Modeling User Expertise for Choosing Levels of Shared Autonomy

Lauren Milliken and Geoffrey A. Hollinger
School of Mechanical, Industrial & Manufacturing Engineering
Oregon State University, Corvallis, OR 97331
Email: {millikel,geoff.hollinger}@oregonstate.edu

***Abstract*—In shared autonomy, the robot and human user both share some level of control in order to achieve a shared goal. Choosing the correct balance of control given to the user and the robot can be a challenging problem, since different users have different preferences and vary in skill levels when teleoperating a robot. We propose using a Partially Observable Markov Decision Process (POMDP) to represent a model of the user's expertise in controlling the robot. The POMDP uses observations from the user's actions and from the environment to update the belief of the user's skill and chooses a level of control between the robot and the user. The level of control given between the user and the robot is encapsulated in macro-action controllers, which are used in place of the POMDP choosing a low-level action at each time step. A small pilot study of three users is done to test the model in a simulated robot driving environment. The results of the pilot study show how the system successfully models the belief in the user's skill level.**

I. Introduction

When using robots in the field, there are many advantages to using shared autonomy, where both the human operator and the robot have some level of control. Direct teleoperation may be tedious or difficult, and assistance from the robot may be able to take away a great deal of burden from the human user. However, different users may have varying needs when it comes to the level of autonomy the robot should be given. A novice user may need a great deal of assistance in performing even basic tasks in order to complete the goal safely. On the other hand, a user with more experience may be able to accomplish these tasks easily without as much assistance from the robot, and may even dislike the lack of control in certain situations. We propose a method of using Partially Observable Markov Decision Processes (POMDPs) to model user expertise and choose the optimal level of shared autonomy. The shared control of the system is encompassed by macro-action controllers. Rather than solving the POMDP with many low-level actions, the macro-action controllers encapsulate the shared autonomy control that maximizes the performance based on the user's expertise.

Previous work using shared autonomy has looked at the arbitration between the user's input and the robot's assistance [3] [4]. Such work has found that even when assistance from the robot decreases the task completion time, some users still preferred feeling in control of the robot. By encompassing the user's actions, both desirable and undesirable, the system can choose behaviors that both optimize reaching the goal and perform actions preferred by the user. There has also been previous work developing robots to adapt to seek greater trust from the user [6]. The user's frustration and fatigue when operating the robot can be decreased if they have greater trust in the robot's capabilities. Markov models have also been used to model human behavior in human-robot teams. [5] models the human's willingness to adapt in order to improve the effectiveness of the teams while retaining human trust in the robot. We hypothesize that by modeling user expertise, the user's trust in the robot and the robot's trust in the human can both be used to find the appropriate level of shared autonomy to accomplish the goal. This paper provides a system that will be used in future work to test this hypothesis.

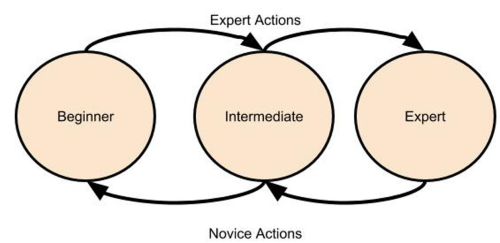

Fig. 1. As the user performs more actions expected from an expert, the likelihood the user is classified as an expert increases. If the user performs more actions expected from a novice, they will be more likely to be classified as a beginner.

II. Algorithm Description

We propose using a POMDP to learn a human's level of expertise and choose the level of autonomy to give the robot based on this level. The POMDP model is a tuple $<S,A,O,T,\Omega,R,b_o,\gamma>$. The set of states, S, encompasses the user's level of expertise and some low-level state of the environment. The set of observations, O, includes the observations from the environment as well as the observations that may indicate a user's level of expertise, such as how many times the user came close to or hit an obstacle, how quickly they completed the goal, or if they are staying at constant speeds or rapidly accelerating. The actions, A, are the macro-actions controllers. T, Ω, R, b_o, and γ represent the conditional transition probabilities between states, the conditional observation probabilities, the reward function, the initial belief, and the discount factor respectively.

The macro-action controllers can be set up for different user levels and environmental states. By selecting macro-action controllers rather than low-level motions, the number of actions the POMDP must be solved for can be greatly reduced [1]. In a human-robot system, some of the burden of modeling the state of the world is taken over by the human rather than relying on the POMDP alone. Our macro-action controllers combine the actions of the user and the robot so that the POMDP requires a fewer number of states and observations to navigate through the environment.

For our initial experiments, we generated the POMDP policy offline using hand-tuned values. A policy, π is computed to maximize the expected total reward. The states of the world represent the user's expertise and the "difficulty" of the map. Easy maps contained fewer, more spread out obstacles while hard maps contained a greater number of obstacles which require finer maneuvering to avoid. The observations in the model represented the perceived difficulty of the map based on the number of obstacles found and the amount the user collided with an obstacle. Four controllers were created that provided different levels of assistance to the user. These controllers are described in more detail below. Table I shows the states, actions, and observations chosen for our initial experiments.

