Towards Learning User Preferences for Remote Robot Navigation

Cory J. Hayes*, Matthew Marge*, Ethan Stump*, Claire Bonial*, Clare Voss*, Susan G. Hill†
*U.S. Army Research Laboratory, Adelphi, MD, USA
†Aberdeen Proving Ground, MD, USA

Abstract—**We are interested in the design of autonomous robot behaviors that learn the preferences of users over continued interactions so that they may efficiently execute navigation commands given by the user. In this paper, we discuss our planned work to learn a generalized model for robot navigation behaviors using inverse reinforcement learning techniques, and subsequently modify this model per individual user through reinforcement learning. Our approach is motivated by the data collected in an ongoing series of experiments where naive participants provide navigation instructions to a remote four-wheeled robot through natural language dialogue. Participants are tasked with uncovering specific information about an unfamiliar environment via the robot and its sensors, requiring fine-grained robot movement that meets the intention of a given command in order to accomplish the task in a timely manner; this sensitivity often leads to clarification commands to augment the position of the robot in order to achieve the desired user intention. By taking into account user navigation preferences learned from interaction histories, we seek to minimize the number of correction or clarification commands needed for the robot to perform the intended behaviors.**

I. INTRODUCTION

A critical component of teamwork is the ability of group members to understand and predict the intention behind the actions of other members to ensure that the team works as one cohesive unit while addressing multiple subtasks simultaneously. Each team member is a unique individual with different characteristics, perspectives, and capabilities. Together they must work efficiently to achieve an overall goal since, depending on the specific task, it may not be possible for any one member to have full knowledge of what specifically needs to be done to accomplish the goal and thorough awareness of environmental changes [4, 8].

Advances in artificial intelligence have created an opportunity for effective teaming between humans and robots. Reliable robot teammates could enable increased situational awareness and reduce cognitive burden on their human counterparts. Robots can acquire, process, and transmit precise low-level data at a rate that far exceeds the capabilities of their human operators who must process and make decisions based on this information. In the cases where one operator supervises multiple robots, this deluge of data is amplified to an even greater degree. Mental overload has been shown to negatively affect pilots in simulations [10] as well as remote drone operators [17].

A potential solution is to effectively distribute the burden of processing information so that human operators are not overwhelmed. This would be possible by designing robots that can behave in intelligent ways using contextual information and leveraging the knowledge gained from previous interactions with their human operators. For cognitively intensive tasks that require a team, one or more human operators may be faced with a great deal of responsibility, needing to make quick decisions as the environment changes constantly; adding unintelligent robots, ones that need to be instructed at each step, only increases this burden. However, the opposite end of the spectrum where robots fully act on their own as “lone wolves” and do not communicate with their operator is not an ideal solution, since their actions cannot be predicted by human teammates and in turn may negatively impact a person’s trust towards the robot [6, 9].

Wang and Lewis [19] discovered, using virtual simulations, that balancing the workload between a human operator and autonomous robots resulted in a more effective team compared to using either a fully autonomous robot team or a teleoperated robot team. The need for active robot team members that can appropriately shift between directly following instructions and making effective decisions on their own motivates the design of robots that can infer human intent. These human-robot teams would draw parallels with existing human teams that can effectively act as a cohesive whole without requiring explicit commands for every single action, while also not performing any individual actions that could jeopardize achieving the desired goal.

In order to acquire information about an environment in the quickest and most efficient manner, an intelligent robot teammate must be able to operate both within the proximity of their human counterparts as well as when they are remotely located. A robot located in an area separate from a human operator must still operate within the expectations of their user in order to relay relevant information about an environment. For example, if a robot is instructed to survey an interior area and send images, there are many possible ways for the robot to accomplish the task, but there may be preferences by their human operators for how exactly the task should be carried out. One user who only requires minimal information to make quick decisions may want the robot to navigate to the entrance of a room, take a picture that shows most of the room, and then immediately move to the entrance of a new unexplored room (or a different standby location) before waiting for the next instruction. Another user may have the tendency to want as much information as possible about the robot’s surroundings and prefer the robot to behave in a manner similar to the above for small rooms, but for larger rooms that cannot be fully viewed from an entrance, want the robot to move into the room and take images at specific intervals to fully capture the room through images. These are two possible variations to execute an instruction such as “Take a picture of the room”, among other potential variations for this one instruction, which itself is just one of a potential large number of instructions needed to complete complex tasks. Ideally, an intelligent robot teammate would be able to learn these user navigation preferences over continued interactions with the user, instead of needing these nuanced preferences to be explicitly known and programmed into its autonomous behavior in advance or explicitly specified by the user throughout the interaction.

