

1-2013

# Teleoperation with Intelligent and Customizable Interfaces

Anca Dragan  
*Carnegie Mellon University*

Kenton C.T. Lee  
*University of Pennsylvania*

Siddhartha Srinivasa  
*Carnegie Mellon University*

Follow this and additional works at: <http://repository.cmu.edu/robotics>



Part of the [Robotics Commons](#)

---

## Published In

Journal of Human-Robot Interaction, 1, 3.

This Article is brought to you for free and open access by the School of Computer Science at Research Showcase @ CMU. It has been accepted for inclusion in Robotics Institute by an authorized administrator of Research Showcase @ CMU. For more information, please contact [research-showcase@andrew.cmu.edu](mailto:research-showcase@andrew.cmu.edu).

# Teleoperation with Intelligent and Customizable Interfaces

Anca D. Dragan, Siddhartha S. Srinivasa

Carnegie Mellon University

and

Kenton C. T. Lee

University of Pennsylvania

---

In this paper, we explore a class of teleoperation problems where a user controls a sophisticated device (e.g. a robot) via an interface to perform a complex task. Teleoperation interfaces are fundamentally limited by the indirectness of the process, by the fact that the user is not physically executing the task. In this work, we study intelligent and customizable interfaces: these are interfaces that mediate the consequences of indirectness and make teleoperation more seamless. They are intelligent in that they take advantage of the robot's autonomous capabilities and assist in accomplishing the task. They are customizable in that they enable the users to adapt the retargetting function which maps their input onto the robot. Our studies support the advantages of such interfaces, but also point out the challenges they bring. We make three key observations. First, although assistance can greatly improve teleoperation, the decision on how to provide assistance must be contextual. It must depend, for example, on the robot's confidence in its prediction of the user's intent. Second, although users do have the ability to provide intent-expressive input that simplifies the robot's prediction task, this ability can be hindered by kinematic differences between themselves and the robot. And third, although interface customization is important, it must be robust to poor examples from the user.

Keywords: Teleoperation, Motion Retargetting, Shared Autonomy

---

## 1. Introduction

Imagine a user teleoperating a robot manipulator to clear up the dining room table through a whole-body interface, a running example in this paper and illustrated in Fig.1. As he extends his arm, the robot extends its. This problem is fraught with a fundamental challenge, that teleoperation is an *indirect* process. The user is compelled to work through an interface, work with a robot whose motion might be kinematically very different from human motion, work without exact tactile or proprioceptive sensing, and potentially perform a task that challenges their dexterity. As a consequence, the process is not seamless, and often leads to tedious or even difficult teleoperation experiences. He is constantly correcting motion, stopping, restarting, and bumping into the table or objects.

Despite its challenges, teleoperation remains an important field that receives a lot of attention (detailed in Section 2). Interfaces have been proposed for teleoperation of various robots – ground vehicles (Fisher, McDermott, & Fagan, 2009), aerial vehicles (Baturone,

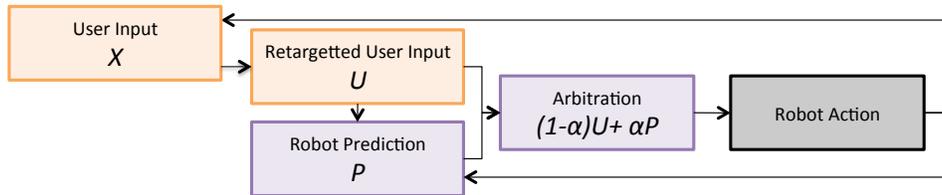
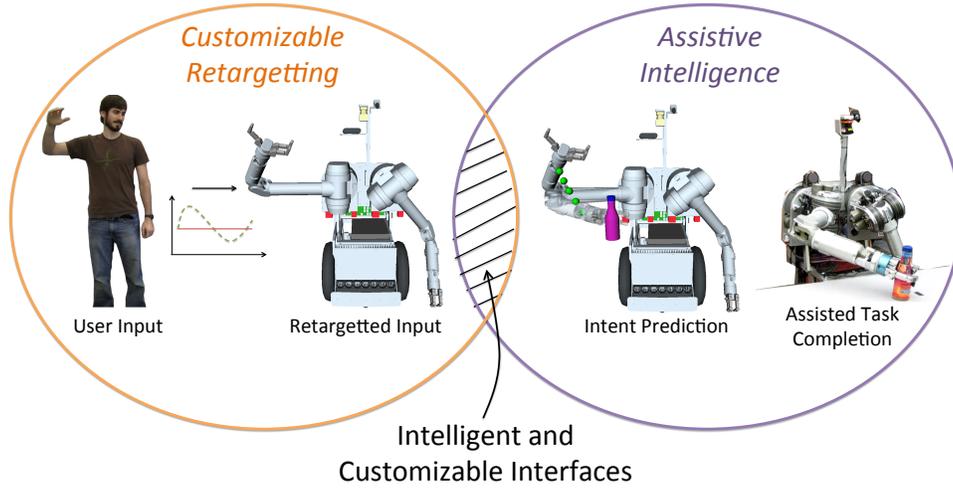


Figure 1. Classical teleoperation interfaces retarget the user’s input onto the robot. We investigate customizable and intelligent interfaces: ones in which users can customize the retargetting function that maps their input, and in which the robot uses its autonomous intelligence to assist the user in accomplishing the task. Such an interface arbitrates between the retargetted user input and the robot’s prediction of the user’s intent.

Rojo, Alczar, Lpez-Pichaco, & Nogales, 2003) and manipulators (Goertz, 1963; Howe & Kontarinis, 1994; D.-H. Kim, Kim, Kim, & Cha, 2001; C. Smith, Bratt, & Christensen, 2008) alike – through various input devices, from graphical interfaces (Paravati, Sanna, Lamberti, & Celozzi, 2011) to joysticks (C. H. Park & Howard, 2007) to motion capture (Hu, Meng, Liu, & Wang, 2003), and for various tasks, ranging from military reconnaissance (Chen, 2010; Kadous, Sheh, & Sammut, 2006) to telesurgery (Okamura, 2004) to caregiving at home (Labonte et al., 2006).

Although diverse, teleoperation interfaces share two fundamental aspects: 1) They map the user’s input onto the robot via a *retargetting function*. In doing so, they create an opportunity for improving teleoperation via *retargetting customization* –by enabling users to customize this retargetting function themselves. 2) They operate on devices that are often intelligent: robots are becoming better and better at doing more tasks autonomously. This creates an opportunity for *assistive intelligence* – for having the robot assist the user in accomplishing the task by using its autonomous intelligence. In this paper, we present a unified view of our work on improving teleoperation by taking advantage of these two common aspects.

Retargetting customization (Section 5) is the equivalent of *user adaptivity and personalization* in Intelligent User Interfaces (IUI) (Ehlert, 2003). It enables the user to explore the interface online, and to customize the retargetting function by providing new examples. We accomplish this by feeding the examples into a kernel-based function approximator that smoothly alters the retargetting function (much like when a rubber sheet is pulled at a point) while satisfying critical constraints (Dragan & Srinivasa, 2012c).

Assistive intelligence (Section 3) enables the robot to assist the user intelligently, by first predicting what the user’s intentions are, planning how it would satisfy those intentions, and injecting its plans harmoniously into the user’s commanded motion. In doing so, it parallels the concepts of *user modelling* and *plan recognition* (Waern, 1996) in IUI. We accomplish assistance via policy blending, where a machine learning algorithm predicts the user’s intentions, and the prediction is used to generate a robot control policy, which is then blended with the user’s input via an arbitration function which decides how aggressively the robot is assisting (Dragan & Srinivasa, 2012b).

Together, these algorithms provide a framework for intelligent and customizable interfaces (Fig.1). The user’s input  $X$  is customized to  $U$  via the online examples. The robot’s prediction  $P$  is then linearly blended with this input via an arbitration function  $\alpha$ . The robot’s actual actions are fed back, closing the loop.

The fact that retargetting customization and assistive intelligence can improve teleoperation, when done right, is clear. Doing them right, however, is not straightforward. Microsoft’s Office Assistant, Clippy™, is a great example of how assistance can go wrong even when the assistant has the best of intentions: Clippy always offers help, regardless of situation, and this become irritating to a lot of users (USA Today, 2002). Customization and intelligence come with many challenges, and our contribution lies in revealing and exploring these challenges.

In this paper, we make three key observations:

1. **Arbitration in assistance:** Although assistance can greatly improve teleoperation, arbitration must be contextual. It must depend, among others, on the robot’s confidence in its prediction of the user’s intent. (Section 3, based on (Dragan & Srinivasa, 2012b))
2. **Intent prediction in assistance:** Although users do have the ability to provide intent-expressive input that simplifies the robot’s intent prediction task, this ability can be hindered by kinematic differences between themselves and the robot. (Section 4, novel)
3. **Customization:** Although interface customization is important, it must be robust to poor examples from the user. (Section 5, based on (Dragan & Srinivasa, 2012c))

Customization and intelligence can bring important benefits to teleoperation, but they are not straightforward. Our work has merely touched upon the tip of the iceberg in exploring these interfaces. In Section 6, we discuss some of the limitations of our work as well as promising future challenges. Teleoperation is a fundamental challenge in robotics, good teleoperation enabling caregivers to remotely care for the elderly via personal robots like HERB from Fig.1, disabled patients to care for themselves via rehabilitation robots like wheelchairs with arms, NASA operators to help astronauts in space via robots on the space shuttle or the moon, or nuclear disaster relief agents to assist in a crisis without getting injured. Teleoperation is a step towards the broader goal of collaboration, where humans and robots are equal partners in performing a complex task and we are excited to be opening the way towards a deeper analysis in this area.

