The International Journal of Robotics Research

http://ijr.sagepub.com/

A policy-blending formalism for shared control

Anca D Dragan and Siddhartha S Srinivasa

The International Journal of Robotics Research 2013 32: 790

DOI: 10.1177/0278364913490324

The International Journal of
Robotics Research
32(7) 790–805

DOI: 10.1177/0278364913490324
ijr.sagepub.com

A policy-blending formalism for shared control

Anca D Dragan and Siddhartha S Srinivasa

Abstract

In shared control teleoperation, the robot assists the user in accomplishing the desired task, making teleoperation easier and more seamless. Rather than simply executing the user's input*, which is hindered by the inadequacies of the interface, the robot attempts to predict the user's* intent*, and assists in accomplishing it. In this work, we are interested in the scientific underpinnings of assistance: we propose an intuitive formalism that captures assistance as* policy blending*, illustrate how some of the existing techniques for shared control instantiate it, and provide a principled analysis of its main components: prediction of user intent and its arbitration with the user input. We define the prediction problem, with foundations in inverse reinforcement learning, discuss simplifying assumptions that make it tractable, and test these on data from users teleoperating a robotic manipulator. We define the arbitration problem from a control-theoretic perspective, and turn our attention to what users consider good arbitration. We conduct a user study that analyzes the effect of different factors on the performance of assistance, indicating that arbitration should be contextual: it should depend on the robot's confidence in itself and in the user, and even the particulars of the user. Based on the study, we discuss challenges and opportunities that a robot sharing the control with the user might face: adaptation to the context and the user, legibility of behavior, and the closed loop between prediction and user behavior.*

Keywords

teleoperation, shared control, sliding autonomy, intent prediction, arbitration, human-robot collaboration

1. Introduction

We focus on the problem of teleoperating dexterous robotic manipulators to perform everyday manipulation tasks (Figure 1). In direct teleoperation, the user realizes their intent, for example grasping the bottle in Figure 1, by controlling the robot via an interface. Direct teleoperation is limited by the inadequacies and noise of the interface, making tasks, especially complex manipulation tasks, often tedious and sometimes impossible to achieve.

In shared control, the robot assists the user in accomplishing the task by attempting to predict their intent and augment their input. Here, the robot faces two challenges when assisting: 1) predicting what the user wants, and 2) deciding how to use this prediction to assist.

We contribute a principled analysis of these two challenges. We first introduce policy blending as one useful interpretation of shared control: a formalism for the robot's assistance as an arbitration of two policies, namely, the user's input and the robot's prediction of the user's intent. At any instant, given the input, U, and the prediction, P, the robot combines them using a state-dependent arbitration function $\alpha \in [0, 1]$ (Figure 1, middle). Policy blending with accurate prediction has a strong corrective effect on the user input (Figure 1, bottom). Of course, the burden is on the robot to predict accurately and arbitrate appropriately.

Prediction. Prior work in this area usually assumes that the robot knows the user's intent (Rosenberg, 1993; Aigner and McCarragher, 1997; Debus et al., 2000; Crandall and Goodrich, 2002; Marayong et al., 2002, 2003; Kofman et al., 2005; Kim et al., 2011; You and Hauser, 2011). Other work assumes that the user is following one of a set of predefined paths or behaviors, and trains a classifier for prediction (Demiris and Hayes, 2002; Li and Okamura, 2003; Fagg et al., 2004; Aarno et al., 2005; Yu et al., 2005). In many real-world scenarios, however, environments and goals change significantly, restricting the utility of fixed paths. For example, in the situation from Figure 1, the user must adapt to various locations of the goal object and its surrounding clutter.

The Robotics Institute, Carnegie Mellon University, USA

Corresponding author:
Anca D Dragan, The Robotics Institute, Carnegie Mellon University, USA.
Email: adragan@cs.cmu.edu

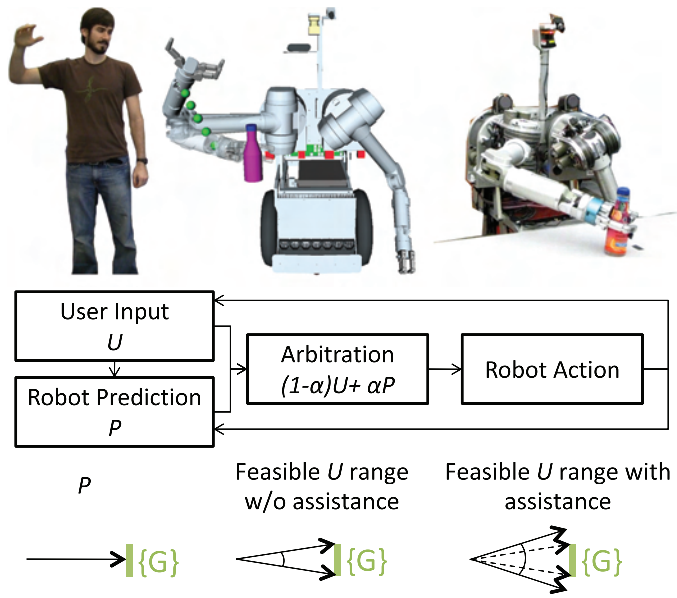


Fig. 1. (Top) The user provides an input U. The robot predicts their intent, and assists them in achieving the task. (Middle) Policy blending arbitrates user input and robot prediction of user intent. (Bottom) Policy blending increases the range of feasible user inputs (here, $\alpha = 0.5$).

In our work, we stray away from predefined paths, and instead formulate the prediction problem based on inverse reinforcement learning (Abbeel and Ng, 2004; Ratliff et al., 2006; Ziebart et al., 2008). We take advantage of the fact that:

> *because there is a user in the loop, prediction does not need to be perfect.*

This leads to the possibility of introducing several approximations that keep the prediction accurate, while making it tractable and even real-time (Section 3.1). We discuss how these approximations perform on real teleoperation data (Section 4.2), and point out directions for improving prediction performance (Section 6).

In some situations, the user might be able to specify the intended goal (including the exact grasp) using other interfaces, like a GUI (Leeper et al., 2012) or speech. However, prediction via motion can be natural, fast, and seamless, enabling a user to, for example, easily change their mind and switch to another grasp or object, and can complement other interfaces.