TABLE I
POMDP FOR MODELING USER EXPERTISE

S	Beginner-Easy, Beginner-Hard, Expert-Easy, Expert-Hard
A	Controller1, Controller2, Controller3, Controller4
O	No Obstacles Hit-Easy, Few Obstacles Hit-Easy, Many Obstacles Hit-Easy, No Obstacles Hit-Hard, Few Obstacles Hit-Hard, Many Obstacles Hit-Hard

At each time step, the policy selects the macro-action controller $a \in A$ based on the current belief state b. The controller provides some level of shared autonomy to the user. When the time step completes, the observation $o \in O$ is received based on the user's actions and the actions taken by the controller as well as some observation of the environment. The belief state is updated based on the current belief, the controller used, and the observation given by $b' = \tau(b, a, o)$ and the process continues with the new belief state.

The macro-action controllers were created to help beginners avoid obstacles and to help both beginners and experts navigate through maps with many obstacles. Two controllers were made with "expert" users in mind. One offers no assistance to the user, while the other limits the acceleration to make driving slightly slower, but also easier. The two controller's created with the "beginner" in mind use a potential field method similar to the method presented in [2]. When obstacles are within range of the robot, the controller attempts to slow the robot down and steer it away from the incoming obstacle. θ_R is the repelling angle from potential field of the obstacles. $U = [0, 1]$ is the amount of user influence. As the user presses the control buttons on the keyboard, U increases. Once the button is released, U decreases. θ_U is the angle of rotation from the user's control. $\theta = U\theta_U + (1-U)\theta_R$ is the final angle of rotation for the robot leveraging both the user's and the robot's steering commands. While the controllers designed for the beginner states were able to avoid obstacles, they also took away some of the control from the user and would decrease the speed of the robot. This may be preferred by a beginner who may require the robot to assist them, but could slow down an expert user.

III. Pilot Study and Results

The algorithm was tested using a simple robot driving simulation game developed in Python. The user drove the robot through eight maps, trying to reach a marked goal position as quickly as possible while avoiding obstacles. Once the user reaches the goal, the next map loads and the user returns to the start position. During a trial, the user drives through the eight maps. After the eighth map is completed the user begins the next trial starting back at map 1. A screenshot of one of the maps is shown in Figure 2. For each user, the belief state is set so that the user has an equal probability of being either a beginner or expert. The belief state is updated once the user reaches the goal. After the belief state is updated, the macro-action controller to be used is chosen based on the update belief state. For the first trial, the belief state is updated, but the controller chosen never changes. The user was alerted by a message on the screen when they hit and obstacle and when the controller would change to provide less or more autonomy.

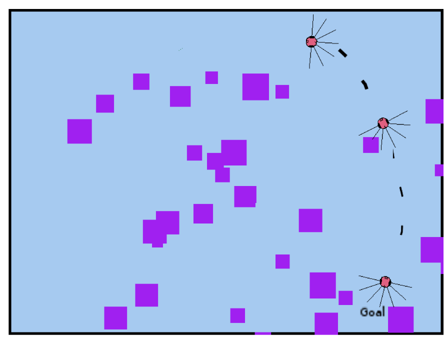

Fig. 2. The robot driving simulator created to test the algorithm. The user and robot's aim is to drive to the Goal position without hitting any of the obstacles.

The pilot study was performed with three users to confirm that our POMDP model could predict a user's level of expertise and chose a macro-action controller that provides the most effective amount of assistance for a user with their skill. For two users, it was the first time running the simulator, while the third user had substantial prior experience with the simulator. Figure 3 shows the number of times the user collided with an obstacle during the trial. Except for the first trial, the number of obstacles users 1 and 2 hit did decrease. Figure 4 shows the probability of the user being a beginner after completing each map. Users 2 and 3 follow similar trends for the belief of their expertise level, while user 1 spends more time as a beginner. Figure 5 shows the average time to complete the map over the four trials. User 3 always took the shortest amount of time to complete the maps while users 1 and 2 had similar times.

There are some interesting differences between the first two users who were using the system for the first time versus user 3 who had prior experience with it. User 3 hit many more obstacles initially, then hit fewer obstacles when the controllers were changed by the POMDP policy, while on the other hand users 1 and 2 showed the opposite. One possible reason for this may be that user 1 and 2 may have been initially more cautious in their control in the first trial since it was the first time seeing the map and driving the robot. Another reason may be the first two users were not used to using the controllers with more assistance, and instead of using the robot's autonomy to help them, responded negatively to the robot's help.