## 2. Related Work

**Assistive Intelligence.** Assistive intelligence can be thought of as a form of supervisory control (Sheridan, 1992). The history of assistive intelligence for teleoperation begins in 1963, with the work of Goertz (Goertz, 1963) on using teleoperated manipulators for handling radioactive material. Since then, research on this topic has proposed a great variety of methods for assistance, ranging from the robot having full control over all or some aspect of the motion (Rosenberg, 1993; Marayong, Li, Okamura, & Hager, 2003; Debus, Stoll, Howe, & Dupont, 2000; You & Hauser, 2011; Hauser, 2012; D.-J. Kim et al., 2011; Marayong, Okamura, & Bettini, 2002; Demiris & Hayes, 2002; Fagg, Rosenstein, Platt, & Grupen, 2004), to taking control (or releasing it) at some trigger (Kofman, W., Luu, & Verma, 2005; Li & Okamura, 2003; Shen, Ibanez-Guzman, Ng, & Chew, 2004; C. Smith et al., 2008; Anderson, Peters, Iagnemma, & Overholt, 2010), to never fully taking control (Crandall & Goodrich, 2002; Aigner & McCarragher, 1997; You & Hauser, 2011; Marayong et al., 2002; Aarno, Ekvall, & Kragic, 2005). For example, (Debus et al., 2000) propose that the robot should be in full control of the orientation of a cylinder while the user is inserting it into a socket. In (Kofman et al., 2005), the robot takes over to complete the grasp when close enough to the target. (Crandall & Goodrich, 2002) propose to mix the user input with a potential field in order to avoid obstacles.

Two control methods are predominant in assistance: potential fields and virtual fixtures. Potential field methods (e.g. (Crandall & Goodrich, 2002; Aigner & McCarragher, 1997; Yu, Alqasemi, Dubey, & Pernalet, 2005)) help the user avoid obstacles and reach the desired goal, while virtual fixture-based methods (e.g. (Marayong et al., 2002; Li & Okamura, 2003; Aarno et al., 2005; Yu et al., 2005)) are commonly used to guide the user along a predefined path. Other types of assistance are targeted towards dealing with time delay (C. Smith et al., 2008), or enable the user to choose manipulator grasps through an intuitive interface (Leeper, Hsiao, Ciocarlie, Takayama, & Gossow, 2012).

Despite this diversity in methods for assistance, they can all be seen as arbitrating between the user's retargeted input and the robot's prediction or knowledge of his intent (Dragan & Srinivasa, 2012b). This common lens enables a principled analysis of the factors that affect its performance.

**Retargeting Customization.** Customization of teleoperation interfaces ranges from customizable stiffness in haptic teleoperation (J. Park, Corteso, & Khatib, 2003) to customizable GUIs (Kawamura, Nilas, Mugeruma, Adams, & Zhou, 2003) and touch gestures (Paravati et al., 2011), to customizable brain interfaces (Hochberg, 2012). In this work, we study customization of the retargeting function that maps the user's input onto the robot. Different robots, users, and even tasks require different retargeting functions. We propose an online training process. The robot starts with an initial retargeting function, and as the user teleoperates he pauses the robot and provides example correspondences mapping user input to robot pose. Each example instantly bends the current retargeting function. In doing so, our work bridges two paradigms, one stemming from graphics and animation, and the other from teleoperation interfaces.

In the animation community, motion retargeting is used to map an actor's trajectory onto an animated character (e.g. (Gleicher, 1998; Hecker et al., 2008; Yamane, Ariki, & Hodgins, 2010; Ikemoto, Arikan, & Forsyth, 2009)). Although the traditional way to do so is via constrained optimization (Gleicher, 1998), recent work has proposed to learn a retargeting function offline, based on key pose correspondences (Yamane et al., 2010), or based on a different, manually retargeted trajectory (Ikemoto et al., 2009). Our approach extends this idea to an online setting, where users can provide examples by teleoperating

the robot or character using the current version of the retargetting function.

In the teleoperation community, unlike in animation, motion retargetting happens in real-time, and across the entire space rather than along a pre-set trajectory. While there are many interfaces for teleoperation, especially for robot manipulators (Kofman et al., 2005; Chang et al., 1999; Fukuda, Tsuji, Kaneko, & Otsuka, 2003), the retargetting function is usually designed a-priori and cannot be changed. Some robots, however, such as the DaVinci from Intuitive Surgical (Rockall & Darzi, 2003), allow users to change the retargetting function online, albeit only by a constant (or global) offset, equal throughout the input space. The surgeon operating the DaVinci can use a clutch to stop teleoperation and reset himself to a comfortable position (Rockall & Darzi, 2003), thus offsetting the origin of the retargetting function. Our approach extends this idea to variable offsets: the user can modify different parts of the space in different ways, and can reach *any* (smooth) retargetting function.

Our method is an instance of Learning from Demonstration (Argall, Chernova, Veloso, & Browning, 2009), where the robot learns a policy based on examples. A significant amount of work in this area (Atkeson, 1991; Kaiser & Dillmann, 1996; Grollman & Jenkins, 2008) has applied function approximation techniques to learn the policy from examples provided by a human or another robot.<sup>1</sup> We adopt the same principle, and use an extension to Online Learning with Kernel Machines (Kivinen, Smola, & Williamson, 2004) as the function approximator.

### 3. Assistive Intelligence: Arbitration must be contextual.

The recent progress in autonomous robotics creates an opportunity for making teleoperation easier and more seamless by bringing autonomy into the teleoperation process. Ideally, while the user is teleoperating, the robot should be doing what the user *intends*: a rough reaching motion of the user’s arm should map to the robot grasping the bottle, then a rough motion to the side should map to the robot placing the bottle in the bin. In order to realize this *assistive* teleoperation paradigm, the robot is facing two challenges. Its first challenge is predicting the user’s intent – what is the user trying to do, and how would he want the robot to do it? Its second challenge is far more subtle: once this prediction is made, the robot must decide how to use it: how should the robot arbitrate between the prediction and the retargetted input?

The arbitration function  $\alpha$ , which decides the next action given the input  $U$  and the prediction  $P$ , 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. In trading off between not over-assisting (providing unwanted assistance) and not under-assisting (failing to provide needed assistance), the arbitration lies on a spectrum: On the one hand, the assistance could be very timid, with  $\alpha$  taking small values even when the robot is confident in its prediction. On the other hand, it could be very aggressive:  $\alpha$  could take large values even when the robot does not trust the predicted policy.

How does the *aggressiveness* of arbitration affect the performance of assistance? A natural hypothesis is that aggressive assistance is more efficient, i.e. the time it takes to complete the task is lower than with timid assistance. This is in line with (Marayong et al., 2002), who found that more autonomous assistance is faster, and with (You & Hauser,

<sup>1</sup>The motion retargetting problem is also related to the correspondence problem (Argall et al., 2009): that of mapping a demonstration onto a robot. The latter, however, goes beyond motion, to task-level differences, sensing differences, and capability differences in general.

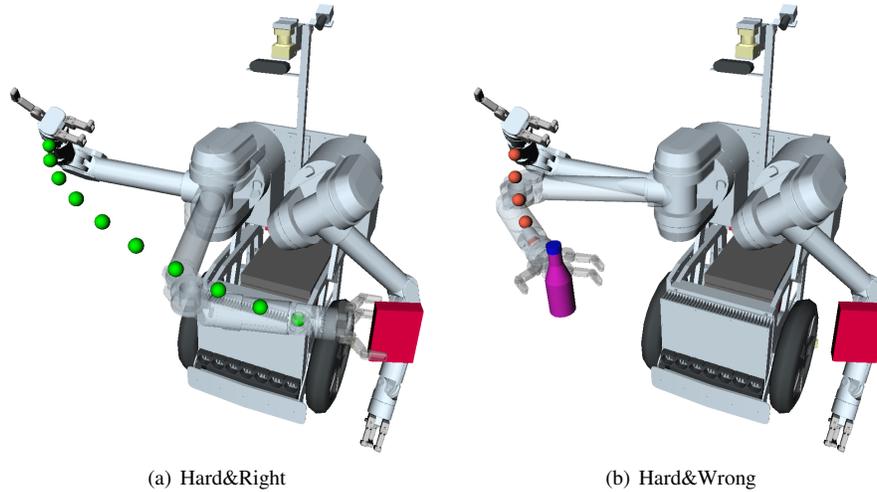


Figure 2. Hard tasks, with the goal at the limit of the reachable area, for both right and wrong prediction. Figure from (Dragan & Srinivasa, 2012b).

2011) who concluded that it is also what users prefer. However, a study by (D.-J. Kim et al., 2011) showed that users preferred a manual teleoperation mode despite a higher cognitive load. We believe that rather than studying autonomy level or aggressiveness on their own, we must study the interaction among aggressiveness and other factors, like *task difficulty* (users might like assistance more on harder tasks, when they need it more) and *prediction correctness* (users might dislike assistance when the robot incorrectly predicts their intent). With this in mind, we designed a study that aims at understanding these interactions and shedding some light on the matter of aggressiveness in assistance.

### 3.1 Experimental Design

We used the HERB platform (Srinivasa et al., 2012) for our experiments. HERB is comprised of two Barrett WAM arms mounted on top of a Segway RMP 200 platform. The WAM arm joints are roughly anthropomorphic, with a total of 7DOFs. HERB’s joint limits, as well as its link lengths and proportion, differ from those of an average human.

We tasked 8 users with teleoperating HERB to grasp an object from a table, as in Fig.1. There were always two graspable objects, and we gave the user, for every trial, the farther of the two as the goal. We implemented a whole-body interface that tracks their skeleton (OpenNI, [www.openni.org](http://www.openni.org)), yielding an arm configuration which serves as the user input  $U$ . At every time step, the robot makes a real-time prediction of the goal and the policy to it (that minimizes length in configuration-space), leading to  $P$ , and combines the two via the arbitration function  $\alpha$ .

**Hypotheses.** We test the following two hypotheses:

1. Main effects: *Prediction correctness, task difficulty, and aggressiveness of assistance each has a significant effect on task performance.*
2. Interaction effects: *Aggressive assistance performs better on hard tasks if the robot is right, while the timid assistance performs better on easy tasks if the robot is wrong.*

**Manipulated Variables.** We manipulated prediction correctness by using a simple, easy

to manipulate goal prediction method (Dragan & Srinivasa, 2012b) that predicts the closest goal. We setup **wrong** conditions at the limit of the robot being wrong yet rectifiable: We place the intended object further. The robot stubbornly keeps predicting (and, depending on  $\alpha$ , going to) the wrong goal until the user makes his preference clear by providing an input  $U$  closer to the correct goal. We setup **right** conditions by explicitly informing the robot of the user’s intended goal.