Arbitration. 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 et al., 2005), to potential fields towards the goal (Aigner and McCarragher, 1997), a great number of such methods can be interpreted as doing some form of arbitration (with some function α from Figure 1) between user input and robot prediction (details in Table 1). This common lens for assistance enables us to analyze some of the factors that affect its performance, and recommend future work areas that have the potential to improve shared control systems.

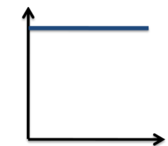
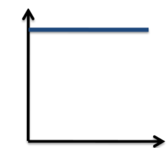
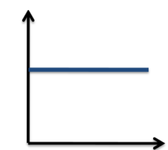
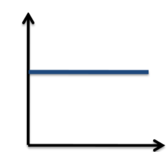
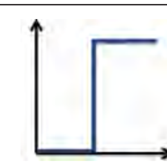
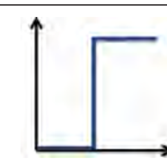
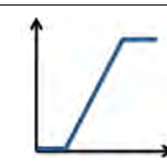
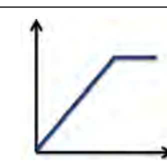
We formulate arbitration in a control-theoretical framework, analyzing the cases in which it can lead to inescapable situations, that get the robot stuck motionless (Section 3.2). Mathematically, arbitration can be any function, with the exception of these few adversarial cases. But to find out what makes an arbitration function good, we turned to real users and studied how they interacted with the assistance (Section 4.1).

Prior work (detailed in Section 2) compared more manual vs more autonomous assistance modes (Marayong et al., 2002; Kim et al., 2011; You and Hauser, 2011) with surprisingly conflicting results in terms of what users prefer: while some studies find autonomy to be better, others advocate no assistance at all. We found that these seemingly contradictory results could potentially be explained by investigating how the arbitration type interacts with other factors (e.g. the type of task). Rather than using autonomy as a factor, we look at aggressiveness: arbitration should be moderated by the robot's confidence in the prediction, leading to a spectrum from very timid to very aggressive assistance, from small augmentation of user input even when confident to large augmentation even when unsure. Rather than analyzing the effect of aggressiveness (or autonomy) alone on the performance of assistance, we conducted a user study that analyzes how aggressiveness interacts with other factors, like prediction correctness and task difficulty, in order to help explain the seemingly contradictory findings above.

Our study suggests that

> *Arbitration should indeed be contextual: it should depend on the robot's confidence in itself and in the user, as well as on the particulars of the user.*

Table 1. Prior work.

Method	Prediction	Arbitration
Rosenberg (1993) Marayong et al. (2003) Debus et al. (2000) You and Hauser (2011) Kim et al. (2011) Marayong et al. (2002) Leeper et al. (2012)	no	
Demiris and Hayes (2002) Fagg et al. (2004)	predefined behaviors	
Crandall and Goodrich (2002) Aigner and McCarragher (1997) Gerdes and Rossetter (2001) You and Hauser (2011) Marayong et al. (2002)	no	
Yu et al. (2005)	predefined behaviors	
Kofman et al. (2005) Shen et al. (2004) Smith et al. (2008)	no	
Li and Okamura (2003)	predefined behaviors	
Anderson et al. (2010) Loizou and Kumar (2007) Weber et al. (2009)	MPC/minimum-jerk	
Aarno et al. (2005) Yu et al. (2005)	predefined behaviors	
Vasquez et al. (2005) Ziebart et al. (2009)	fixed goals (2D) fully flexible (2D)	no no

Challenges and opportunities of shared control. Our formalism and analysis build on machine learning, control theory, and human–robot interaction to provide insight into shared control. We suggest possible challenges, as well as opportunities, that could arise from the tight interaction between the robot and the user: adaptation to the context and the user, predicting and expressing intent, and capitalizing on the user's reactions. These challenges and opportunities are applicable not only to shared control, but conceivably also to human–robot collaboration in general.

2. Prior work

In 1963, Goertz proposed manipulators for handling radioactive material that are able to turn cranks based on imprecise operator inputs, introducing one of the first instances of shared control (Goertz, 1963). 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; Debus et al., 2000; Demiris and Hayes, 2002; Marayong et al., 2002, 2003; Fagg et al., 2004; Kim et al., 2011; You and Hauser, 2011), to taking control (or releasing it) at some trigger (Li and Okamura, 2003; Shen et al., 2004; Kofman et al., 2005; Kragic et al., 2005), to never fully taking control (Aigner and McCarragher, 1997; Crandall and Goodrich, 2002; Marayong et al., 2002; Aarno et al., 2005; You and Hauser, 2011). For example, Debus et al. (2000) proposed 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 and Goodrich (2002) proposed to mix the user input with a potential field in order to avoid obstacles.

The most widely studied paradigm of shared control is that in which the robot shifts between pre-defined discrete levels of autonomy: sliding autonomy (Dias et al., 2008), adjustable autonomy (Kortenkamp et al., 2000), and sliding scale autonomy (Bruemmer et al., 2002). They are typically thought of as the robot performing autonomously, but giving the control to the user in difficult situations, in other words 'incorporate human intervention when needed' (Kortenkamp et al., 2000; Sellner et al., 2005; Dias et al., 2008). More generally, however, the robot can continuously change the level of assistance, much like a more recent definition of sliding scale autonomy (Desai and Yanco, 2005), or semi-autonomy (Anderson et al., 2010).

Attempts to compare different levels of assistance are sometimes contradictory. For example, You and Hauser (2011) found that for a complex motion planning problem in a simulated environment, users preferred a fully autonomous mode, where they only clicked on the desired goal, to more reactive modes of assistance. On the other hand, Kim et al. (2011) found that users preferred a manual mode and not the autonomous one for manipulation tasks like object grasping.