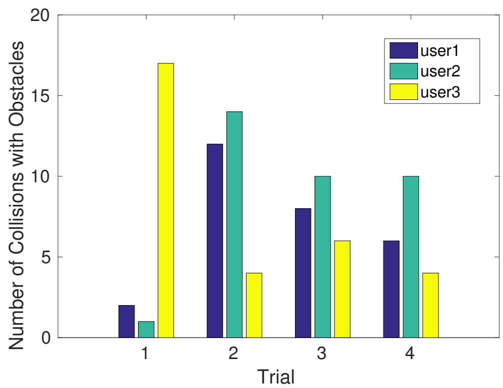

Fig. 3. The number of times each user collided into an obstacle over all the maps in each of the trials.

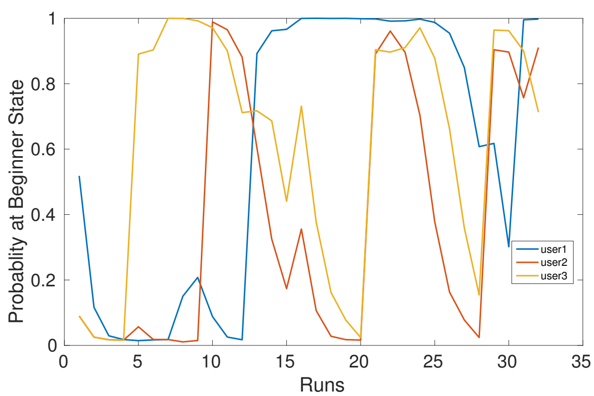

Fig. 4. The probability of the user performing at beginner level based on the belief state of the POMDP.

IV. Conclusion and Future Directions

The results of the pilot trial gave some valuable insight into how well the POMDP modeling worked and how to set up future user studies. Even though user 3 had more experience driving the robot in the simulation, the amount of time they spent as a beginner was similar to the other users. User 3 reached the goal consistently more quickly. This first iteration of modeling expertise using a POMDP only observed the number of obstacles hit in cluttered or uncluttered maps. While

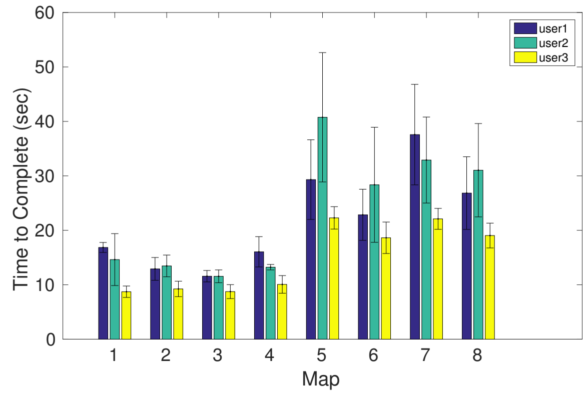

Fig. 5. The average time to complete the map over all four trials from each of the users. The first four maps were "easy" maps with fewer obstacles to avoid, while the last four where "hard" maps. The "hard" maps took longer to complete.

we were able to learn some information on how effective the user was at avoiding obstacles, this is not sufficient to fully encompass what makes a user a beginner or expert.

We are currently working on adding more observations to the POMDP to better model the user's expertise such as the user's time to complete the goal and the stability of the user's control. We also plan to implement the algorithm on a mobile robot in order to test the algorithm's ability to model user expertise in the field. Currently the POMDP parameters are hand-tuned and a policy is generated offline. In the future we would like to incorporate reinforcement learning techniques so that the parameters of the POMDP can be trained instead of hand-tuned. A full-scale user study will also be performed with a greater number of users with varying levels of experience with video games and robotics. With this user study, we will test our hypothesis that a model of user expertise can be used to provide the most effective level of shared autonomy in human-robot systems.

Acknowledgments

This research is funded in part by NSF grant IIS-1317815 and support from PCC Structurals, Inc. and OMI. User studies were performed under Oregon State University IRB 6656.

References

[1] C. Amato, G.D. Konidaris, A. Anders, G. Cruz, J. P. How, and L. P. Kaelbling. Policy search for multi-robot coordination under uncertainty. In *Robotics: Science and Systems Conference*, Rome, Italy, 2015.

[2] J. W. Crandall and M. A. Goodrich. Characterizing efficiency of human robot interaction: A case study of shared-control teleoperation. In *Proc. IEEE/RSJ International Conference Intelligent Robotics and Systems*, volume 2, pages 1290–1295, 2002.

[3] A. D. Dragan and S. S. Srinivasa. A policy-blending formalism for shared control. *The International Journal of Robotics Research*, 32(7):790–805, 2013.

[4] S. Javdani, S. S. Srinivasa, and J. A. Bagnell. Shared Autonomy via Hindsight Optimization. In *Robotics: Science and Systems*, Rome, Italy, 2015.

[5] S. Nikolaidis, A. Kuznetsov, D. Hsu, and S. Srinivasa. Formalizing human-robot mutual adaptation via a bounded memory based model. In *Proc. Eleventh Annual ACM/IEEE International Conference on Human-Robot Interaction*, pages 75–82, 2016.

[6] A. Xu and G. Dudek. Optimo: online probabilistic trust inference model for asymmetric human-robot collaborations. pages 221–228, 2015.