We manipulated task difficulty by changing the location of the two objects and placing the target object in an easily reachable location (e.g. grasping the bottle in Fig.2(b) makes an **easy** task) vs. a location at the limit of the interface’s reachability (e.g. grasping the box in Fig.2(b) is a **hard** task). This leads to four types of tasks: **Easy&Right**, **Easy&Wrong**, **Hard&Right** and **Hard&Wrong**.

Finally, we manipulated the aggressiveness of the assistance by changing the arbitration function, and used a distance-based measure of confidence. As the user makes progress towards the predicted object, the confidence increases. We had two assistance modes, shown in Fig.3(a): the **timid** mode increases the assistance with the confidence, but plateaus at a maximum value, never fully taking charge. On the other hand, the **aggressive** mode eagerly takes charge as soon as the confidence exceeds a threshold.

**Subject Allocation.** We chose a within-subjects design, enabling us to ask users to compare the timid and aggressive mode on each task. Each of our 8 participants (all students aged 18-28, all familiar with robots or video games but not with our system, 4 males and 4 females) executed both modes on each of the four types of tasks once. To avoid ordering effects, we used a balanced Latin square for the task order, and balanced the order of the modes within each task.

**Dependent Measures.** We measure the performance of assistance in two ways: the amount of **time** each user took to complete the task under each condition, and each user’s **preference** for the timid vs. the aggressive mode on each task type (on a 7 point Likert scale where the two ends are the two choices). We expect the two measures to be correlated: if an assistance mode is faster on a task, then the users will also prefer it for that task. We also asked the users additional questions for each condition, about how helpful the robot was, how much its motion matched the intended motion, and how highly they would rate the robot as a teammate.

**Covariates.** We identified the following confounds: the users’ initial teleoperation skill, their rating of the robot without assistance, and the learning effect. To control for these, users went through a *training phase*, teleoperating the robot without assistance for a simple pickup task three times. This partially eliminated the learning effect and gave us a baseline for their timing and ratings. We used these as covariates, together with number of tasks completed at any point — a measure of prior practice.

### 3.2 Analysis

**Teleoperation Timing.** The average time per task was approximately 28s. Since our experimental design was factorial, we performed a factorial repeated-measures ANOVA. We used Bonferroni corrections for multiple comparisons (Tukey corrections yielded the same results). The model fit the data well ( $R^2 = 0.66$ ). In line with our first hypothesis, we found main effects for all three factors: **hard** tasks took 22.9s longer than **easy** ones ( $F(1, 53) = 18.45, p < .001$ ), tasks where the policy was **wrong** took 30.1s longer than when **right** ( $F(1, 53) = 31.88, p < .001$ ), and the **aggressive** mode took overall 19.4s longer than the **timid** ( $F(1, 53) = 13.2, p = .001$ ). We found a significant interaction effect between aggressiveness and correctness, showing that when wrong, being timid is

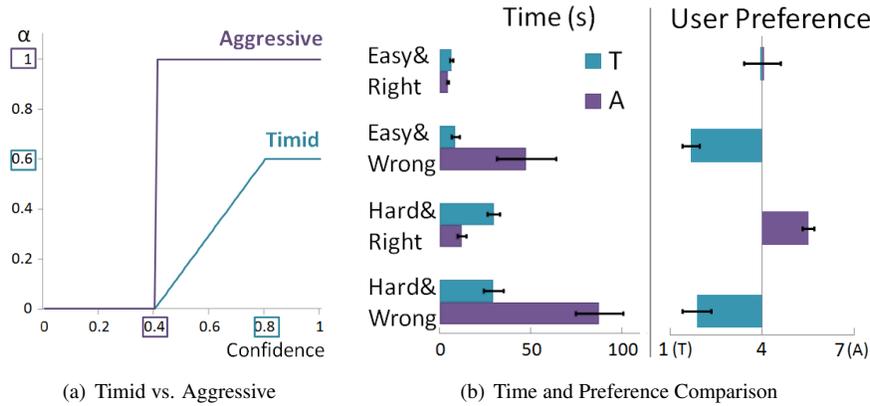


Figure 3. The arbitration functions (a) and the results of the study (b). Figure from (Dragan & Srinivasa, 2012b).

significantly better than being aggressive. This is reinforced in Fig.3(b), which compares the means and standard errors on each task: the timid mode is better on both **Easy&Wrong** and **Hard&Wrong**. The timid mode performed about the same on **Easy&Right**, and, as expected, worse on **Hard&Right** (the time taken for aggressive is smaller than for timid for every user). Surprisingly, the interaction effect among all factors was only marginally significant ( $F(1, 53) = 2.63, p = .11$ ). We believe that increasing our user pool would strengthen this effect.

To conclude based on this regression that the timid mode is overall better would be misleading, because it would assume that the robot is wrong in 50% of the tasks (in general, either by predicting he wrong goal, or by computing a motion that, for example, collides with an unseen obstacle). Our data indicates that the aggressive mode is overall more efficient if the robot is wrong in less than 16% of the cases. However, efficiency is only part of the story: as the next section points out, *some users are more negatively affected than others by a wrong robot policy*.

**User Preferences.** Fig.3(b) also shows the users' preferences on each task, which indeed correlated to the timing results (Pearson's  $r(30) = .66, p < .001$ ). The outliers were users with stronger preferences than the time difference would indicate. For example, some users strongly preferred the timid mode on **Hard&Wrong** tasks, despite the time difference not being as high as with other users. The opposite happened on **Hard&Right** tasks, on which some users strongly preferred the aggressive mode despite a small time difference, commenting that they appreciated the precision of the autonomy. On **Easy&Right** tasks, the opinions were split and some users preferred the timid mode despite a slightly longer time, motivating that they felt more in control of the robot. Despite the other measures (helpfulness, ranking as a teammate, etc.) strongly correlating to the preference rating ( $r(30) > .85, p < .001$ ), they provided similar interesting nuances. For example, the users that preferred the aggressive mode on **Easy&Right** tasks because they liked having control of the robot were willing to admit that the aggressive mode was more helpful. On the other hand, we also encountered users that preferred the aggressive mode, and even users that followed the robot's motion while aggressive, not realizing that they were not in control and finding the motion of the robot to match their own very well (i.e. the predicted policy

*P* matched what they intended, resulting in seamless teleoperation).

Overall, although difference in timing is a good indicator of the preference, *it does not capture a user's experience in its entirety*. First, some users exaggerate the difference in preferences. Second, some users prefer the timid mode despite it being slightly less efficient. Third, *assistance shouldn't just be quick – it should also be intent-transparent*. Our users commented that “Assistance is good if you can tell that [the robot] is doing the right thing”.

### 3.3 Main Lesson

Our study revealed evidence for both studies supporting timid (D.-J. Kim et al., 2011) and those supporting aggressive assistance (Marayong et al., 2002; You & Hauser, 2011). The explanation is that *arbitration must be contextual*: it must depend on the robot's confidence in itself (in its prediction, in how well it can achieve the task the user wants to achieve), as well as on the robot's confidence in the user (in how good the user is at achieving the task). If the robot isn't sure, it should keep arbitration low. If the task is hard, it should ramp the arbitration up. Moreover, users have different preferences on assistance when the task is easy and the robot's prediction is correct: half of our users preferred a timid mode, and half preferred and aggressive mode. As a result, arbitration must also adapt to the type of user.

## 4. Assistive Intelligence: Intent prediction can be improved through intent-expressive input.

So far, we looked at the robot providing assistance to the user. But intelligent interfaces become a medium for collaboration, during which both entities are interested in achieving the task. Here, we explore how users can assist the robot, so that the robot can better assist them.

Intelligent interfaces predict the user's intent based on the user inputs so far (Dragan & Srinivasa, 2012b). This prediction problem is similar to tracking and recognizing human activity. However, assistance during teleoperation differs in a key aspect: the human being tracked is directly affected by this prediction. Getting it wrong means the robot will not only fail to assist with the task, but might even become a hindrance. Therefore users have an incentive to change their behavior in a way that makes their intent more transparent to the predictor. *While a lot of research focuses on improving predictions through better algorithms (Ziebart et al., 2009), we believe that improving the user's input itself, by exploiting this incentive, is just as important.*

This has a strong analogy in IUI and the idea of *intended plan recognition* (Waern, 1996), in which the observed actor makes the plan easier to recognize. It also has an analogy to teaching robots new skills: while most research focuses on improving the learner (Argall et al., 2009), we take a human-centric approach and improve the teacher by making examples more suitable to the robot (Dragan & Srinivasa, 2012a, 2011).

Our central hypothesis is that users can be trained to provide input that increases prediction performance – users naturally get better at using interfaces, be it search systems (C. L. Smith & Kantor, 2008), sensory rearrangement (Biocca & Rolland, 1998), or brain-computer interfaces (Schalk et al., 2008). We envision that the robot can autonomously give the user relevant practice scenarios, in the spirit of work in learning from demonstration on robots actively probing users (Chernova & Veloso, 2009; Cakmak & Thomaz, 2012).

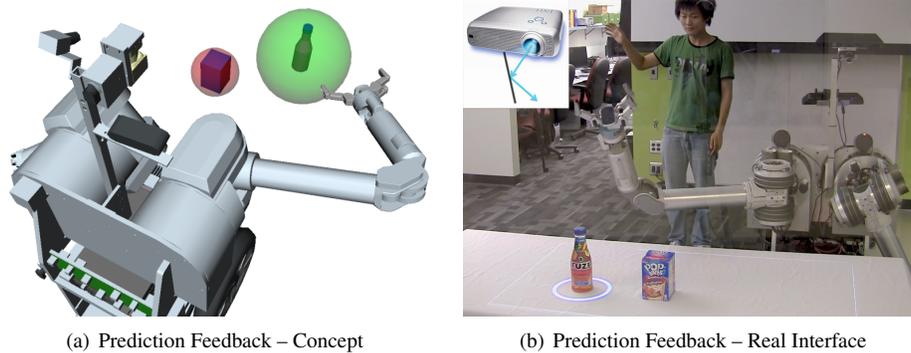


Figure 4. During the training phase, the robot provides the users feedback on their input by showing them the predicted goal and the confidence.