Policy blending provides a common perspective on assistance, leading to an analysis which helps reconcile these differences. Table 1 shows how various methods proposed arbitrate user input and robot prediction (or simply robot policy, in cases where intent is assumed to be known). For example, potential field methods (e.g. Aigner and McCarragher, 1997; Gerdes and Rossetter, 2001; Crandall and Goodrich, 2002; Yu et al., 2005) that help the user avoid obstacles become blends of the user input with a policy obtained from the repulsive force field, under a constant arbitration function that establishes a trade-off. Virtual fixture-based methods (e.g. Marayong et al., 2002; Li and Okamura, 2003; Aarno et al., 2005; Yu et al., 2005) that are commonly used to guide the user along a predefined path become blends of the user input with a policy that projects this input onto the path. The arbitration function dictates the intensity of the fixture at every step, corresponding to a normalized 'stiffness/compliance' gain. However, the same framework also allows for the less studied case in which the robot is able to generate a full policy for completing the task on its own, rather than an attractive/repulsive force or a constraint (e.g. Shen et al., 2004; Kofman et al., 2005). In this case, the arbitration is usually a switch from autonomous to manual, although stages that trade off between the two (not fully taking control but still correcting the user's input) are also possible (Aarno et al., 2005; Anderson et al., 2010). Weber et al. (2009) and Anderson et al. (2010) have both used a linear blend (as in Figure 1) between the user's input and a proposed policy computed via model predictive control or via a minimum-jerk model of intent, which could be interpreted as the robot's predicted policy. Both suggest a threshold-like function to implement the blend, leading to a dynamic switch of the control from (mostly) the user to (mostly) the robot, for vehicle control or for contacting stiff objects. Blending human and robot policies via a generic function has also been proposed by Enes and Book (2010). Desai and Yanco (2005) proposed a linear blend in two dimensions between maximum user speed and maximum robot speed. Outside the teleoperation domain, a type of arbitration (averaging) is used for mediating between two human input channels (Glynn and Henning, 2000), and blending in general between two human policies is used in surgery training (Nudehi et al., 2005). Therefore, an intuitive formalism (linear policy blending) can act as a common lens across a wide range of literature, enabling us to identify prediction and arbitration as two key challenges that a shared-control robot faces.

Analyzing assistance based on how arbitration is done, together with other factors like prediction correctness and task difficulty, provides one way of explaining seemingly contradictory previous findings: our data suggest that aggressive assistance is preferable on hard tasks, like the ones from You and Hauser (2011), where autonomy is significantly more efficient; opinions are split on easier tasks, like the ones from Kim et al. (2011), where the autonomous and manual modes were comparable in terms of time to completion.

The same table shows how prior methods handle prediction of the user's intent: the equivalent of plan recognition in the area of intelligent user interfaces (Waern, 1996). Aside from work that classifies which one of a predefined set of paths, behaviors, or behavior types the user is currently engaging (Demiris and Hayes, 2002; Fagg et al., 2004; Hauser, 2012), most work assumes the robot has access to the user's intent, for example that it knows what object to grasp and how. Exceptions are minimum-jerk-based models, for example Weber et al. (2009). Similarly, Smith et al. (2008) deals with time delays in ball catching by projecting the input forward in time using a minimum-jerk model. Predicting or recognizing intent has received a lot of attention outside of the teleoperation domain, dating back to high-level plan recognition (Schmidt and D'Addamio, 1973). Predicting intended motion, however, is usually again limited to classifying behaviors, or is done in low-dimensional spaces (Zollner et al., 2002; Vasquez et al., 2005; Ziebart et al., 2009). In the following section, which presents the building blocks of assistance, we present the general prediction problem, along with simplifying assumptions that make it tractable.

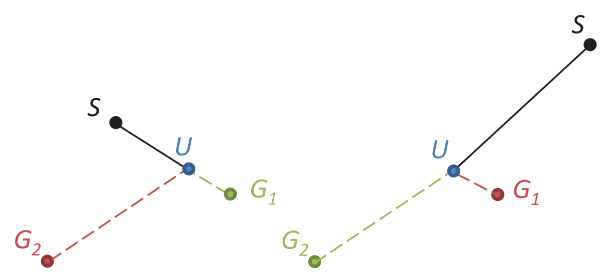


Fig. 2. Even though G_1 is the closest goal to U in both cases, $\xi_{S\to U}$ indicates that G_2 is more likely in the situation on the right.

3. The components of assistance

In what follows, Q denotes the robot's current configuration, U denotes the desired next configuration obtained via the interface that maps the user's direct input through a device onto the robot (e.g. through a GUI, joystick, or a whole-body teleoperation interface, as in Figure 1, or by sending a velocity command to the robot),[1] and P denotes the configuration the robot predicts it should be at next. We denote the user's starting input as S, and the trajectory of user inputs until U as $\xi_{S\to U}$.

Each new scene has a (possibly continuous) set of accessible goals $\mathcal{G}$, known only at runtime to both robot and user. The robot does not know which goal the user is trying to reach, nor does it have predefined trajectories to each of these goals: the starting and goal configurations can differ from every other situation the robot has ever encountered.

3.1. Prediction

The robot must predict where the user would like it to move next, given $\xi_{S\to U}$ and any other cues, for example each goal's reachability, or a high-level description of the overall task.

We break this problem down into two successive steps (see Figure 5):

1. **Goal prediction** where we predict the most likely goal G^* given available data.
2. **Trajectory prediction** where we predict how the user would want to move towards a predicted goal.

Goal prediction. We formulate goal prediction as:

$$G^* = \arg\max_{G\in\mathcal{G}} P(G|\xi_{S\to U}, \theta) \tag{1}$$

in other words, given $\xi_{S\to U}$ and any other available cues θ, the robot predicts the goal G^* that maximizes posterior probability.

Several simplifying assumptions help us to solve this problem. The strongest is amnesic prediction, which ignores all information except the current input U: $G^* = \arg\max_{G\in\mathcal{G}} P(G|U)$. There are many ways to estimate $P(G|U)$. For example, given a distance metric on goals d, we can assume that closer goals have higher probability:

$$G^* = \arg\min_{G\in\mathcal{G}} d(U, G) \tag{2}$$

Under the Euclidean metric, $d(U,G) = ||U - G||$, the method predicts the goal closest in the robot's configuration space. Under $d(U,G) = ||\phi(U) - \phi(G)||$ (with ϕ denoting the forward kinematics function), the method predicts the goal closest in the robot's workspace.