In this paper, we present a plan towards the research goal of exploring the feasibility of capturing user preferences for robot navigation.Using inverse reinforcement learning, we plan to train a general model for robot navigation from leveraging our experience in an ongoing series of natural language studies where participants command a robot to explore an unfamiliar environment to gather specific information. Using this general model of navigation, we then plan to modify it on an individual user basis using reinforcement learning where both measurements from the environment and user feedback as the robot executes commands serve as rewards to create personalized navigation policies for the robot.

II. Background

A. Reinforcement Learning

Reinforcement learning enables an agent to learn how to achieve a certain goal by having it carry out actions and receive feedback from the environment that informs it on how valuable the action was towards achieving said goal [18]. Therefore, reinforcement learning is an ideal technique to use for problems where one cannot explicitly provide the specific actions and sequence needed to accomplish a goal for all possible situations, but can evaluate if the actions taken bring the agent closer to the desired behavior (e.g. driving a car, walking, etc.). This applies to our problem of capturing user preferences for robot navigation because it is not possible for participants to list all of the specific actions that a robot should take for any given environment in advance, especially for environments that are unknown to them. However, an robot that can incorporate feedback from both the environment and participants as it takes actions should be able to learn to act intelligently and in a way that best suits the user.

Formally, the reinforcement learning problem is defined as a Markov Decision Process with *states* s_1,...,s_n, *actions* a_1,...,a_m, *transition function* t(s,a,s’) which is the probability that taking action *a* in state *s* results in state *s’* and *reward* r(s,a) which is the reward for taking action *a* in state *s*. The agent performs actions in these states and receives rewards for doing so, and the overall goal is for the agent to choose actions that will lead to maximum expected reward until task completion (reaching an end state); this deliberate choice of actions is called a *policy*. Another common parameter included is the discount factor γ that ranges from 0 to 1.

$$\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \qquad 0 \leq \gamma \leq 1$$

The reward function in the context of our problem will be a combination of measurements from the environment and user feedback from the robot executing different actions, though the specifics of this are to be determined as we make progress towards training and then modifying a generalized navigation model based on our ongoing experimental setup.

B. Inverse Reinforcement Learning

We plan to use inverse reinforcement learning [1] techniques to create the general navigation model before modifying parameters to create personalized navigation behaviors per user. In inverse reinforcement learning, the agent is not provided with rewards from the environment. Instead, the agent is provided demonstrations $\{s_0,a_0\}$,$\{s_1,a_1\}$,...,$\{s_m,a_m\}$ performed by an expert with the assumption that the teacher is acting optimally; therefore this state-action sequence defines a trace (since it not an exhaustive demonstration) of the optimal policy π^*. The agent’s goal is to estimate the reward function $R(s, a)$ that underlies the optimal policy as demonstrated by the teacher. Formally, we want to find R^* such that

$$E\left[\sum_{t=0}^{\infty} \gamma^t R^*(s_t) | \pi^*\right] \geq E\left[\sum_{t=0}^{\infty} \gamma^t R^*(s_t) | \pi\right] \qquad \forall\, \pi$$

The left side of the above equation is the expected discounted reward over time given the optimal policy π^* provided by the expert, and similarly the right side of the equation is the expected discounted reward over time for all possible policies. We will use inverse reinforcement learning to automatically parameterize the teleoperated demonstrations of multiple participants, and from this create a generalized model of navigation behaviors that reflect the range of acceptable sub-actions needed to perform specific types of navigation instructions.

C. User Uniqueness and Human-Robot Interaction

Human-Robot Interaction (HRI) is a multi-disciplinary field that combines aspects of robotics, social sciences, human-computer interaction, machine learning, and artificial intelligence amongst others. The main focus of HRI is to understand and actively address the dynamics that affect interactions between robots and humans. User-centered design has been shown to be a critical part of HRI that addresses the needs and preferences of human users since each individual person has unique characteristics that can affect how they interact with technology [15]. Existing literature has discussed the importance and feasibility of addressing user uniqueness [2, 7, 5]. Our proposed methodology seeks to directly address individual preferences for how a robot should perform navigation tasks by using reinforcement learning to eventually lead to fluid and efficient interactions.