#### 4.1 Experimental Design

In the study from Section 3, we asked users to teleoperate the robot as best they can, never asking them to be intent-expressive. Next, we ran a study that tests the improvement in prediction brought about by explicitly asking users to provide intent-expressive input. We tasked the users with teleoperating HERB through the same whole-body interface as in the studies above to reach one of two goals on the table in front of it. The robot used its predictor (Dragan & Srinivasa, 2012b) to identify the desired goal at every point based on the inputs thus far.

**Manipulated Variables.** We manipulated *prediction awareness*: whether users teleoperate normally, or whether they are told explicitly to provide input that helps the robot in its intent prediction. Because what is intent-expressive to the robot (i.e. what improves intent prediction) is not necessarily the same as what is intent-expressive to a human, we designed a *training phase* during which the robot provides the participants feedback on their input by indicating the predicted goal and its assigned probability, as in Fig.4. Each user gets a set number of scenarios (configurations of the two goals together with a target goal assignment) for training.

To make the process easier for the users, we start the training process with a simple, two-dimensional interface (Fig.5) rather than directly with the full-dimensional one. Like in the real interface, the users are presented with two potential goals, and construct trajectories from the starting point to the given goal by placing intermediate points along the way. With each point, they see which goal the predictor would choose, along with the actual probability assigned to it. The users can improve the predictor’s confidence (i.e. increase the probability of the correct goal) by exaggerating the trajectory towards the left or right side rather than following the straight line, as in Fig.5.

**Dependent Measures.** Our manipulation leads to *two evaluation phases*: one before and one after training. Each phase consists of the same 6 scenarios, leading to a total of 12 trajectories executed in normal teleoperation, without this explicit feedback on what the robot is predicting. We measure the *intent-expressiveness* of each trajectory by averaging a prediction score across a set number of points representative of the trajectory’s shape (calculated by interpolating between measured samples). Because intent-expressive trajectories should lead to accurate predictions with high confidence, we choose a prediction

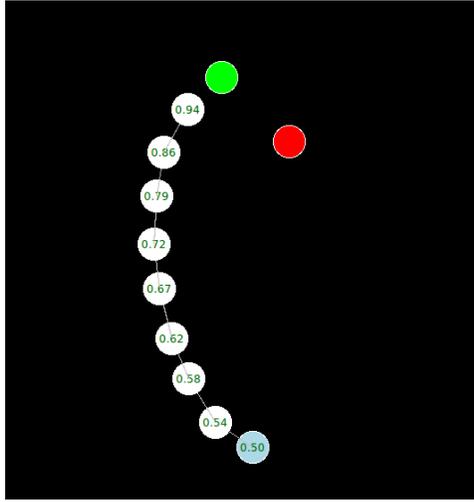


Figure 5. In the first part of the training phase, the users are exposed to a simplified, 2D interface in order to understand how the geometry of the trajectory influences predictions.

score that takes both the accuracy and the actual probability assigned into account. The score ranges from -1 to 1, where -1 represents predicting the wrong goal with high probability, 1 represents predicting the correct goal with high probability, and 0 represents a 50-50 prediction.

**Hypothesis.**

1. Users provide more intent-expressive input after training: *The intent-expressiveness is significantly higher for trajectories from the post-training evaluation phase.*

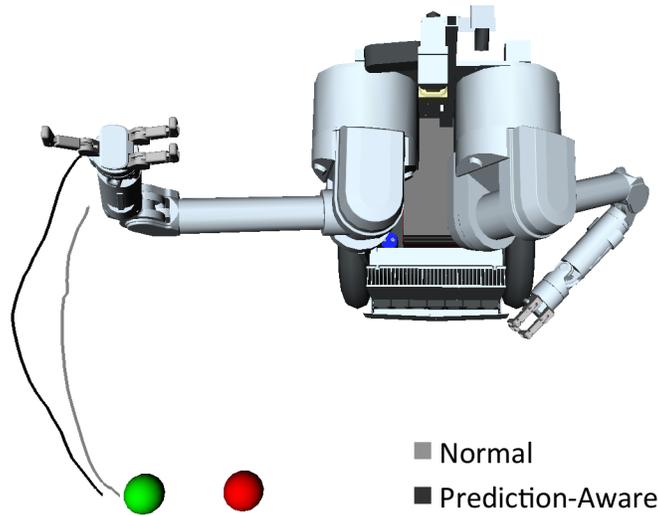
**Subject Allocation.** We chose a within-subjects design because we wanted a direct comparison between performance before and after training. We had 4 novice users (2 male and 2 female, all students aged 18-28, all familiar with robots or video games but not with our system).

**Covariates.** The teleoperation skill of each user could have an effect on how well they can teleoperate the robot along the envisioned intent-expressive trajectory, and thus influence the improvement we see in the predictions. To bring users to a similar skill, we had a *practice phase* before the first evaluation, during which the users got accustomed to teleoperating. We timed the users during the first evaluation, which is a measure of the teleoperation skill after practice.

Furthermore, the side (left or right) of the target goal can also affect the improvement: goals on the left require trajectories exaggerated towards the left for intent-expressiveness, which is much harder with HERB’s right arm than exaggeration towards the right.

4.2 Analysis

Because the design consists of one factor, but has several covariates, we analyzed the data via a one-way repeated-measures ANOVA as opposed to a simple *t*-test (here, the 6 tasks are repeated across users and the user assignment is repeated across evaluation phases), which supported our hypothesis: the main effect was significant ( $F(1, 23) = 11.67$ ,  $p = .002$ ), indicating that the intent-expressiveness when the users are prediction-aware



*Figure 6.* The average end-effector trajectory taken by the users for a task before (grey) and after (black) intent-expressiveness training. The users exaggerate the trajectory to the goal to make it easier for the robot to predict which goal they are targeting.

is significantly higher than when they teleoperate normally.

All the users learned to exaggerate their trajectories towards the right or left in the 2D interface, producing trajectories like the one from Fig.5. After training on the real robot, the users exaggerated the end-effector trajectories in a similar way. Fig.6 shows this intent-expressive exaggeration as compared to the normal way of teleoperation: we compute the average trajectory for one of the tasks across all users for each evaluation phase. The difference in trajectories is consistent with the intent-expressive exaggeration from the 2D interface, suggesting a purposeful change in the geometry rather than a practice effect.

Although the intent-expressiveness improvement is statistically significant, the effect size is small ( $\eta^2 = .023$ ): the average intent-expressiveness score increased from 0.19 to 0.27. The danger of a small improvement is that the robot might not be able to use it to provide better assistance.

### 4.3 Follow-up Study

The user’s response to the 2D interface, along with the difference in geometry for trajectories with the real interface, suggest that the users understood how to modify their input to improve intent predictions. Because of this, we believe that the limitation in the improvement arises from the limitations of the teleoperation interface. Since HERB has different kinematics than the user, the desired exaggeration in the trajectory might be hard to obtain. In a follow-up study, we wanted to explore the improvement in intent-expressiveness when this impediment is removed: when the robot has the same kinematics as the user, the retargeting function is the identity mapping and the user gets much better control over what the robot executes. This, in a way, mimics a very good retargeting function for HERB.

We ran the same study from above, now with a simulated robot that matches the user’s kinematics. We had a new set of 4 users that went through the same process as before. We

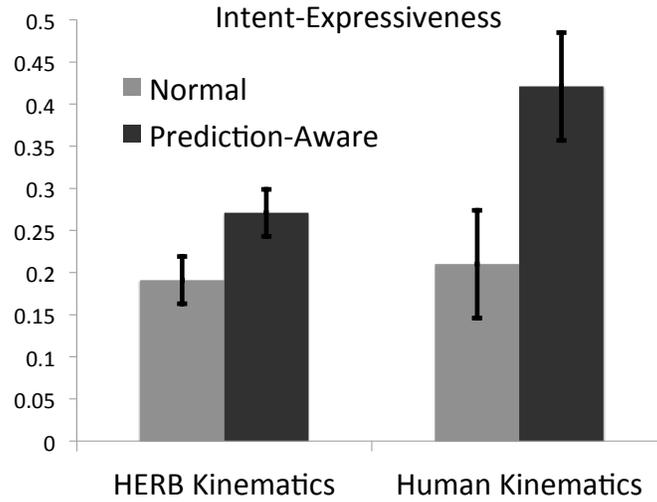


Figure 7. A comparison of average intent-expressiveness ( $y$  axis is our dependent measure, the intent-expressiveness score defined in Section 4.1) of trajectories when teleoperating HERB vs. a simulated robot with human-like kinematics. The scores are similar when the users teleoperate normally, but the improvement from explicitly attempting to provide intent-expressive input is higher for the easier, human-like kinematics interface. Error bars here represent the standard error of the mean with repeated-measures corrections as suggested by (Cousineau, 2005)

expect that these users would achieve a much larger improvement in intent-expressiveness. **Additional Hypotheses.**

2. With the new kinematics, users still provide more intent-expressive input after training: *The intent-expressiveness is significantly higher for trajectories from the post-training evaluation phase.*

3. The improvement is larger with the new kinematics: *The difference in intent-expressiveness for tasks with human kinematics is significantly larger than the difference for tasks with HERB kinematics.*

**Follow-up Study Analysis.** An equivalent ANOVA for the data with the human kinematics interface did reveal the same main effect of prediction awareness ( $F(1, 23) = 15.47$ ,  $p < .001$ ). The effect size is much larger ( $\eta^2 = .097$ ): intent-expressiveness went from 0.21 with normal teleoperation to 0.42 with intent-expressive teleoperation. To test the second hypothesis, we ran a repeated-measures ANOVA with the improvement as the dependent variable, which yielded a main effect ( $F(1, 46) = 5.18$ ,  $p = .027$ ), supporting the significant improvement in the means (Fig.7).