Although intuitive, amnesic prediction does suffer from its amnesia. Where the user came from is often a critical cue for where they want to go. Prediction can also be memory-based, taking into account the trajectory $\xi_{S\to U}$ of user inputs (Figure 2):[2]

$$G^* = \arg\max_{G\in\mathcal{G}} P(G|\xi_{S\to U}) = \arg\max_{G\in\mathcal{G}} P(\xi_{S\to U}|G)\,P(G) \tag{3}$$

In order to compute $P(\xi_{S\to U}|G)$, we need a model of how users teleoperate the robot to get to a goal. A possible assumption is that the user's input noisily optimizes a goal-dependent cost function C_G (one that depends, as an example, on the dot product between the user's velocity and the direction to the goal). Using the principle of maximum entropy, we can use C_G to induce a probability distribution over trajectories $\xi \in \Xi$ given a goal as $P(\xi|G) \propto \exp\big(-C_G(\xi)\big)$, in other words the probability of a trajectory decreases exponentially as its cost increases (Ziebart et al., 2008). Given this distribution and if the cost is additive along the trajectory,

$$P(\xi_{S\to U}|G) = \frac{\exp\big(-C_G(\xi_{S\to U})\big)\int_{\xi_{U\to G}} \exp\big(-C_G(\xi_{U\to G})\big)}{\int_{\xi_{S\to G}} \exp\big(-C_G(\xi_{S\to G})\big)} \tag{4}$$

In low-dimensional spaces, (4) can be evaluated exactly through soft-maximum value iteration (Ziebart et al., 2009). This is too expensive, however, in the high-dimensional spaces that manipulation tasks induce.

Fortunately, this prediction is used with a human in the loop. The direct consequence is that prediction does not need to be perfect: the robot can wait a while longer and get more information from the user before starting to assist.

It is this consequence that enables us to find approximations that perform well in the context of shared autonomy, and are tractable and even real-time.

We approximate the integral over trajectories using Laplace's method. First, we approximate $C(\xi_{X\to Y})$ by its second-order Taylor series expansion around $\xi^*_{X\to Y} = \arg\min_{\xi_{X\to Y}} C(\xi_{X\to Y})$:

$$C(\xi_{X\to Y}) \approx C(\xi^*_{X\to Y}) + \nabla C(\xi^*_{X\to Y})^T(\xi_{X\to Y} - \xi^*_{X\to Y}) + \frac{1}{2}(\xi_{X\to Y} - \xi^*_{X\to Y})^T \nabla^2 C(\xi^*_{X\to Y})(\xi_{X\to Y} - \xi^*_{X\to Y}) \quad (5)$$

Since $\nabla C(\xi^*_{X\to Y}) = 0$ at the optimum, we get

$$\int_{\xi_{X\to Y}} \exp\big(-C(\xi_{X\to Y})\big) \approx \exp\big(-C(\xi^*_{X\to Y})\big) \int_{\xi_{X\to Y}} \exp\Big(-\frac{1}{2}(\xi_{X\to Y} - \xi^*_{X\to Y})^T H_{X\to Y}(\xi_{X\to Y} - \xi^*_{X\to Y})\Big) \quad (6)$$

with $H_{X\to Y}$ the Hessian of the cost function around $\xi^*_{X\to Y}$. Evaluating the Gaussian integral leads to

$$\int_{\xi_{X\to Y}} \exp\big(-C(\xi_{X\to Y})\big) \approx \exp\big(-C(\xi^*_{X\to Y})\big) \frac{\sqrt{2\pi^k}}{\sqrt{|H_{X\to Y}|}} \quad (7)$$

and the optimal prediction G^* becomes

$$\arg\max_G \frac{\exp\big(-C_G(\xi_{S\to U}) - C_G(\xi^*_{U\to G})\big)\sqrt{|H_{U\to G}|}}{\exp\big(-C_G(\xi^*_{S\to G})\big)\sqrt{|H_{S\to G}|}} P(G) \quad (8)$$

If the cost function is quadratic, the Hessian is constant and (8) simplifies to

$$G^* = \arg\max_{G\in\mathcal{G}} \frac{\exp\big(-C_G(\xi_{S\to U}) - C_G(\xi^*_{U\to G})\big)}{\exp\big(-C_G(\xi^*_{S\to G})\big)} P(G) \quad (9)$$

This 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.

In low-dimensional spaces, it is possible to learn C_G from user teleoperation data via maximum entropy inverse reinforcement learning (Ziebart et al., 2008) (or via algorithms that assume no user noise, e.g. Abbeel and Ng, 2004; Ratliff et al., 2006). In larger domains such as manipulation, simple guesses for C_G can still be useful for prediction. This is illustrated in Figure 3, which shows a toy problem with two possible goals and a user trajectory $\xi_{S\to U}$. Even with a very simple C_G, the sum of squared velocity magnitudes, this method still holds an advantage over the amnesic version. When C_G is the same for all goals, the cost of $\xi_{S\to U}$ is common across goals and does not affect the probability, leaving only the starting point S and the current input U as the crucial components of the trajectory so far. The comparison between G_1 and G_2 leads to the correct result, because the path to G_2 through U is much shorter relative to the optimal, $\xi^*_{S\to G_2}$, than the path to G_1 through U is, relative to $\xi^*_{S\to G_1}$. Whether this very simplified memory-based prediction is still helpful in an analogous problem in the real world is one of the questions we answer in our user study.

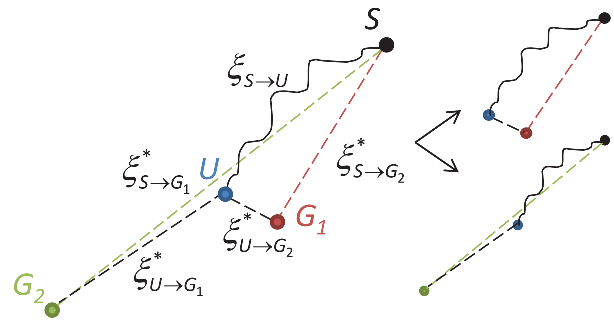


Fig. 3. An example user trajectory leading to two possible goals. A simple prediction implementing (9) with a cost function based on length compares the two situations on the right and determines that the user's trajectory is a smaller deviation from the optimal trajectory to G_2 than to G_1.

A more sophisticated prediction method learns to choose G^* by also considering θ, and using the goal probabilities as features. From labeled data of the form $F[\xi_{S\to U}, \theta, G] \to 0/1$ (features computed relative to G, paired with whether G was the desired goal) a range of methods will learn to predict a goal 'score' given feature values for a new situation. See Dragan et al. (2011a) for goal prediction based on such feature constructs.