III. Human-Robot Dialogue Experiments

We are currently conducting a series of experiments that collect natural language dialogue between humans and robots, referred to as "BotLanguage" [3, 11, 12], and approach the goal of capturing user preferences by building upon its collected data and methodology. The overall goal of BotLanguage is to create an autonomous dialogue system that can convert unconstrained and potentially ambiguous natural language into precise navigation instructions for a robot. The focus on unconstrained language follows the motivation to minimize burden and allow anyone, regardless of technical expertise, to utilize a robotic teammate to its fullest potential for exploration tasks. In BotLanguage, naïve participants instruct a robot to navigate through a house-like environment and are tasked with finding specific objects within a certain amount of time. The participants are not given the layout of the environment prior to interacting with the robot, and the robot provides information about the environment to the participant through real-time occupancy grid mapping and image snapshots that they can request; participants are not given a live RGB video feed in order to mimic realistic degraded environments with limited bandwidth.

BotLanguage currently uses the Wizard-of-Oz [16] approach where two researchers substitute for the autonomous dialogue and navigation system under development. The participant communicates verbally over a microphone with one researcher who serves as the Dialogue Manager (DM-wizard). The DM-wizard provides text feedback to the participant through one computer interface and translates the verbal commands to explicit and precise textual navigation instructions to the second researcher, who acts as the Robot Navigator (RN-wizard) via another computer interface. Together these two "wizards" mimic the desired capabilities of our autonomous dialogue system.

As we have discovered over the course of the BotLanguage studies, natural language commands can elicit many possible navigation trajectories for a robot, even with two trained human operators working together and using previously agreed upon guidelines for navigation. For example, when instructed to "Move to [object]", the agreed guideline for the two operators is that the robot's position after executing this command should be at a distance where the entire object is visible and the object is situated in the center of the robot's field of vision. In the ideal conversational exchange, a participant would primarily use this kind of high-level language when communicating with a robot. However, when the teleoperated robot does not move in a manner exactly as anticipated by a participant, we have observed language shifting to more precise and low-level commands such as "Turn X degrees left and move forward Y feet". Such unnatural language is type of communication that we seek to minimize for users interacting with an autonomous agent.

Our initial effort into understanding participant preferences manifested in the work described in Moochandani et al. [14] where we analyzed the annotated dialogue corpus and discovered eight main instruction classes that provided the basis of the navigation commands given across all participants. These instruction classes were 1) approach an entrance, 2) enter a room, 3) approach an object, 4) navigate unexplored areas, 5) handle object obstruction, 6) follow a direction-only command, 7) follow a direction and orientation command, and 8) fulfill an impractical task; see Table I for example commands of each class.

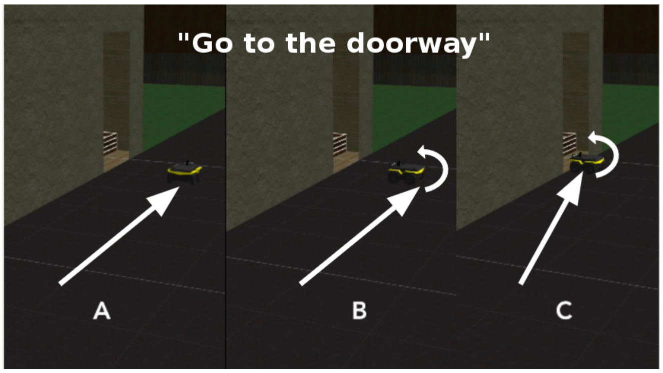

Fig. 1. Volunteers in our prior study viewed a virtual robot executing navigation tasks, varied (A-C) by parameters of distance and orientation, in response to natural language instructions and were asked to rank the behaviors. Arrows depict the robot's path of motion and rotation.

Along with uncovering the eight instruction classes from our dialogue corpus, we also investigated how to describe robot motion as it carried out these instructions, and from the robot movement literature we pinpointed five parameters. These parameters were 1) distance traveled, 2) orientation, 3) smoothness, 4) speed, and 5) self-safety, the last one describing how the robot moves with a sense of self-preservation (e.g. not navigating too close to objects/walls). Using the eight instruction classes, we operated and recorded a virtual robot performing example instructions for each class with variations in the distance traveled and final orientation after executing an instruction; see Figure 1 for an example of variations for the instruction "Go to the doorway".

