Assistive Teleoperation: A New Domain for Interactive Learning

– Extended Abstract –

Anca D. Dragan and **Siddhartha S. Srinivasa**
The Robotics Institute
Carnegie Mellon University
{adragan,siddh}@cs.cmu.edu

Assistive Teleoperation as Policy Blending

In *assistive* teleoperation, the robot attempts to predict the user's intent and augments his or her input based on this prediction, in order to simplify the task. Our recent work on *policy blending* (Dragan and Srinivasa 2012) formalizes assistance as an arbitration of the user's input and the robot's prediction. At any instant, the robot combines the input, U, and the prediction, P, using a state-dependent arbitration function $\alpha \in [0, 1]$ (Fig.1(bottom)). Policy blending can have a strong corrective effect on the actual input provided by the user, but burdens the robot to *predict* accurately and *arbitrate* appropriately.

Prediction. The first step in policy blending is predicting the user's intent, given the trajectory of user inputs so far and any other cues that can assist prediction, such as gaze, or context. For manipulation tasks, which are often described in terms of grasping goals and possibly constrained trajectories to these goals, intent prediction becomes the union of two problems: identify what goal the user is trying to achieve, and predict the trajectory the robot should use to move to that goal.

Prior work addressed these problems by assuming that the goals and trajectories to them are static, i.e. they are the same during training and testing. This enables a multi-class classifier to identify which trajectory matches the current user's input best (Demiris and Hayes 2002; Fagg et al. 2004; Yu et al. 2005; Li and Okamura 2003; Aarno, Ekvall, and Kragic 2005). However, many real-world problems are in fact described by goals that change: when we clean up the dining room table, we find different objects in different locations today than we did yesterday or the day before.

In this general case of variable goals, the robot must first identify the user's intended goal. Given the trajectory of user inputs so far $\xi_{S\rightarrow U}$ and any other available cues θ, the robot must predict the user's intended goal configuration, G^*, from a set of possible goals $\{G_1, G_2, \ldots, G_N\}$ identified at run-time. That is, the robot must find the goal that maximizes the posterior probability:

$$G^* = \underset{G\in\{G_1,G_2,\ldots,G_N\}}{\arg\max} P(G|\xi_{S\rightarrow U}, \theta) \quad (1)$$

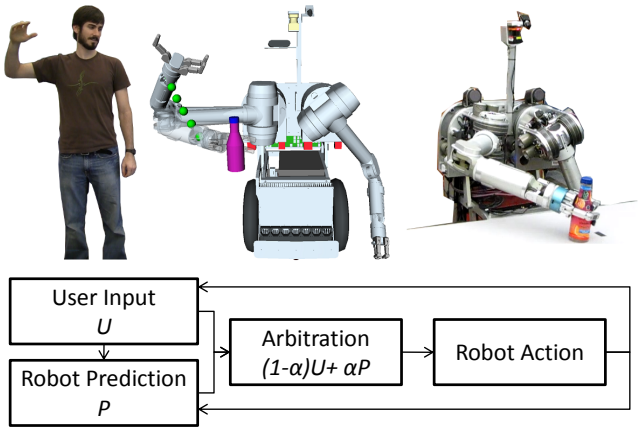

Figure 1: (Top) The user provides an input U. The robot predicts their intent, and assists them in achieving the task. (Botttom) Policy blending arbitrates user input and robot prediction of user intent.

Once it has made a goal prediction, the robot must compute the next action P to take in order to achieve this goal. This next action can come from a policy to G^* mapping each state to a corresponding action, or it can come from a trajectory from the robot's current configuration to G^*.