Trajectory prediction. Once the robot predicts a goal G^*, it must also compute how the user wants it to move towards G from the current state Q. It can do so by computing a policy or a trajectory based on a cost function. Note that the cost function the robot must optimize in this stage is not (necessarily) the same as the cost function the human is optimizing during teleoperation: the idea behind sharing the autonomy is that the robot executes what the user actually wants rather than what the user commands. Approximations for this function also range from very simple (e.g. the length of the trajectory, or a trade-off between the length and the distance from obstacles) to very complex (e.g. functions learned via inverse optimal control from users physically moving the robot on a trajectory they would want the robot to take).

Given such a cost, the robot would use a trajectory optimizer (Todorov and Li, 2005; Ratliff et al., 2009; Toussaint, 2009; Dragan et al., 2011b; Kalakrishnan et al., 2011) to acquire the intended trajectory (Figure 4). Although trajectory optimizers are known to struggle with high-cost local minima in the complex spaces induced by manipulation tasks, this issue can be alleviated to a certain extent by learning to place the optimizer in good basins of attraction

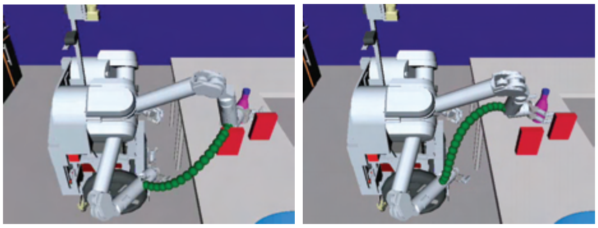


Fig. 4. The trajectory optimizer goal set CHOMP smoothly bends a seed trajectory out of collision.

from prior experience (Jetchev and Toussaint, 2009; Dragan et al., 2011a; Dey et al., 2012).

3.2. Arbitration

Given U and P, the robot must decide on what to do next. The arbitration function, α, is in charge of this decision.

In general, when arbitrating between two policies, we need to guarantee that inescapable local minima do not occur.

Inescapable local minima do not occur. In our case, these are states at which the arbitration results in the same state as at the previous time-step, regardless of the user input.

Theorem 1. *Let Q be the current robot configuration. Denote the prediction velocity as $p = P - Q$, and the user input velocity as $u = U - Q$. Arbitration never leads to inescapable local minima, unless $\forall u \neq 0$, $p = -ku$ for some $k \geq 0$, and $\alpha = 1/(k+1)$ (i.e. the policy is always chosen to directly oppose the user's input, and the arbitration is computed adversarially, or $p = 0$ and $\alpha = 1$ for all user inputs).*

Proof. Assume that at time t, a local minimum occurs in the arbitration, in other words $(1-\alpha)(Q+u)+\alpha(Q+p) = Q$. Further assume that this minimum is inescapable, in other words $(1-\alpha')(Q+u')+\alpha'(Q+p') = Q$, $\forall u'$, where p' and α' are the corresponding prediction and arbitration if u' is the next user input $\Leftrightarrow (1-\alpha')u' + \alpha' p' = 0$, $\forall u'$.

Case 1: $\forall u' \neq 0$, the corresponding $\alpha' \neq 0 \Rightarrow p' = -(1-\alpha/\alpha)u'$, $\forall u' \neq 0 \Rightarrow p' = -ku'$ and $\alpha = 1/(k+1)$, with $k \geq 0$ (since $\alpha \in [0,1]$) $\forall u' \neq 0$. Contradiction with the problem statement.

Case 2: $\exists u' \neq 0$ such that the corresponding $\alpha' = 0 \Rightarrow (1-0)u' + 0p' = 0 \Rightarrow u' = 0$. Contradiction with $u' \neq 0$.

$$\Rightarrow \exists u' \text{ such that } (1-\alpha')(Q+u')+\alpha'(Q+p') \neq Q. \quad \square$$

Therefore, with an adversarial exception, the user can always take a next action that escapes a local minimum.

Monotonicity with confidence. Mathematically, α can be any function, and it 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. But what should this function be? We propose a simple principle: that arbitration must be moderated (among others) by how good the prediction is. Furthermore, we believe that arbitration should increase monotonically with confidence: the higher the confidence, the higher the arbitration value. This is an intuitive design guideline, linking how much the robot assists with how much ground it has for assuming it is assisting in the right way.

Timid vs aggressive. The monotonicity of arbitration leads to a spectrum defined by the trade-off between not over-assisting (providing unwanted assistance) and not under-assisting (failing to provide needed assistance). 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 robot does not trust the predicted policy.

Evaluating confidence. Earlier, we had proposed that the arbitration should take into account how good the prediction is, in other words a measure of the confidence in the prediction, *conf*, that correlates to prediction correctness. A simple definition of *conf* might be as a hinge-loss, where we assume that the closer the predicted goal gets, the more likely it becomes that it is the correct goal:

$$conf = \max\left(0, 1 - \frac{d}{D}\right)$$

with d the distance to the goal and D some threshold past which the confidence is 0. We select this particular measure in our study in Section 4 particularly because of its simplicity, enabling us to design experimental conditions with predictable outcomes.

A more sophisticated definition, however, is the probability assigned to the prediction:

$$conf = P(G^*|\xi_{S\to U})$$

It can also relate to the entropy of the probability distribution (higher entropy meaning lower confidence),

$$conf = \sum_{G\in\mathcal{G}} P(G|\xi_{S\to U}) \log P(G|\xi_{S\to U})$$

or to the difference between the probability of the predicted goal and that of the next most probable candidate:

$$conf = P(G^*|\xi_{S\to U}) - \max_{G\in\mathcal{G}\setminus G^*} P(G|\xi_{S\to U})$$

If a cost function is assumed, the match between the user's input and this cost should also factor in. If a classifier is used for prediction, then such a probability is obtained through calibration (Platt, 1999).

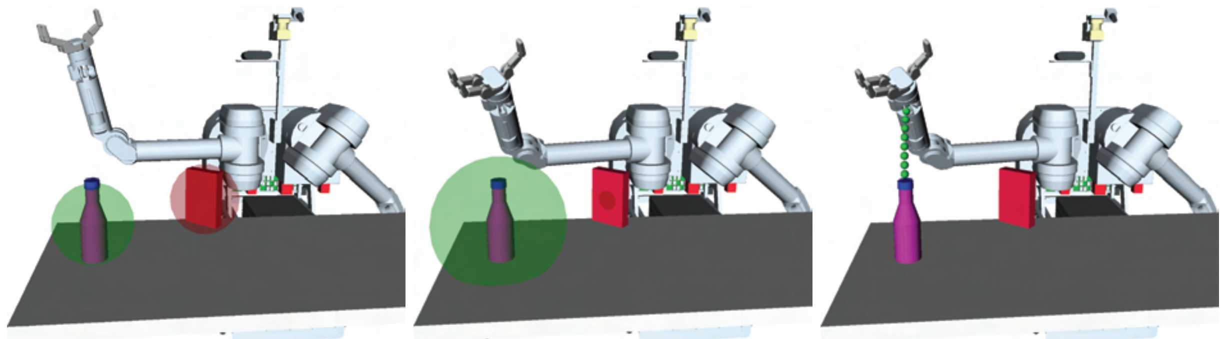


Fig. 5. Left and center: goal prediction. Right: trajectory prediction.