We then asked new participants to observe the recordings of the robot performing these instructions, select their preferred behavior, and rate the quality of the robot movement. As noted in [14], one limitation of this study was that since it was video-based, we could only provide participants with a small subset of parameter variations to have appropriate lengths for sessions. A solution to this problem is to allow participants to give real-time feedback for robot movement so that it may more accurately capture user preferences for navigation, which motivates the proposed line of research described in this paper.

We are focusing on addressing one subproblem for the creation of intelligent robot teammates that can accomplish a wide variety of tasks by first building upon the knowledge gained from BotLanguage for navigation tasks. A robot that would be an ideal member in a hybrid team must be able to plan its actions in an intelligent manner when given a command.

TABLE I
INSTRUCTION TYPES

Instruction Type	Example Instruction
Approach an entrance	“Go to the doorway”
Enter an unexplored room	“Go into the room in front of you”
Approach an object’s personal space	“Move forward until you reach the crate”
Navigate unexplored areas	“Go to the end of the hallway”
Handle object obstruction	“Move around the cone”
Follow direction-only instruction	“Go to the left”
Follow direction & orientation instruction	“Head three feet north”
Fulfill an impractical task	“Move forward eight feet”[1]

[1] This instruction is impractical due to the robot’s proximity to a wall.

IV. PROPOSED METHODOLOGY

In order to explore the problem of personalized navigation behaviors on an autonomous robot, we propose the following experimental methodology that segments the problem into two main portions that use reinforcement learning and inverse reinforcement learning.

A. Experiment 1: Establishing a General Model of Robot Navigation

The first step in our proposed process for autonomous robot navigation that takes into account user preferences is to model general behaviors for specific navigation instruction classes. In order to allow a model to be modified for individual users, we must first establish a baseline for navigation behaviors. This process follows the methodology used in our aforementioned work[14] where the dialogue corpus was analyzed and summarized into eight instruction types, with variations of each being manually determined and recorded in video clips. Rather than continuing to create these variations ourselves, which introduces navigation biases due to our familiarity with the overall experiment design and data analysis over the multiple phases, as well as imposing a limitation for the number of variations that can be covered, we plan to use reinforcement learning to train a general model that can reproduce navigation behaviors for a given instruction type. The training data for this process would be navigation trajectories provided by participants who directly control a robot, and inverse reinforcement learning allows for these demonstrations to be analyzed and replicated by an autonomous system.

The proposed experiment for this step is to create a virtual environment where participants control a robot using a joypad, and we motivate the navigation demonstrations by providing participants with mock missions to accomplish within a certain time limit, similar to the objectives given to participants in BotLanguage. Confederates will provide the participants with instructions to carry out, and the missions given to the participants will require various combinations of the instruction types uncovered in the dialogue corpus.

1) Experimental Framework: We have used the Robot Operating System (ROS) and the Gazebo simulator to create virtual urban and home-like environments for robot navigation in previous experimental setups. Our familiarity with Gazebo in previous experimental setups makes it the primary candidate for creating the virtual environment for participants as we have already created multiple indoor/outdoor environments utilizing its capabilities. Two other potential virtual simulators are the Unity and Unreal game engines, which can now integrate with ROS in a manner similar to Gazebo through rosbridge, a protocol that allows for interprocess communication between ROS and non-ROS programs. Both Unity and Unreal allow for the creation of highly-detailed and dynamic environments which may prove to be more advantageous than Gazebo since these environments would more closely mimic dynamic real-world environments in which autonomous robots must operate. All three of these simulators allow for virtual environments to be sensed by a virtual robot, which is critical because in order to analyze and train robot behavior, we must be able to pick up features from the environment and continuously monitor the state of both the robot and the environment.

2) Navigation Features: We plan to use Bayesian non-parametric techniques to reproduce and generalize the navigation trajectories provided by the participants. Non-parametric techniques use models that automatically expand to accommodate data as it is observed and will be necessary because it is not possible to enumerate all of the possible states and actions within each instruction variation in advance. Because a long demonstration trajectory is likely to only be optimal for a very complicated reward function, we break the demonstration into smaller pieces for which we can learn simpler reward functions and then compose them in sequence. The Bayesian Non-Parametric Inverse Reinforcement Learning algorithm [13] gives us an approach for performing this by using Markov Chain Monte Carlo sampling techniques to approximate the trajectory partition and associated reward functions that best explain the full demonstration in a maximum likelihood sense.