#### 4.4 Main Lesson

Making users aware that they should be intent-expressive (and training them to understand what that means for the robot) leads to significant improvements in the intent predictions the robot makes. However, these improvements are only large enough to make a practical difference in the quality of assistance when the users are able to easily produce the desired alterations to the retargetted input. Unless users have a lot of experience with the system, this puts a large burden on the retargetting function to compensate for the difference in kinematics.

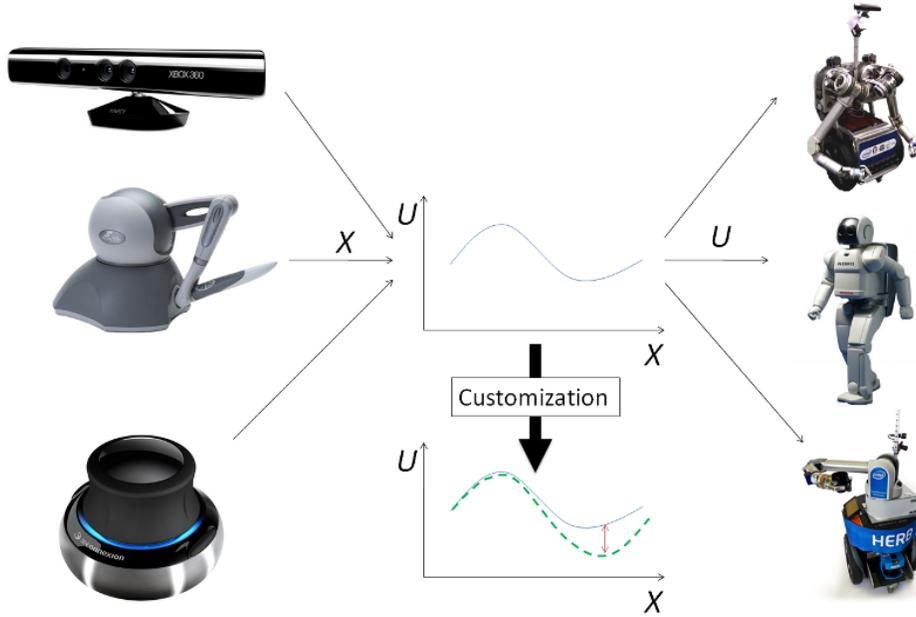


Figure 8. We propose a method that enables users to customize the retargeting function mapping their input onto the robot. Figure adapted from (Dragan & Srinivasa, 2012c).

## 5. Retargetting customization must be robust to poor examples.

The retargetting function mapping the user’s input onto the robot is a fundamental part of teleoperation – even in the presence of assistance, a counter-intuitive retargetting function can lead to counter-intuitive robot behavior and even failure to predict the correct intent. Unfortunately, there is no universally good retargetting function. As outlined in Section 2, this function must be different for different robots, users and tasks. Because of this need for adaptation to different contexts, we explore the idea of enabling users to customize the retargetting function (see Fig.8). Our method can be used to adapt any prior function to the three factors stated above: the kinematics of a particular robot, the user’s preferences, or the specific task at hand.

### 5.1 The Training Process

Our goal is to enable users to produce a retargetting function  $f : X \mapsto U$  that maps their pose  $X$  to a configuration for the robot,  $U$ . The process through which users do so, outlined in Fig.9, happens online. The robot starts with an initial retargetting function, i.e. it has a smooth *prior*  $f_{prior}(X)$ . We decompose  $f(x)$  into this prior and an offset function:  $f(X) = f_{prior}(X) + o(X)$ . As the user teleoperates, he pauses the robot and provides examples that instantly bend  $o(X)$  via function approximation. This changes the retargetting function online, enabling the user to base his future examples on the outcome of previous ones.

In what follows, we describe the requirements on the online function approximator for  $o(X)$ , and briefly outline an algorithm that satisfies these requirements.

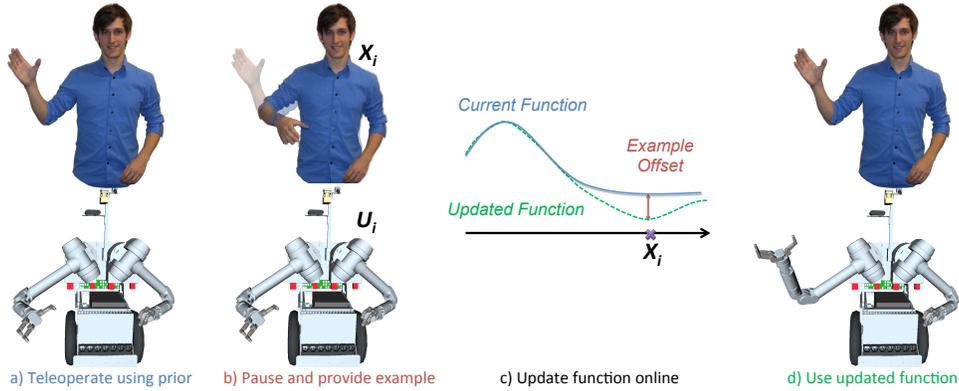


Figure 9. The training process. Figure adapted from (Dragan & Srinivasa, 2012c).

### Problem Requirements.

1. **Online:** The user should be able to provide further examples based on the outcome of the previous examples. Therefore, changes should be incorporated online, and in real-time.
2. **Smooth:** The function  $o(x)$  needs to be smooth, as there can be no sudden shifts that cause the robot to move abruptly while the user is barely changing pose.
3. **Propagation decay:** Different areas of the input space require different offsets. Therefore, examples typically have local support and should decay as we move away from them in the input space to allow different alterations in different areas.
4. **Propagation of larger changes further:** Large changes have different semantics than small changes when training the retargetting function. Usually, large changes need to be propagated wider through the space (i.e. with a very wide kernel), because they are meant to change the prior in a fundamental way – the prior is in some way offset by a considerable amount from the desired mapping. Small changes, however, should affect the space only locally (i.e. with a narrow kernel), because they usually represent local corrections.
5. **Bound maximum curvature:** Safety and hardware constraints suggest a bound on the acceleration of the robot. This is equivalent to bounding the curvature of the retargetting function.

### The Function Approximator.

We adapt the function approximator from (Kivinen et al., 2004). The algorithm satisfies the first three requirements from above: if it works well online, smoothness is ensured by the function representation, and propagation decay is the result of the gradient of the loss function. The approximator represents the function  $o(X)$  as

$$o(X) = \sum_i \alpha_i k(X_i, X)$$

for some input poses  $X_i$ , and  $\alpha_i \in \mathbb{R}$ .  $k$  is a reproducing kernel function, a common choice for  $k$  being a Radial Basis Function (RBF):  $k(X_i, X) = \exp(-\frac{1}{2}w(X_i - X)^2)$

With every example mapping a user input  $X_i$  to a desired retargetted input  $U$ , the method discounts previous data and adds an additional term which places an offset at  $X_i$

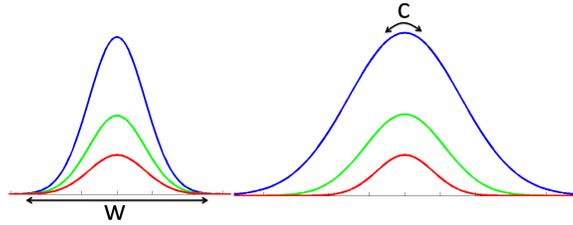


Figure 10. Comparison of kernels with fixed width vs. fixed max curvature. Fixing the maximum curvature makes larger changes propagate further. Figure from (Dragan & Srinivasa, 2012c).

proportional to the desired change and propagates to the rest of the space according to the kernel width corresponding to  $w$ :

#### RBF-LEAST-SQUARES

```

1  $o \leftarrow 0$ 
2 for  $i \leftarrow 1$  to  $N$ 
3     do  $\alpha_i \leftarrow U_i - o(X_i)$ 
4      $o \leftarrow (1 - \eta\lambda)o + \eta\alpha_i k(X_i, \cdot)$ 

```

RBF-LEAST-SQUARES uses a fixed kernel width  $w$  for all examples. The changes propagate in the same way to the rest of the space, independently of the nature of the change, as in Fig.10(left). This violates requirements 4 and 5 from above. We resolve these two requirements by keeping the curvature  $c$  at the center of the kernels constant. Fig.10 shows that this enables us to enforce the first requirement: larger changes do propagate further. With respect to the curvature requirement, we assume that examples are sparse enough that the curvature at the center of the kernel is negligibly affected by other examples. Then, the maximum curvature of the offset function  $o(X)$  occurs at an example point  $X_i$ , and is therefore  $c$ . As a result, the maximum curvature of  $f(X)$  will only depend on  $c$ , and can be selected to satisfy the desired bound. We show in (Dragan & Srinivasa, 2012c) that keeping  $c$  constant is equivalent to setting  $w$  as:  $w = \frac{c}{\alpha}$

## 5.2 Experimental Design

Does customization help? Does our method enable users to successfully train prior retargeting functions? To test the method’s usability, we ran a study that asked novice users to train two priors: a natural prior, and “bad” prior – one that we specifically designed to be faulty. Our main goal was to test that our novice users can improve upon the bad prior. We further wanted to investigate if and how novice users would customize the natural prior. We do not necessarily expect novice users to successfully customize a good prior, because we believe this requires some experience with the system: errors in examples are easier to make when the desired changes are subtle, in particular if the user does not have a particular task in mind.