In certain situations, the robot could also shift its confidence based on the trajectory prediction outcome. For example, if it has a probabilistic model of the occupancy grid of the scene, then it could compute a collision probability along the trajectory and use this as part of its confidence computation.

4. A study on assistance

The monotonicity and local minima requirements leave a large set of potential arbitration functions possible, from very timid to very aggressive. But sharing control is fundamentally a human–robot interaction task, and this interaction imposes additional requirements on arbitration: the robot must arbitrate in an efficient and user-preferred way. Therefore, we embarked upon a user study that analyzed the effect of the aggressiveness of arbitration on the performance of assistance.

Although previous work has not analyzed aggressiveness as a factor, there are studies on more manual vs more autonomous assistance modes. Some studies find that users prefer more autonomous assistance (You and Hauser, 2011), which is justified by the fact that autonomy makes the task easier. Other studies found that users prefer to teleoperate the robot manually, with no assistance (Kim et al., 2011), which is justified by users preferring to remain in control of the robot. We believe this apparent contradiction happens because of the contextual nature of good arbitration, in other words because of the interaction of aggressiveness or autonomy level with other factors, like prediction correctness (users might not appreciate assistance if the robot is wrong) and task difficulty (users might appreciate assistance if the task is very hard for them).

Although this analysis of arbitration is our primary goal, we will also test the performance of the simplifying assumptions from Section 3.1 on real data of users teleoperating the robot through our whole-body interface.

We tasked eight users with teleoperating the robot to grasp an object from a table, as in Figure 1. There were always two graspable objects, and we gave the user, for every trial, the farther of the two as goal (an analogous situation to the one from Figure 3). We implemented a whole-body interface that tracks their skeleton (OpenNI, 2010), and directly maps the extracted user joint values onto the robot (which is fairly anthropomorphic), yielding an arm configuration which serves as the user input U (more details in Dragan and Srinivasa, 2012). The robot makes a 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 α. We committed to simple parameters of our formalism in order to explicitly manipulate the factors we were interested in without introducing confounds.

4.1. Goal 1: Factors that affect assistance

Hypotheses.

> **Hypothesis 1**. Aggressiveness, task difficulty and prediction correctness affect the performance of assistance.

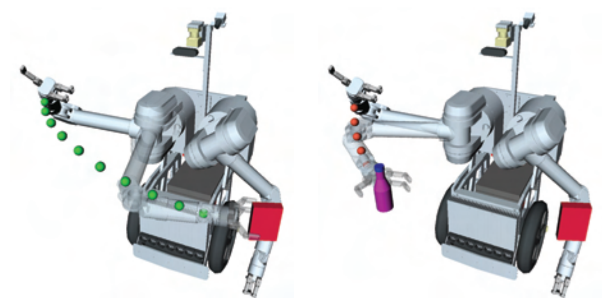
However, particularly for aggressiveness, we expect to find that the choice of an arbitration mode should depend on the context:

> **Hypothesis 2**. Aggressive assistance has better performance on hard tasks if the robot is right, while the timid assistance has better performance on easy tasks if the robot is wrong.

We leave it to exploratory analysis to find what happens with both complications (hard and wrong), or with neither (easy and right).

Manipulated variables. We manipulated prediction correctness by using a simple, easy-to-manipulate goal prediction method: the amnesic prediction based on workspace distance, which always selects the closest object. We set up wrong conditions at the limit of the robot being wrong yet rectifiable. We place the intended object further, guaranteeing wrong prediction until the user makes his preference clear by providing an input U closer to the correct goal. We set up 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 Figure 6(b)



(a) Hard&Right (b) Hard&Wrong

Fig. 6. Hard tasks from our study, with the goal at the limit of the reachable area, for both right and wrong prediction. On the left, the robot correctly predicts the box as the target object (indicated by the green trajectory). On the right, the robot incorrectly predicts the bottle as the target object (indicated by the red trajectory).

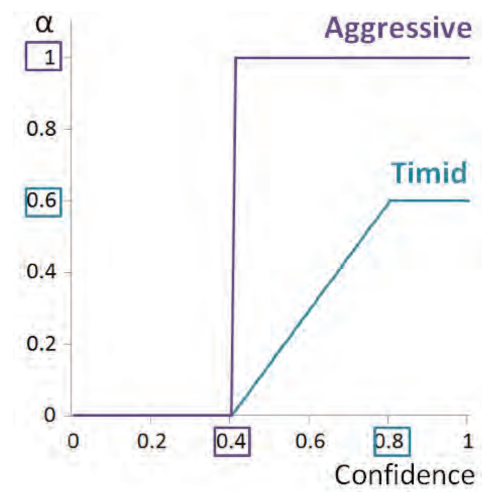


Fig. 7. The timid and aggressive modes as a function of confidence in the prediction.

makes an easy task) vs a location at the limit of the interface's reachability (e.g. grasping the box in Figure 6(b) is a hard task, because it is just outside of the reachable area through the interface that we use, in other words not even an expert user would be able to achieve the task without any assistance). 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 the distance-based measure of confidence: $c = \max(0, 1 - d/D)$, with d the distance to the goal and D some threshold past which the confidence is 0. As the user makes progress towards the predicted object, the confidence increases. We had two assistance modes, shown in Figure 7: 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. We chose these values such that a) the timid mode provides as little assistance as possible while still enabling the completion of the hard task, and b) the aggressive mode provides as much assistance as possible while still being correctible by the user when wrong.