Arbitration. Given U and P, the robot must decide on what to do next. Despite the diversity of methods proposed for assistance, from the robot completing the grasp when close to the goal (Kofman et al. 2005), to virtual fixtures for following paths (Aarno, Ekvall, and Kragic 2005), to potential fields towards the goal (Aigner and McCarragher 1997), all methods can be seen as arbitrating user input and robot prediction. The arbitration function α can depend on a number of inputs, such as the distance to the goal or to the closest object, or even a binary switch operated by the user. We propose a simple principle: that arbitration must be moderated by how good the prediction is. This leads to a spectrum of arbitration functions. On the one hand, the assistance could be very timid, with α taking small values even when the robot is confident in its prediction. On the other hand, it could be very aggressive: α could take large values even when the

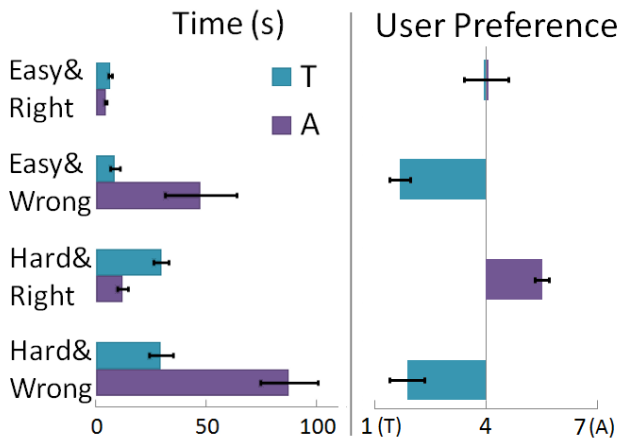

Figure 2: A comparison of timid vs. aggressive assistance in terms of the time users took to complete manipulation tasks with the robot HERB (Srinivasa et al. 2012), as well as the users' preferences for one mode or the other.

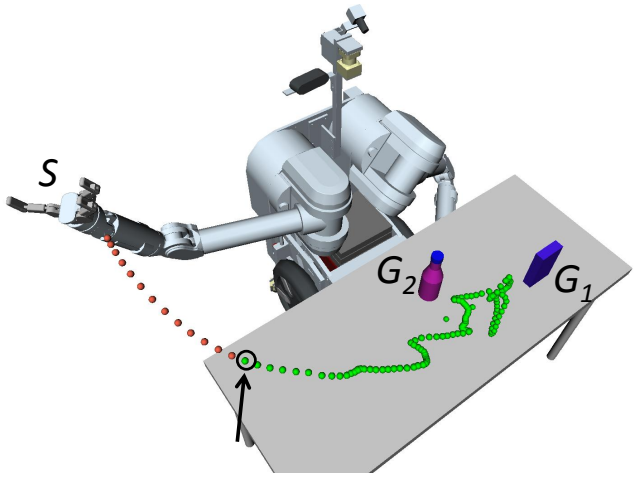

Figure 3: A user's teleoperated trajectory to the box (G_2), with points along it marked in green vs. red corresponding to whether the goal identification using the trajectory up to that time-point was successful or not.

robot does not trust the predicted policy.

Where arbitration should lie on this spectrum remains an open question. Although the dependence on confidence has not been studied before, previous work has analyzed how more autonomous vs. more manual assistance modes affect the performance of assistive teleoperation, in terms of both efficiency and user preferences. The results are surprisingly mixed, with some studies reporting that users favor autonomous assistance due to its improved efficiency (You and Hauser 2011; Marayong, Okamura, and Bettini 2002), while others report that users prefer direct teleoperation (Kim et al. 2011). We found that this could be explained by analyzing the interaction of aggressiveness with other factors, such as prediction correctness or task difficulty: users prefer aggressive assistance on tasks that are very hard to complete with direct teleoperation, they prefer timid assistance when the robot makes the wrong prediction, and their opinions are mixed on easy tasks – some prefer the slight improvement in efficiency that an aggressive mode provides, while others want to remain in control of the robot's actions. Fig.2 visualizes this comparison, and more details can be found in (Dragan and Srinivasa 2012).

Interactive Learning

Prediction comprises of identifying the intended goal and generating a trajectory to it. Both components are characterized by the need to adapt to a particular user, as well as the opportunity to interact with the user to achieve this adaptation. These characteristics make prediction for assistive teleoperation an exciting domain for interactive learning.

Goal Identification

The problem of predicting a trajectory's goal given the trajectory so far was addressed by Ziebart et al. (Ziebart et al. 2009) in the context of predicting a pedestrian's destination. The idea is to assume that the pedestrian is optimizing a cost function, subject to noise. Given this assumption and by adopting the principle of maximum entropy (Ziebart et al. 2008), soft-maximum value iteration can be used to compute the probability of a goal(Ziebart et al. 2009).