3) Evaluation: Once the navigation behaviors are trained, we plan to evaluate their accuracy in a multi-stage process. First, we will perform a cross-validation phase where we replicate the segmented demonstrations provided by participants who directly controlled the robot in the same environment. The autonomous individual behaviors will be performed in the order determined by the training demonstrations; correctness of behavior will be done by analyzing the end positions of the robots using metrics of distance and orientation relative to goal position; behaviors cannot be replicated exactly per participant demonstration since the trained model will be a collective "average" of participant behaviors. The immediate next phase of evaluation would be to analyze the robot behaviors in an environment that differs from the training environment using the same method.

The third phase in the evaluation process would be to replicate the BotLanguage setup where participants once again provide navigation commands to a robot. The Wizard-of-Oz approach will remain the same with two wizards, one acting as DM-wizard and the other as RN-wizard; however, instead of the robot navigator manually operating the robot when given a command, the RN-wizard will control the robot through an interface that allows him/her to select the automated behavior depending on the instruction type. The interface will also provide the optional ability for the RN-wizard to take control of the robot and operate it as before in cases of sensor failure or behavioral failure. After experiment runs, we will follow an approach similar to the one used in [14] where participants answer qualitative survey questions about the robot's movement behavior.

B. Experiment 2: Customizing the General Model of Navigation Per User

The second portion of this experimental process is to build upon the generalized model trained from the first portion and allow for customizable modifications per participant using traditional reinforcement learning methods. The portion will once again be similar to the BotLanguage setup with the DM-wizard translating and sending commands to the Robot Navigator, except in this stage the RN-wizard will be entirely automated, where any movements performed by the robot will be done automatically through the software. This experiment will use longitudinal studies where participants will have interactions with the autonomous robot across multiple trials, with a currently unspecified period of time between each. Each trial will differ from previous trials due to either the specific task that is given to the user to accomplish, or the virtual environment in which the robot will navigate. The main objective of this experiment is to explore the feasibility of incorporating real-time user feedback to train a reinforcement learning policy, with initial attempts focusing on offline learning where changes to the navigation model will not be reflected within the same trial but updated between interactions.

1) Experimental Framework and Navigation Features: Similar to the experiment described in Section IV-A, this experiment will leverage the capabilities of ROS, Gazebo (or Unity/Unreal), and rviz to create multiple virtual environments.

The general navigation behavior model would reflect the range of acceptable actions for a given instruction type that were captured from teleoperated demonstrations in the first stage of the experiment, with its own underlying reward function and optimal policy that build from that reward function. In this second stage of the experiment, speech will be used as part of the new reward function to augment the model's existing policy, which will result in the modification of robot behaviors on an individual user basis. We will focus on two types of user speech, correction/clarification and acceptance, which define negative and positive reward respectfully.

Clarifications or corrections are reflected by any speech that re-issues or provides additional detail in regards to a previously given instruction. An example of a clarification/correction would be a user saying something such as "Robot turn to face the doorway", observing the robot perform the action, and then following up the previous command with "Robot turn 15 degrees to the right", which is a minor modification to the previous command to achieve the desired behavior. A more apparent example of a clarification/correction, which we have observed occasionally in previous trial and real runs in BotLanguage sessions, would be a user explicitly saying something along the lines of "No, I did want you to do [action], I wanted you to do [other action]".

On the other hand, a user continuing the interaction by giving a different command signifies an acceptance of the performed robot action and will be reflected as a positive reward. An example of an acceptance would be "Robot turn to face the doorway", observing the robot's action, and saying "Take a picture", reflecting that the user believed that the robot's action was acceptable and it should now perform a different action. Both clarifications/corrections and acceptances will require manual annotation of the user's dialogue to pinpoint since they will be largely dependent on the dialogue history of the interaction, and this is the primary reason why we plan to use an offline learning process where the navigation model is updated between user trials.

2) Evaluation: Evaluation of the user-specific models will be both qualitative and quantitative. Quantitative metrics will focus on whether the robot was able to accomplish the given objectives from user instructions, the time required to complete the task, and the number of verbal corrections/clarifications a participant gives to the robot in order to perform correct behaviors. Qualitative metrics will once again be based on post-interaction surveys where participants answer questions about the robot's movement and overall satisfaction with how the robot performed.

