, James F Kramer, Christopher Middendorff, and Matthias Scheutz. Diarc: A testbed for natural human-robot interaction. In *AAAI*, pages 1972–1973, 2006.

[20] Stefanie Tellex, Thomas Kollar, Stephen Dickerson, Matthew R. Walter, Ashish Banerjee, Seth Teller, and Nicholas Roy. Understanding natural language commands for robotic navigation and mobile manipulation. In *Proceedings of the AAAI*, 2011.

[21] Stefanie Tellex, Pratiksha Thaker, Robin Deits, Thomas Kollar, and Nicholas Roy. Toward information theoretic human-robot dialog. 2012.

[22] Stefan Waldherr, Roseli Romero, and Sebastian Thrun. A gesture based interface for human-robot interaction. *Autonomous Robots*, 9(2):151–173, 2000.

[23] David Whitney, Miles Eldon, John Oberlin, and Stefanie Tellex. Interpreting multimodal referring expressions in real time. In *International Conference on Robotics and Automation*, 2016.

[24] Jason Williams, Antoine Raux, Deepak Ramachandran, and Alan Black. The dialog state tracking challenge. In *Proceedings of the SIGDIAL 2013 Conference*, pages 404–413, 2013.

[25] Emily Wu, Yuxin Han, David Whitney, John Oberlin, James MacGlashan, and Stefanie Tellex. Robotic social feedback for object specification. *AAAI: Artificial Intelligence for Human-Robot Interaction*, 2015.

[26] Steve Young, M Gašić, Blaise Thomson, and Jason D Williams. POMDP-based statistical spoken dialog systems: A review. 2013.