Although this method is tractable in two-dimensional spaces, like the ones in which pedestrians move, value iteration becomes intractable for the high-dimensional spaces induced by manipulation tasks. However, we found that if the user's cost function C can be approximated by a quadratic, prediction reduces to

$$G^* = \arg\max_G \frac{e^{C(\xi^*_{S\to G})}}{e^{C(\xi_{S\to U})+C(\xi^*_{U\to G})}} P(G) \qquad (2)$$

This instantaneous prediction method implements an intuitive principle: if the user appears to be taking (even in the optimistic case) a trajectory that is a lot costlier than the optimal one to that goal, the goal is likely not the intended one. We have also found that even a simple cost function, such as the sum of the squared velocities along the trajectory, performs well in practice (an example in Fig.3).

Learning to better identify goals. In our experiments, we noticed that different users adopt different strategies for teleoperating the robot. Some would try to take direct paths to the goal, while others would make progress in one dimension at a time. Some would go above obstacles, while others would go around. The existence of these differences implies that good predictors would have to adjust their model to a specific user: *good goal predictors must learn from the interaction with a user and adapt to that user's behavior.*

Formally, this adaptation is an online update of the cost function C that the user is assumed to be optimizing. Let C be a weighted combination of features, $C = w^T f$. As the user is teleoperating, the robot can compute an online update on w with each new example trajectory to a goal. This update

is the result of a convex optimization problem, as shown in (Ratliff, Bagnell, and Zinkevich 2006).

Given the online learner, the challenge is in deciding how to arbitrate during the learning process. Aggressive arbitration corrupts the training data if the robot makes the wrong prediction, because the trajectory to the goal will be tamed by the user's negative reaction in controlling the robot. On the other hand, timid arbitration is insufficient on hard tasks. Fortunately, the robot does have at its disposal tools that can help in making this decision: a prior on task difficulty, a measure of how good previous predictions have been, and the possibility of disregarding examples by detecting failures in prediction.

A different kind of interactive learning. Aside from the user training the robot online through example trajectories, the robot can also train the user. We hypothesize that given feedback from the robot (through motion and displays of its predictions, as in Fig.4), users can learn to provide more intent-transparent examples. The degree to which a motion is intent-transparent is actually judged by the robot, by how accurate (and with what confidence) its predictions are for that motion. Through the robot's feedback, users can learn what types of inputs lead to better predictions.

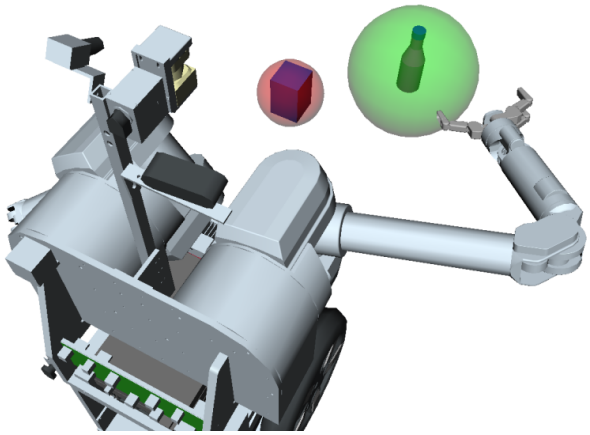

Figure 4: The robot could train the user to provide more intent-expressive examples by giving him feedback on what it is predicting and how confident it is.

Trajectory Generation

At every time-step, the robot makes a goal prediction. This is not enough for assisting the user: the robot must also compute a predicted next action P that the user would like to take towards the goal. One way to so in high-dimensional spaces is to compute a collision-free path to the goal. The default algorithms for solving this problem are randomized planners (e.g. RRT (Kuffner and LaValle 2000) followed by a post-processing trajectory shortcutting stage). The trajectories such planners produce are functional, but they are also high-variance: running the planner on the same task will produce a different outcome every time (Fig.5).