Subject allocation. We chose a within-subjects design, enabling us to ask users to compare the timid and aggressive modes on each task. We recruited eight participants (all students, four males and four females), all with prior exposure to robots or video frames, but none with prior exposure to our system. Each participant executed both modes on each of the four types of tasks. 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 measured 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 seven 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 factors that might also have an effect on performance: the users' initial teleoperation skill, their rating of the robot without assistance, and the learning effect. To control for these, users went though a training phase, teleoperating the robot without assistance. The users were asked to perform a pick-up task (different than the two tasks in our conditions) via direct teleoperation for a total of 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 the number of tasks completed at any point: a measure of prior practice.

4.2. Goal 2: Prediction based on real teleoperation data

Hypothesis.

> On tasks in which the target object is not the closest one to the original human input configuration, replicating the situation from Figure 3, the memory-based prediction will identify the correct goal faster, yielding a higher success rate despite the simplifying assumptions it makes.

Manipulated variables. We used the amnesic prediction during the study for its transparency, which enabled us to manipulate prediction correctness. We compared amnesic vs memory-based prediction on the same data of the users teleoperating the robot under the different conditions, in a post-experimental stage. For memory-based prediction,

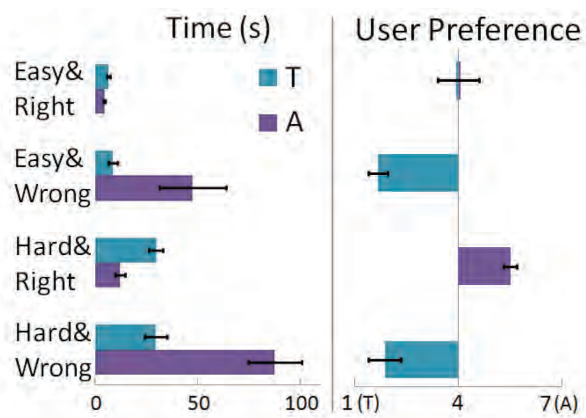


Fig. 8. The results from our study (error bars indicate standard error).

we used workspace sum squared velocity as the cost C, $C = \sum_i ||\xi(i) - \xi(i-1)||^2$, leading to the simplification from (9).

Dependent measures. We took each user trajectory and applied each of the two prediction methods at every point along the trajectory. We measured the percentage of time (success rate) across the trajectory that the prediction identified the correct target object.

Note that this study uses the data post-experiment, implying that differences in predictions could have caused differences in user behavior that are not captured in this data. Nonetheless, they represent real user trajectories with real noise, along which a comparison of predictors provides valuable information on how well these can tolerate imperfect, real-world data.

5. Analysis

Our first goal with this study was to identify the effect of different factors on the performance of assistance, and we do so in the following sections. Our secondary goal was to analyze two simplistic prediction methods (an amnesic and a memory-based one) on teleoperation data under different assistance modes. We discuss our findings in Section 5.2.

5.1. Arbitration

Teleoperation timing. The average time per task was approximately 28 s. We performed a factorial repeated-measures ANOVA with Bonferroni corrections for multiple comparisons and a significance threshold of $p = 0.05$, which resulted in a good fit of the data ($R^2 = 0.66$). All three factors had significant main effects: hard tasks took 22.9 s longer than easy ones ($F(1, 53) = 18.45$, $p < .001$), tasks where the policy was wrong took 30.1 s longer than when right ($F(1, 53) = 31.88$, $p < .001$), and the aggressive mode took 19.4 s longer overall than the timid one ($F(1, 53) = 13.2$, $p = .001$).

Interpreting these main effects as such, however, would be misleading, due to disordinal interaction effects. For example, it is not true that the timid mode is better overall, the interaction suggesting that some situations benefit from the timid mode more while others benefit from the aggressive one. We found one significant interaction effect between aggressiveness and correctness ($F(1, 53) = 36.28$, $p < .001$), and the post-hoc analysis indicated that when wrong, being timid is significantly better than being aggressive ($t(15) = 7.09$, $p < .001$) (in fact, being aggressive and wrong is significantly worse than all three other situations).

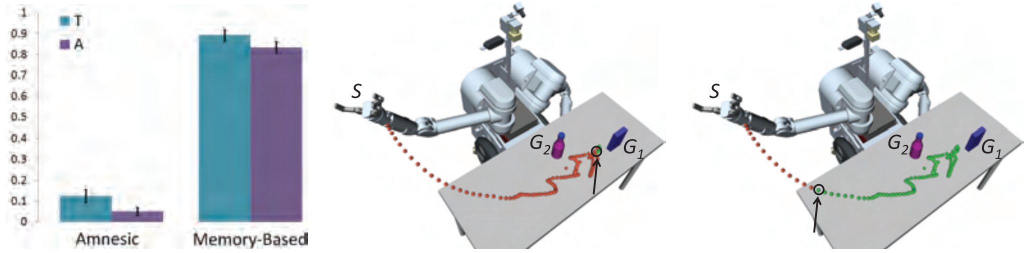
Surprisingly, the three-way interaction was only marginally significant ($F(1, 53) = 2.63$, $p = .11$). To test H2 specifically, we conducted two planned comparisons: one on Hard&Right, for which we found support for the aggressive mode ($t(7) = 4.92$, $p = .002$), and one on Easy&Wrong, for which we found support for the timid mode ($t(7) = 4.95$, $p = .002$).

Figure 8 compares the means and standard errors of each task. The timid mode is better on both Easy&Wrong, as expected, and on 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).

Overall, the effect sizes are influenced by our manipulation of the factors: we see that the hard task did not make it as hard for the timid mode as the wrong prediction made it for the aggressive mode. Because of this, and because of the limited number of subjects, all we conclude from our data is that there are strong trends indicating that aggressiveness is useful when right, especially when the task is difficult, but harmful to the teleoperation process when wrong.

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. Figure 8 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 it taking a slightly longer time, saying 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 timid 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



(a) Percent of Time Prediction is Successful (b) Amnesic Prediction (c) Memory-Based Prediction

Fig. 9. A comparison of amnesic vs memory-based prediction on data from our study. The two examples (b) and (c) indicate the predictions for the same trajectory using the two algorithms: green trajectory points indicate that the prediction using the trajectory up to that point is correct, red points indicate that it is incorrect. The arrows identify the first point at which the prediction becomes correct, happening much earlier in the trajectory for the memory-based algorithm.

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 should not just be quick: some users also want it to be intent-transparent. Our users commented that 'assistance is good if you can tell that [the robot] is doing the right thing'.