The expectation is that the success rate of missions will increase in the ideal outcome or, in the worst case scenario, be non-decreasing and plateau after a certain period of time with a performance that is satisfactory to individual participants. Additionally as the navigation model learns more about user preferences across sessions, the number of corrections/clarifications given by a participant should decrease per session along the time required to complete individual trials.

C. Challenges

There are a few immediate challenges to address during the design and implementation of this experiment in addition to the expected challenges for any machine learning training and testing process.

One challenge is to ensure the validity and robustness of features used to analyze, train on, and replicate participant demonstrations. We plan on starting off with the features mentioned in our previous work [14], which are *distance traveled*, *robot orientation*, *travel velocity*, *smoothness of travel*, and *consideration of self-preservation* which is a multifaceted parameter that can be broken down in various ways but ultimately compares the position/orientation of the robot relative to specific objects or locations of interest. Additional features related to the analysis of robot movement may need to be incorporate or some of the above parameters may need to be modified/removed, but that is a challenge to address once the virtual environment is fully developed and the piloting phase of the experiment has begun. Also inherent to the choice of features that parameterize participant demonstrations is the discretization of the environment which affects how accurately a demonstration can be replicated and the robustness of these demonstrations in different areas of the environment (or new environments entirely).

A problem common to all machine learning approaches is that as the number of features increase, the more training data that is required, which translates to more "expert" demonstrations provided by participants. In the first experiment, each participant will be able to provide multiple demonstrations per instruction class type, however we will still require more participants than the ones who volunteered in prior Bot-Language experiments which were limited to a couple dozen per phase. As the feature set from the general model training becomes more apparent, we will be able to perform statistical power analysis to estimate the number of participants and teleoperation demonstrations needed.

Perhaps the main challenge for the second experiment, which uses reinforcement learning, is the incorporation of real-time user feedback as part of the reward function for the autonomous robot as it performs actions. Agent rewards will be a function of both rewards gathered from the environment and direct user input, and determining how these two can be combined to calculate reward will take some considerable analysis. The user input we plan to use as negative rewards are corrections and clarifications, which are reflected through re-issuing the immediately preceding command in a different method and/or making a direct comment on the correctness of the robot's actions (e.g. "No I did not want you to do that..."). These user responses will have to be manually annotated by coders and may be reflected by a wide range of utterances. Additionally, the assumption that the absence of a user providing feedback for a previously given command and instead moving on to a different instruction signifies a correct robot action may not necessarily be true. Instead this lack of a correction could be a result of impatience with the robot among other potential reasons; this may require careful consideration once we reach this stage of experimentation.

V. Summary

In this paper, we present a plan and methodology to investigate learning user preferences for robot navigation using rewards that are a combination of environment features and natural language from a user during interactions. Our overall goal is to design algorithms for robots that can be tailored to a user's expectations for robot navigation. More specifically, we will first collect data to train a general model of navigation that, given specific language commands, will execute an instruction in a manner that accomplishes the task intention. This model, trained with inverse reinforcement learning on data collected from human subjects, will provide the range of acceptable paths that users actively demonstrate through manual teleoperation of a robot in a simulated environment, and which would ultimately be implemented on an autonomous virtual robot. Given that the obtained set of demonstrations may not thoroughly capture the range of acceptable behaviors, the inverse reinforcement learning process will involve estimating the parameters that define the collective users' trace behaviors.

In order to customize the general navigation model per individual user, we will then learn the navigation features that reflect the user's preferences for robot navigation behavior beginning with the five types of features that we have used in previous research to define robot movement. These features are *distance traveled*, *orientation*, *smoothness*, *speed*, and *self-safety*. The five features may cover only a subset of the unnamed navigation features captured by the general model through inverse reinforcement learning, and a critical step in this second phase of experimentation is evaluating if they are sufficient for capturing user preferences or if the feature set will need to be amended.

Users will command the robot to perform instructions to accomplish a task and provide feedback regarding the robot's movements through command corrections, clarifications, and acceptance of the robot's actions signified by continuing the interaction with a new instruction. Using quantitative analysis of these corrections and acceptances, in addition to a qualitative analysis of post-experiment surveys where users rate the robot's movements, we will then update each user's model and iterate it over multiple trials where users once again evaluate the robot's behavior in each trial with new tasks and environments. The terminating point on these iterations will be when the robot no longer produces errors, or the more likely scenario where or reaches a plateau with some small number of errors. We anticipate that the proposed methodology of the (1) collection of user preferences in real-time human-robot interactions, (2) training of a general model of user preferences with inverse reinforcement learning, and (3) user-tailoring of navigation model updates via reinforcement learning can be applied to many applications within human-robot communication.