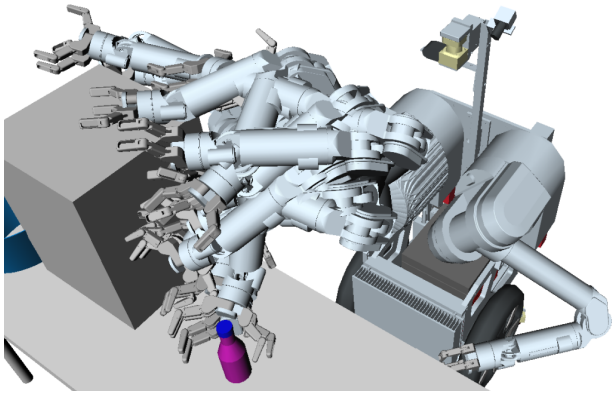

Figure 5: The unrepeatability of randomized planners. A snapshot (at the same time-point) for 20 trajectories that were obtained with a randomized planner (with path shortcutting) for the same reaching task.

Although this is not an issue when robots perform tasks on their own, placing a human in the loop raises a new requirement on the robot's motion: repeatability, and even *transparency of intent*: the robot's motion must make its intent clear to a human observer. We heard this directly from our users, e.g. "Assistance is good if you can tell that it [the robot] is doing the right thing": the robot's motion must convey that it is indeed achieving the intended goal.

A first step towards intent-transparent motion is producing repeatable, optimal motion. If the robot keeps its motions efficient and solves similar tasks in similar manners, then the user can get accustomed to the robot's ways. We create optimal motion via trajectory optimization, a technique that smoothly bends an initial trajectory out of collision (Ratliff et al. 2009). Although trajectory optimizers are known to struggle with high-cost local minima in the complex spaces induced by manipulation tasks, we were able to alleviate this issue by exploiting flexibility specific to manipulation problems (Dragan, Ratliff, and Srinivasa 2011) or by learning to place the optimizer in good basins of attraction from prior experience (Dragan, Gordon, and Srinivasa 2011). However, even with improved optimizers, there is still an open question of what to optimize in order to actually express intent – there are a lot of cases in which efficiency is not the correct metric for making the robot's goal obvious to a human.

Learning to generate intent-transparent trajectories. The robot's predicted trajectory to the goal must be intent-transparent. The robot's only manner of learning what this means is by interacting with the user: *good trajectory generators must adapt to how the user perceives the motion's intent*.

We envision that the robot will execute trajectories and ask the user to guess or predict its intent, as in Fig.6. With every guess, the robot will update its model of how the user predicts intent, much like how users can update their model of how the robot predicts their intent. We also find the idea of actively exploring the space of features that could affect

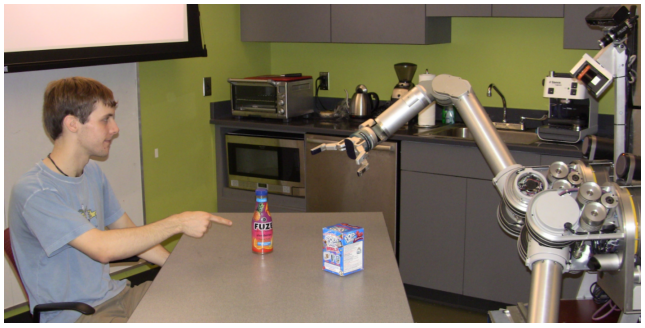

Figure 6: The robot can learn to produce intent-expressive motion by asking the user to predict its intent and adapting its optimizer to enable better predictions.

this prediction very interesting: the robot could vary certain features of its motion (e.g. its hand orientation or aperture in a reaching trajectory) in order to test whether the variation has an effect on the prediction (both in terms of correctness, as well as in terms of how fast the user is able to make this prediction as the robot is moving).

Conclusion

Our previous work on assistive teleoperation formalized assistance as the arbitration of the user's input and the robot's prediction of the user's intent, and analyzed the two challenges that the robot faces: to predict accurately and to arbitrate appropriately. Here, we gave a short overview of our findings, and discussed the important role interactive learning can play in improving assistance. First, the robot can learn better goal predictors by adapting them online, based on each user's way of teleoperating the robot. Second, in a different interaction paradigm, the robot can be the one doing the training: it can train the user to provide more intent-expressive input, in order to make predictions easier. Third, the robot can learn how the user predicts the its intent, in order to generate more intent-expressive trajectories to the identified goal. These avenues of future work are not only exciting in the context of assistive teleoperation: they are important for the while field of human-robot collaboration, for which robots must predict their collaborators' intentions, as well as make their own intentions clear.