**Manipulated Variables.** We manipulated two factors: *training* (whether the user trains the prior or uses it as-is) and *prior quality* (which function acts as the prior). We used two priors in this study. The first one is a natural prior (which we call *NP*), obtained by taking advantage of HERB’s fairly anthropomorphic kinematics: the function directly maps the operator’s shoulder and elbow joint angles (obtained by processing the OpenNI

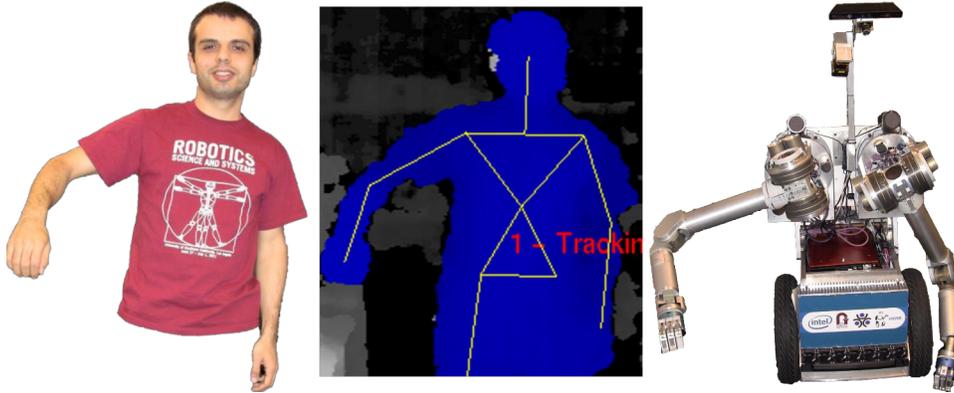


Figure 11. In this example interface, the user’s input consists of the joint angles extracted from the Kinect skeleton, and the retargetting function maps these onto HERB. Figure adapted from (Dragan & Srinivasa, 2012c).

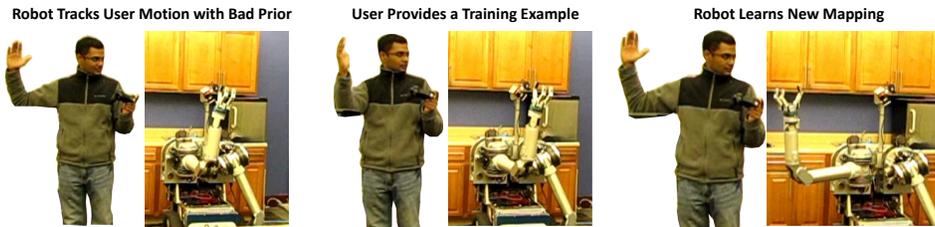


Figure 12. Training with an unnatural prior. Figure from (Dragan & Srinivasa, 2012c).

(www.openni.org) skeletal tracker output) onto the robot (see Fig.11). To manipulate the quality and create a bad prior, we added a large constant offset to the shoulder pitch:  $UP(X) = NP(X) + c$ . Fig.12 shows a user’s configuration with its retargetted configuration on the robot via  $UP$ . This leads to four experimental conditions and four retargetting functions whose performances we compare:  $NP$ ,  $NP - T$ ,  $UP$ , and  $UP - T$  (where  $x - T$  refers to the function obtained by training the prior  $x$ ).

**Subject Allocation.** We chose a within-subjects design, where each of our 6 users (all students aged 18-28, all familiar with robots or video games but not with our system) was assigned to all four conditions, in order to enable paired comparisons among the conditions. We first had an introductory phase, where the users interacted with the natural prior  $NP$ . We then tasked the users with training both priors. We instructed them to train the functions for general purposes, thinking about any task they might want to achieve with the robot, and stop when they are satisfied with the retargetting function they reached. After training, we moved the robot in front of a table and tasked the users with testing each of the four functions (the two untrained and the two trained) on a simple manipulation task – grasping a bottle on the table. To avoid ordering effects, we randomized the order of the conditions. Each user attempted each task three times, in order to reduce the variance in the data.

**Dependent Measures.** For each condition, we measured the **success rate** and the **time** to completion for the grasping attempts. Potential failures included knocking over the target

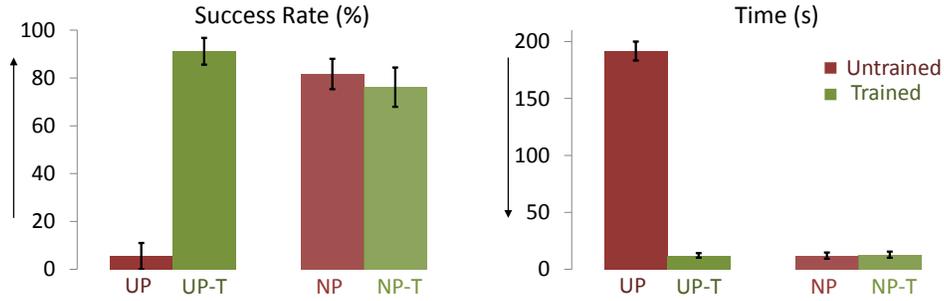


Figure 13. Left: The mean success rate on within condition. Right: The mean time to completion within each condition. Figure from (Dragan & Srinivasa, 2012c).

object, colliding with the table, or being unable to complete the task within 200 seconds.

### Hypotheses.

1. Correct manipulation of priors: *Prior quality has a significant positive effect on the dependent variables.*
2. Training improves overall performance: *Training has a significant positive effect on the dependent variables.*
3. The trained unnatural prior does not perform worse than the natural prior and the trained natural prior (i.e. through training, users can bring the bad prior up to the level of the natural one). We define “does not perform worse than” using the concept of “non-inferiority” (Lesaffre, 2008; Scott, 2009): a distribution is non-inferior to another if their difference is significantly greater than a negative margin.  $UP - T$  is non-inferior to  $NP$  and  $NP - T$ .

### 5.3 Analysis

We performed a factorial repeated-measures ANOVA on the success rate with Bonferroni corrections for multiple comparisons (Tukey corrections yielded the same results), which resulted in a good fit of the data ( $R^2 = 0.845$ ). In line with our first two hypotheses, we found that both factors had significant effects. Prior quality improved the success rate by 30.6% ( $F(1, 20) = 22.07, p < .001$ ), and training resulted in an average improvement of 40.1% ( $F(1, 20) = 37.9, p < .001$ ). The interaction term was also significant ( $F(1, 20) = 49.02, p < .001$ ), the post-hoc analysis revealing that indeed,  $UP$  was significantly worse than the other three conditions. The timing results were analogous, and Fig.13 plots the means for both success and time over the four conditions. We saw a slight mean decrease in success rate of  $NP - T$ . This could be due to measurement noise, to the fact that the user’s weren’t targeting a particular task in their training, or, as we discuss below, to poor examples.

The largest outlier in our ANOVA was a user who performed much worse on the trained natural prior,  $NP - T$ , than on the untrained prior  $NP$ . This user commented during the study that proprioception was an issue (“It’s really difficult sometimes because I am not good at perceiving myself”). Indeed, *poor perception of the input causes users to provide poor examples*, especially when focusing on fixing a particular problem in the prior and introducing undesired offsets in other dimensions. An example is in Fig.15: the user was fixing a problem with the elbow angle, but introduces a large offset in the shoulder roll with

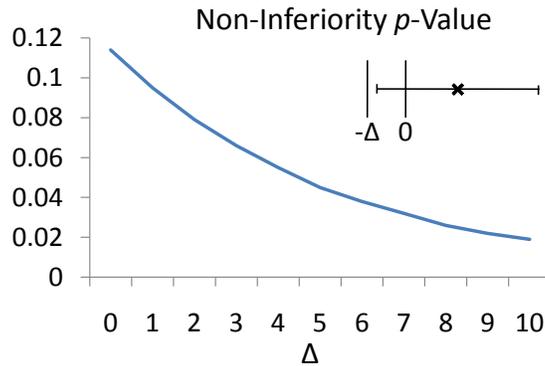


Figure 14.  $p$ -values for non-inferiority tests with different margins. As the margin increases, we are more and more confident that the difference between  $UP - T$  and  $NP$  is greater than the negative margin. The same figure also depicts a typical non-inferiority test, where the lower bound for the 95% confidence interval must be greater than the negative margin. Figure from (Dragan & Srinivasa, 2012c).

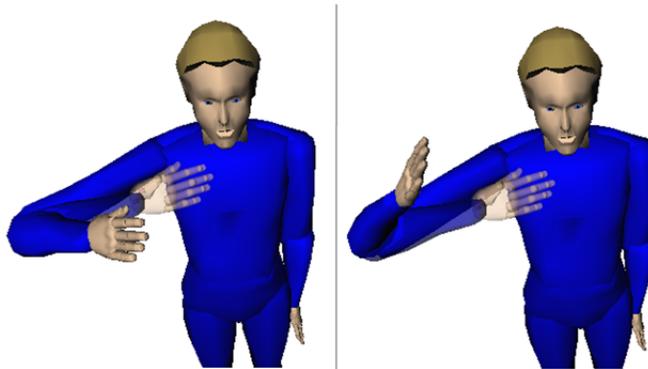


Figure 15. The user's poor self-perception leads to poor examples. This is a focus side-effect: focusing on fixing a particular problem sometimes introduces undesired additional offsets. On the left, the approximate intended example, and on the right the provided example, which adds a large offset to the roll. Figure from (Dragan & Srinivasa, 2012c).

his example.

Proprioception was not the only factor contributing to this user's poor training performance. Fig.16 shows the difference between this user and second user, with better training performance: the latter explores more parts of the space, including configurations close to those necessary for the grasping task that served as a test after training, which puts him at an advantage. We see this as a side effect of not having told users what the task was ahead of time, in order to prevent them from training the priors only for the test task: a user that missed the task-related areas of the space performed worse on the task, *stressing the importance of task-specific retargetting functions*.