Acknowledgments

This research was sponsored by the Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-16-2-0117. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

References

[1] B.D. Argall, S. Chernova, M. Veloso, and B. Browning. A survey of robot learning from demonstration. *Robotics and autonomous systems*, 57(5):469–483, 2009.

[2] C. Bartneck and J. Forlizzi. A design-centered framework for social human-robot interaction. *13th IEEE International Workshop on Robot and Human Interactive Communication*, pages 591–594, 2004.

[3] C. Bonial, M. Marge, A. Foots, F. Gervits, C.J. Hayes, C. Henry, S.G. Hill, A. Leuski, S.M. Lukin, P. Moolchandani, K.A. Pollard, D. Traum, and C.R. Voss. Laying down the yellow brick road: Development of a wizard-of-oz interface for collecting human-robot dialogue. *AAAI Fall Symposium Series: Natural Communication for Human-Robot Collaboration*, 2017.

[4] N.J. Cooke, J.C. Gorman, C.W. Myers, and J.L. Duran. Interactive team cognition. *Cognitive Sciences*, 37:255–285, 2013.

[5] K. Dautenhahn. Robots we like to live with?! - a developmental perspective on a personalized, life-long robot companion. *13th IEEE International Workshop on Robot and Human Interactive Communication*, pages 17–22, 2004.

[6] M. Desai, P. Kaniarasu, M. Medvedev, A. Steinfeld, and H. Yanco. Impact of robot failures and feedback on real-time trust. *8th ACM/IEEE International Conference on Human-Robot Interaction*, pages 251–258, 2013.

[7] J.L. Drury, J. Scholtz, and H.A. Yanco. Applying cscw and hci techniques to human-robot interaction. *Proceedings of the CHI 2004 Workshop on Shaping Human-Robot Interaction*, pages 13–16, 2004.

[8] S.M. Fiore and T.J. Wiltshire. Technology as teammate: Examining the role of external cognition in support of team cognitive processes. *Frontiers in Psychology*, 7, 2016.

[9] P. Hancock, D. Billings, K. Schaefer, J. Chen, E. de Visser, and R. Parasuraman. A meta-analysis of factors affecting trust in human-robot interaction. *Human Factors: The Journal of Human Factors and Ergonomics Society*, pages 517–527, 2011.

[10] K. Huttunen, H. Keranen, E. Vayrynen, R. Paakkonen, and T. Leino. Effect of cognitive load on speech prosidy in aviation: Evidence from military simulator flights. *Applied Ergonomics*, pages 348–357, 2011.

[11] S.M. Lukin, F. Gervits, C.J. Hayes, A. Leuski, P. Moolchandani, J.G. Rogers, C.S. Amaro, M. Marge, C.R. Voss, and D. Traum. Scoutbot: A dialogue system for collaborative navigation. *56th Annual Meeting of the Association for Computational Linguistics - Demonstrations*, 2018.

[12] M. Marge, C. Bonial, A. Foots, C.J. Hayes, C. Henry, K. Pollard, R. Artstein, C. Voss, and D. Traum. Exploring variation of natural human commands to a robot in a collaborative navigation task. *Proceedings of the First Workshop on Language Grounding for Robotics*, pages 58–66, 2017.

[13] B. Michini, T.J. Walsh, A. Agha-Mohammadi, and J.P. How. Bayesian nonparametric reward learning from demonstration. *IEEE Transactions of Robotics*, 31(2): 369–386, 2015.

[14] P. Moolchandani, C.J. Hayes, and M. Marge. Evaluating robot behavior in response to natural language. *Companion of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, pages 197–198, 2018.

[15] D.A. Norman. *The design of everyday things*. Basic Books, 1988.

[16] L. Riek. Wizard of oz studies in hri: A systematic review and new reporting guidelines. *Journal of Human-Robot Interaction*, 1(1), 2012.

[17] T. Shanker and M. Richtel. New military, data overload can be deadly. 2011. URL http://www.nytimes.com/2011/01/17/technology/17brain.html.

[18] R.S. Sutton and A.G. Barto. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

[19] J. Wang and M. Lewis. Human control for cooperating robot teams. *2nd ACM/IEEE International Conference on Human-Robot Interaction*, pages 9–16, 2007.