Acknowledgments

This material is based upon work supported by NSF-IIS-0916557, NSF-EEC-0540865, ONR-YIP 2012, DARPA-BAA-10-28, and the Intel Embedded Computing ISTC. We thank the members of the Personal Robotics Lab at Carnegie Mellon for fruitful discussions and advice.

References

Aarno, D.; Ekvall, S.; and Kragic, D. 2005. Adaptive virtual fixtures for machine-assisted teleoperation tasks. In *IEEE ICRA*.

Aigner, P., and McCarragher, B. 1997. Human integration into robot control utilising potential fields. In *ICRA*.

Demiris, Y., and Hayes, G. 2002. Imitation as a dual-route process featuring predictive and learning components: a biologically plausible computational model. In *Imitation in animals and artifacts*.

Dragan, A., and Srinivasa, S. 2012. Formalizing assistive teleoperation. In *R:SS*.

Dragan, A.; Gordon, G.; and Srinivasa, S. 2011. Learning from experience in manipulation planning: Setting the right goals. In *ISRR*.

Dragan, A.; Ratliff, N.; and Srinivasa, S. 2011. Manipulation planning with goal sets using constrained trajectory optimization. In *ICRA*.

Fagg, A. H.; Rosenstein, M.; Platt, R.; and Grupen, R. A. 2004. Extracting user intent in mixed initiative teleoperator control. In *AIAA*.

Kim, D.-J.; Hazlett-Knudsen, R.; Culver-Godfrey, H.; Rucks, G.; Cunningham, T.; Port ande, D.; Bricout, J.; Wang, Z.; and Behal, A. 2011. How autonomy impacts performance and satisfaction: Results from a study with spinal cord injured subjects using an assistive robot. *IEEE Trans. on Systems, Man and Cybernetics*.

Kofman, J.; Wu, X.; Luu, T.; and Verma, S. 2005. Teleoperation of a robot manipulator using a vision-based human-robot interface. *IEEE Trans. on Industrial Electronics*.

Kuffner, J.J., J., and LaValle, S. 2000. Rrt-connect: An efficient approach to single-query path planning. In *ICRA*.

Li, M., and Okamura, A. 2003. Recognition of operator motions for real-time assistance using virtual fixtures. In *HAPTICS*.

Marayong, P.; Okamura, A. M.; and Bettini, A. 2002. Effect of virtual fixture compliance on human-machine cooperative manipulation. In *IROS*.

Ratliff, N. D.; Bagnell, J. A.; and Zinkevich, M. A. 2006. Maximum margin planning. In *ICML*.

Ratliff, N.; Zucker, M.; Bagnell, J.; and Srinivasa, S. 2009. Chomp: Gradient optimization techniques for efficient motion planning. In *ICRA*.

Srinivasa, S.; Berenson, D.; Cakmak, M.; Collet, A.; Dogar, M.; Dragan, A.; Knepper, R.; Niemueller, T.; Strabala, K.; Weghe, M. V.; and Ziegler, J. 2012. Herb 2.0: A robot assistant in the home. *Proceedings of the IEEE, Special Issue on Quality of Life Technology*.

You, E., and Hauser, K. 2011. Assisted teleoperation strategies for aggressively controlling a robot arm with 2d input. In *R:SS*.

Yu, W.; Alqasemi, R.; Dubey, R.; and Pernalete, N. 2005. Telemanipulation assistance based on motion intention recognition. In *ICRA*.

Ziebart, B. D.; Maas, A.; Bagnell, J. A.; and Dey, A. 2008. Maximum entropy inverse reinforcement learning. In *AAAI*.

Ziebart, B.; Ratliff, N.; Gallagher, G.; Mertz, C.; Peterson, K.; Bagnell, J.; Hebert, M.; Dey, A.; and Srinivasa, S. 2009. Planning-based prediction for pedestrians. In *IROS*.