In terms of the number of examples, the users did provide significantly more examples to  $UP$ , as expected ( $t(6) = 16.44, p < .001$ ):  $UP$  received an average of 11 examples,

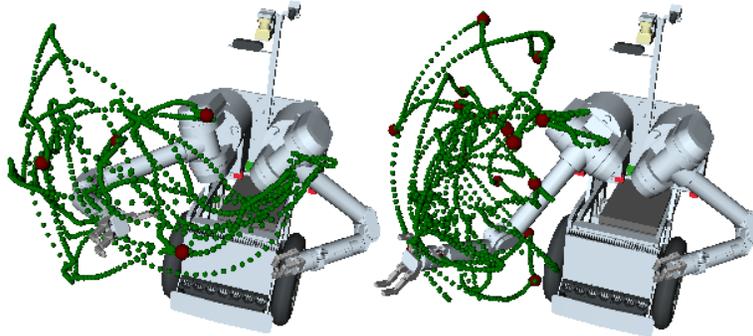


Figure 16. The space two users explored while training. The green curves trace the end effector of the robot. The red points are locations of examples. The robot is positioned in the last example’s configuration. Figure from (Dragan & Srinivasa, 2012c).

with 4 being the minimum and 20 the maximum. Interestingly, not all users provided examples to the natural prior  $NP$ , some deeming it suitable as-is within a couple of minutes of exploring the space. However, there was no significant (or marginal) correlation between the number of examples provided and the performance: more examples did not mean better or worse performance. We did find a significant correlation between the number of examples given to  $NP$  and the one given to  $UP$  (Pearson’s  $r(3) = 0.94, p = .004$ ), suggesting that some users are generally more lenient than others.

To test the last hypothesis, we need to show that the difference in performance between  $UP - T$  and  $NP$  is significantly greater than a negative margin  $-\Delta$ . Setting  $\Delta = 10\%$ , a one-tailed paired  $t$ -test supported the hypothesis:  $9.5 > -10, t(6) = 2.81, p = .019$ . Therefore, *we can be confident that training the bad prior does not decrease the performance, on average, by more than 10%*. Fig.14 provides a graphical explanation of the test, together with the  $p$ -values obtained from various such margins between -10 and 0 (notice that the larger the margin, the more confident we are that  $UP - T$  is not worse by more than that margin). Regarding the non-inferiority of  $UP - T$  w.r.t.  $NP - T$ , the result was similar:  $t(6) = 4.34, p = .004$  for the same  $\Delta = 10\%$ .

#### 5.4 Main Lesson

Our study revealed that while users can improve upon bad priors, *poor examples constitute a real danger to customization*. Users did not improve upon the good prior. This can be partially due to measurement noise: we measured performance on a particular task with a few samples. It can also be due to the task we chose: users were not asked to customize to that particular task. However, it is also possible that this problem was caused by an increased risk of poor examples that good priors might run: when the desired improvement is more subtle, it becomes less obvious what example to give and the risk of unintentionally altering something increases.

In an exploratory experiment, we analyzed what happens when the customization is task-specific, and there are no poor examples. We found that an expert user can successfully improve upon  $NP$  when customizing it with a given task in mind. Although  $NP$  is a natural retargeting function in general, the kinematic differences between a human and HERB make particular tasks (such as grasping an object on the far left of the robot) partic-

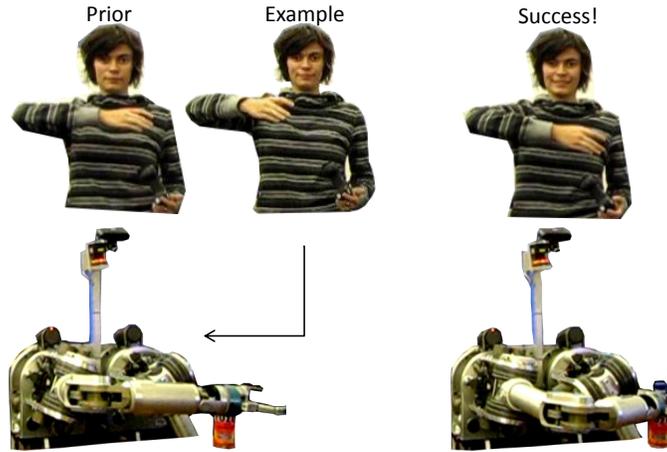


Figure 17. Adapting a good prior to a specific task. Figure from (Dragan & Srinivasa, 2012c).

ularly difficult. An expert example makes the task area easier to reach, as in Fig.17. This is in no way reflective of the performance of a novice user with the system – it is merely a vote of confidence in what the training method enables an experienced user to accomplish. We are therefore interested in pursuing a long-term study to analyze the capabilities of retargetting customization when users become accustomed to the robot and the training method. Our hope is that with experience, users become better at providing the right examples.

## 6. Discussion

**Challenges of Intelligent and Customizable Interfaces.** Our work studied interfaces that make teleoperation more seamless by adding assistive intelligence and by lending their retargetting functions to customization. We believe that our main contribution is not in showing that these ideas can be useful, but in revealing the challenges that these ideas bring about.

Perhaps our most baffling observation, directly stemming from the key challenges our experiments identify, is that getting a robot to perform a task with the user’s help can be, in many ways, more difficult than getting it to perform the same task autonomously. Collaboration, in the form of assistance during teleoperation or even in the larger sense of humans and robots working together, brings about new requirements on robotics and machine learning.

The robot must predict the collaborator’s intent, and it must itself act in an intent-expressive way – a way that enables the collaborator to predict what the robot is doing. Both intent-prediction and intent-expressiveness must be adapted to the collaborator. Different people act or teleoperate according to different strategies (some users reached for an object by moving the robot one joint at a time rather than executing a natural reaching motion). As a result, the robot must adapt its prediction model to the user. Similarly, it is conceivable that what is intent-expressive to one user is confusing to another, in which case the robot must also adapt its planner to the user.

Leaving the general collaboration realm and entering the specifics of assistive intelligence, the robot must decide how to assist depending on the confidence it has in itself

and in the user, as well as the type of user – some users are more accepting of assistance than others. Furthermore, because the user’s behavior is affected by how much the robot is assisting, the intent predictor must also adapt to the level of arbitration. This becomes challenging in particular for high arbitration: if the robot is correct and intent-expressive, the user loses the incentive to provide good input, which affects the prediction. This stresses the importance of asking users to provide intent-expressive input, which we have shown they are capable of, especially if the retargetting function can somewhat mask the kinematics differences.

The retargetting function is important not only for enabling intent prediction, but also in the lack of assistance, when the robot directly executes the retargetted input. Although enabling customization of this function can lead to more efficient teleoperation, the customization process itself has its risks. Customization must be robust to poor examples and even task-driven, to ensure that the outcome works well on the types of tasks the user will encounter.

**Limitations of Our Work.** Although our studies reveal important aspects of these interfaces, our work is limited in many ways. Our studies have a small number of users (usually students), they are performed in a lab setting, which can alter a user’s behavior, and are not diverse in terms of teleoperation devices, robots and tasks. Our variable manipulations were not straight-forward and could have been done differently. We had to decide how hard a hard task should be, and chose it at the limit of the reachability through the teleoperation interface – but there are many alternatives. We had to decide what a good prior retargetting function would be, and we chose what we consider a natural mapping between the human and the robot, taking advantage of the fact that the robot is somewhat anthropomorphic. We had to decide what an incorrect prediction behavior meant, and we chose it such that the robot is stubborn in its incorrectness, but could still be rectified by the user. We had to manipulate the robot kinematics, and had to settle for a simulated robot, leading to a potential confound – could the difference be caused by the fact that teleoperating in simulation is easier than teleoperating the real robot?

We focused on certain factors but missed other potentially important ones. For example, it is likely that healthy users would have different assistance preferences from users with certain disabilities. For our study on training users to provide intent-expressive input, we do not know training is necessary, if at all: was the 2D simplification required? is any training required? what effect does the intuitiveness of the predictor have on the training needed? We used a predictor that is intuitive, and for which intent-expressive motion correlates to what a human would find intent-expressive – but what happens if the prediction mechanism is less transparent? Our interface customization study is missing the effect of expertise with the system: although we expect experienced users to be successful at making good priors better, we never tested this hypothesis in a formal study.

**Concluding Remarks.** Overall, we believe intelligence and customizability of an interface will be very important as we bring robots from factories into homes. Our work opens the way to many exciting avenues of further research on this topic – in HRI, as well as in planning, controls and machine learning.

## 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.

## References

- Aarno, D., Ekvall, S., & Kragic, D. (2005). Adaptive virtual fixtures for machine-assisted teleoperation tasks. In *Ieee icra*.
- Aigner, P., & McCarragher, B. (1997). Human integration into robot control utilising potential fields. In *Icra*.
- Anderson, S. J., Peters, S. C., Iagnemma, K., & Overholt, J. (2010). Semi-autonomous stability control and hazard avoidance for manned and unmanned ground vehicles. In *Mit, dept. of mechanical eng.*
- Argall, B., Chernova, S., Veloso, M., & Browning, B. (2009). A survey of robot learning from demonstration. *Robotics and Autonomous Systems*, 57(5), 469 - 483.
- Atkeson, C. (1991). Using locally weighted regression for robot learning. In *Icra*.
- Baturone, A. O., Rojo, F. C., Alczar, J., Lopez-Pichaco, F., & Nogales, C. (2003, Octubre). *Helicopter teleoperation for aerial monitoring in the comets multi-uav system*.
- Biocca, F. A., & Rolland, J. P. (1998, June). Virtual eyes can rearrange your body: Adaptation to visual displacement in see-through, head-mounted displays. *Presence: Teleoper. Virtual Environ.*, 7(3), 262–277. Available from <http://dx.doi.org/10.1162/105474698565703>
- Cakmak, M., & Thomaz, A. L. (2012). Designing robot learners that ask good questions. In *Proceedings of the international conference on human-robot interaction (hri)*.
- Chang, S., Kim, J., Kim, I., Borm, J., Lee, C., & Park, J. (1999). Kist teleoperation system for humanoid robot. In *IROS*.
- Chen, J. Y. (2010). Uav-guided navigation for ground robot tele-operation in a military reconnaissance environment. *Ergonomics*, 53(8), 940-950. Available from <http://www.tandfonline.com/doi/abs/10.1080/00140139.2010.500404>
- Chernova, S., & Veloso, M. (2009). Interactive policy learning through confidence-based autonomy. *J. Artificial Intelligence Research*, 1–25.
- Cousineau, D. (2005). Confidence intervals in within-subject designs: A simpler solution to Loftus and Masson's method. *Tutorial in Quantitative Methods for Psychology*, 1(1), 42–45.
- Crandall, J., & Goodrich, M. (2002). Characterizing efficiency of human robot interaction: a case study of shared-control teleoperation. In *Iros*.
- Debus, T., Stoll, J., Howe, R., & Dupont, P. (2000). Cooperative human and machine perception in teleoperated assembly. In *Iser*.
- Demiris, Y., & 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., & Srinivasa, S. (2011). Learning to provide better examples for our robots. In *Hri pioneers workshop*.
- Dragan, A., & Srinivasa, S. (2012a). Assistive teleoperation: A new domain for interactive learning. In *Aaai fall symposium on robots interactively learning from human teachers*.
- Dragan, A., & Srinivasa, S. (2012b). Formalizing assistive teleoperation. In *RSS*.
- Dragan, A., & Srinivasa, S. (2012c). Online customization of teleoperation interfaces. In *Ro-man*.
- Ehlert, P. A. M. (2003, feb). *Intelligent user interfaces: introduction and survey*.
- Fagg, A. H., Rosenstein, M., Platt, R., & Grupen, R. A. (2004). Extracting user intent in mixed initiative teleoperator control. In *Aiaa*.
- Fisher, A., McDermott, P. L., & Fagan, S. (2009). Bandwidth allocation in a military teleoperation task. In *Proceedings of the 4th acm/ieee international conference on human robot interaction* (pp. 287–288). New York, NY, USA: ACM. Available from <http://doi.acm.org/10.1145/1514095.1514179>
- Fukuda, O., Tsuji, T., Kaneko, M., & Otsuka, A. (2003). A human-assisting manipulator teleoperated by emg signals and arm motions. *ICRA*.
- Gleicher, M. (1998). Retargetting motion to new characters. In *Siggraph*.

- Goertz, R. (1963). Manipulators used for handling radioactive materials. *Human factors in technology*.
- Grollman, D., & Jenkins, O. (2008). Sparse incremental learning for interactive robot control policy estimation. In *ICRA*.
- Hauser, K. (2012, July). Recognition, prediction, and planning for assisted teleoperation of freeform tasks. In *Proceedings of robotics: Science and systems*. Sydney, Australia.
- Hecker, C., Raabe, B., Enslow, R., DeWeese, J., Maynard, J., & Van Prooijen, K. (2008). Real-time motion retargeting to highly varied user-created morphologies. In *Siggraph*.
- Hochberg, L. (2012). Reach and grasp by people with tetraplegia using a neurally controlled robotic arm. *Nature*, 485, 372.
- Howe, R. D., & Kontarinis, D. A. (1994). High-frequency force information in teleoperated manipulation. In T. Yoshikawa & F. Miyazaki (Eds.), *Experimental robotics iii* (Vol. 200, p. 341-352). Springer Berlin / Heidelberg. Available from <http://dx.doi.org/10.1007/BFb0027606> (10.1007/BFb0027606)
- Hu, C., Meng, M., Liu, P., & Wang, X. (2003, oct.). Visual gesture recognition for human-machine interface of robot teleoperation. In *Intelligent robots and systems, 2003. (iros 2003). proceedings. 2003 ieee/rsj international conference on* (Vol. 2, p. 1560 - 1565 vol.2).
- Ikemoto, L., Arikan, O., & Forsyth, D. (2009). Generalizing motion edits with gaussian processes. *ACM Trans. Graph.*, 28(1), 1:1-1:12.
- Kadous, M. W., Sheh, R. K.-M., & Sammut, C. (2006). Effective user interface design for rescue robotics. In *Proceedings of the 1st acm sigchi/sigart conference on human-robot interaction* (pp. 250-257). New York, NY, USA: ACM. Available from <http://doi.acm.org/10.1145/1121241.1121285>
- Kaiser, M., & Dillmann, R. (1996). Building elementary robot skills from human demonstration. In *Icra*.
- Kawamura, K., Nilas, P., Muguruma, K., Adams, J., & Zhou, C. (2003, jan.). An agent-based architecture for an adaptive human-robot interface. In *System sciences, 2003. proceedings of the 36th annual hawaii international conference on* (p. 8 pp.).
- Kim, D.-H., Kim, K., Kim, K.-Y., & Cha, S.-M. (2001). Dexterous teleoperation for micro parts handling based on haptic/visual interface. In *Micromechatronics and human science, 2001. mhs 2001. proceedings of 2001 international symposium on* (p. 211 -217).
- Kim, D.-J., Hazlett-Knudsen, R., Culver-Godfrey, H., Rucks, G., Cunningham, T., Port ande, D., et al. (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, Part A: Systems and Humans*.
- Kivinen, J., Smola, A., & Williamson, R. (2004). Online learning with kernels. In *Ieee trans. on signal processing*.
- Kofman, J., W., X., Luu, T., & Verma, S. (2005). Teleoperation of a robot manipulator using a vision-based human-robot interface. *IEEE Trans. on Industrial Electronics*, 52(5), 1206 - 1219.
- Labonte, D., Michaud, F., Boissy, P., Corriveau, H., Cloutier, R., & Roux, M. (2006, oct.). A pilot study on teleoperated mobile robots in home environments. In *Intelligent robots and systems, 2006 ieee/rsj international conference on* (p. 4466 -4471).
- Leeper, A. E., Hsiao, K., Ciocarlie, M., Takayama, L., & Gossow, D. (2012). Strategies for human-in-the-loop robotic grasping. In *Hri*.
- Lesaffre, E. (2008). Superiority, equivalence, and non-inferiority trials. *Bull NYU Hosp Jt Dis*, 66(2), 150-154.
- Li, M., & Okamura, A. (2003). Recognition of operator motions for real-time assistance using virtual fixtures. In *Haptics*.
- Marayong, P., Li, M., Okamura, A., & Hager, G. (2003). Spatial motion constraints: theory and demonstrations for robot guidance using virtual fixtures. In *Icra*.

- Marayong, P., Okamura, A. M., & Bettini, A. (2002). Effect of virtual fixture compliance on human-machine cooperative manipulation. In *Iros*.
- Okamura, A. M. (2004, Dec). Methods for haptic feedback in teleoperated robot-assisted surgery. *Ind Rob*, 31(6), 499–508.
- Paravati, G., Sanna, A., Lamberti, F., & Celozzi, C. (2011). A reconfigurable multi-touch framework for teleoperation tasks. In *Emerging technologies and factory automation*.
- Park, C. H., & Howard, A. (2007, 29 2007-nov. 2). Vision-based force guidance for improved human performance in a teleoperative manipulation system. In *Intelligent robots and systems, 2007. iros 2007. ieee/rsj international conference on* (p. 2126 -2131).
- Park, J., Corteso, R., & Khatib, O. (2003). Robust and adaptive teleoperation for compliant motion tasks. In *In proceedings of the international conference on advanced robotics* (pp. 513–519).
- Rockall, T. A., & Darzi, A. W. (2003). Tele-manipulator robots in surgery. *British Journal of Surgery*, 90(6).
- Rosenberg, L. (1993). Virtual fixtures: Perceptual tools for telerobotic manipulation. In *Virtual reality annual international symposium*.
- Schalk, G., Miller, K. J., Anderson, N. R., Wilson, J. A., Smyth, M. D., Ojemann, J. G., et al. (2008). Two-dimensional movement control using electrocorticographic signals in humans. *Journal of Neural Engineering*, 5(1), 75. Available from <http://stacks.iop.org/1741-2552/5/i=1/a=008>
- Scott, I. (2009). Non-inferiority trials: determining whether alternative treatments are good enough. *Med J*, 190, 326-330.
- Shen, J., Ibanez-Guzman, J., Ng, T. C., & Chew, B. S. (2004). A collaborative-shared control system with safe obstacle avoidance capability. In *Ram*.
- Sheridan, T. (1992). *Telerobotics, automation, and human supervisory control*. MIT Press.
- Smith, C., Bratt, M., & Christensen, H. (2008). Teleoperation for a ball-catching task with significant dynamics. *Neural Networks*, 21(4), 604 - 620.
- Smith, C. L., & Kantor, P. B. (2008). User adaptation: Good results from poor systems. In *Laboratory for advanced information research (lair) technical report*.
- Srinivasa, S., Berenson, D., Cakmak, M., Collet, A., Dogar, M., Dragan, A., et al. (2012). Herb 2.0: Lessons learned from developing a mobile manipulator for the home. *Proceedings of the IEEE, Special Issue on Quality of Life Technology*.
- USA Today. (2002, June). Microsoft banks on anti-clippy sentiment. *The Associated Press*. Available from <http://www.usatoday.com/tech/news/2001-05-03-clippy-campaign.htm>
- Waern, A. (1996). *Recognising human plans: Issues for plan recognition in human-computer interaction*.
- Yamane, K., Ariki, Y., & Hodgins, J. (2010). Animating non-humanoid characters with human motion data. *SCA*.
- You, E., & Hauser, K. (2011). Assisted teleoperation strategies for aggressively controlling a robot arm with 2d input. In *R:ss*.
- Yu, W., Alqasemi, R., Dubey, R., & Pernalet, N. (2005). Telemanipulation assistance based on motion intention recognition. In *Icra*.
- Ziebart, B., Ratliff, N., Gallagher, G., Mertz, C., Peterson, K., Bagnell, J., et al. (2009, oct.). Planning-based prediction for pedestrians. In *Intelligent robots and systems, 2009. iros 2009. ieee/rsj international conference on* (p. 3931 -3936).

---

Authors' names and contact information: Anca Dragan and Siddhartha Srinivasa, The Robotics Institute, Carnegie Mellon University, Pittsburgh PA, USA, and Kenton Lee, University of Pennsylvania, Philadelphia PA, USA. Emails: [adragan@cs.cmu.edu](mailto:adragan@cs.cmu.edu), [siddh@cs.cmu.edu](mailto:siddh@cs.cmu.edu) and [kentonl@seas.upenn.edu](mailto:kentonl@seas.upenn.edu).