5.2. Prediction

Results from the study. A factorial ANOVA on the four manipulated factors obtained a clean fit of the data, with $R^2 = 0.9$. In line with our hypothesis, the factor corresponding to the prediction was significant ($F(1,112) = 1020.95$, $p < .001$): memory-based prediction was significantly better at identifying the correct goal for a longer amount of time along the trajectory. The assistance being timid also made it significantly easier for the prediction method to output the correct answer ($F(1,112) = 7.62$, $p < .01$): users are not as precise about the motion in aggressive mode, making prediction more difficult. Figure 9(a) shows the means and standard errors for the memory-based prediction vs the amnesic one, for each assistance type. Figures 9(b) and 9(c) compare the two methods on one of the trajectories, colored in each case according to whether the prediction was correct (green spheres) or incorrect (red spheres). While the amnesic variant only switches to the correct prediction at the very end, when the user input gets closer to the correct goal, the memory-based prediction is able to identify the goal much earlier.

In our study, the target object was always the farther one (to manipulate prediction correctness, in other words create situations in which the robot is wrong until the user's input gets closer to the correct goal). However, this is disadvantageous to the amnesic method: if the target object were the closer one, this method would usually be 100% accurate. In fact, the amnesic success should be on average almost 0.5 higher. Nevertheless, even with this boost, memory-based remains the out-performer. Furthermore, prediction becomes more challenging as the number of goals increases, changing the results above.

An exploratory experiment. Despite the good performance of the memory-based prediction in the environment configurations from our experiment, we were also interested in exploring its limitations. We conducted a thought experiment by taking real environments and their trajectories from the users, and varying the location of the non-target object. This exposed a few failure modes of the workspace length-based prediction.

First, if the goals are collinear with the start, and the user's trajectory deviates from this line, the method is biased towards the farther goal. We call this the 'null space effect' of using the same C for each goal: given two goals G_1 and G_2 such that the optimal trajectory to G_2 passes through G_1, the two goals get equal probabilities until G_1 along the optimal path (below, Z is the normalization term):

$$P(G_1|\xi^*_{S\to U}) = \frac{1}{Z}\frac{\exp\left(-C(\xi^*_{S\to U}) - C(\xi^*_{U\to G_1})\right)}{\exp\left(-C(\xi^*_{S\to G_1})\right)} = \frac{1}{Z}$$

$$P(G_2|\xi^*_{S\to U}) = \frac{1}{Z}\frac{\exp\left(-C(\xi^*_{S\to U}) - C(\xi^*_{U\to G_2})\right)}{\exp\left(-C(\xi^*_{S\to G_2})\right)} = \frac{1}{Z}$$

When a small deviation arises, it decreases $P(G_1|\dots)$ more, and the method predicts G_2. Even humans are often confused in such situations, for example when interacting in a crowded room. We move towards someone at the back of the room in order to speak with them but are often intercepted along the way by others who predict that social motion incorrectly.

Second, given the trajectories in our data set, the method is in some cases incorrectly biased towards the rightmost object. An example is shown in Figure 10: the trajectory, collected in the environment from Figure 9, heads towards

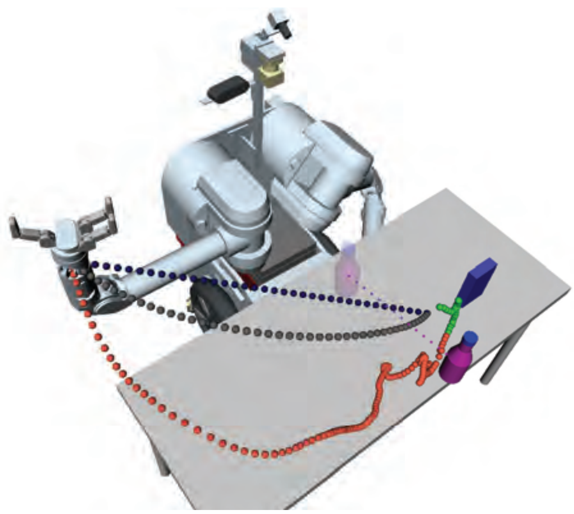


Fig. 10. The user trajectory colored according to the prediction for this artificially generated environment, the trajectory optimizing configuration space length in gray, and the one optimizing work space length in dark blue.

the artificially added obstacle and confuses the predictor. This example should be taken with a grain of salt, because the user would adapt the trajectory to the new environment. Furthermore, the issue could be addressed with priors that prefer closer goals in cases where the goals are 'aligned'. However, the more fundamental problem is the incorrect model of human behavior, as shown in Figure 10.

6. Future work directions

Our study suggests future work directions for robots that share control with their users: adaptation to a user (for both prediction and arbitration), legibility of behavior, and the strong coupling of robot prediction and user action.

6.1. Adaptation

Adapting prediction to the user. In our experiments, we observed different users adopting uniquely different and often surprising motion strategies. While some users took the direct path to the goal, others made progress one dimension at a time, as in Figure 11. Some went above obstacles, while others went around. The existence of these stark differences in strategy implies that it would be beneficial for the quality of the intent predictions if predictors could adjust their model to specific users. Thus, a first research direction is online learning from the interaction with users and adaptation to their behavior.

We can formalize this adaptation as an online update to the cost function C that we assume the user is optimizing (Ratliff et al., 2006), made challenging by the high-dimensionality of the space.

Adapting prediction to the arbitration type. In our experiments, we also observed that users teleoperate differently for different arbitration types. For example, during aggressive arbitration and when the robot was correct in its prediction, some users became more careless in their motion, with their trajectory deviating significantly from one that reaches the desired object (Figure 12) and stopping early on to let the robot finish the task. Alternatively, users can overly compensate for an aggressive mode that only responds to large deviations. This suggests an additional burden on the predictor, of adapting to the arbitration type: the cost function C for a more timid mode is not the same as the one for a more aggressive mode.

Adapting the arbitration function. Our study indicated that the arbitration function should adapt to at least three factors: the robot's confidence in itself, the robot's confidence in the user, and the user type (or their personal preferences).

6.2. Legibility

At every time-step, the robot makes a goal prediction. This alone is not sufficient for assisting the user: the robot must also compute a predicted next action P that the user would like to take towards the goal. In Section 3, we discussed trajectory prediction as a way to compute P: what trajectory would the user want to take toward the goal? We suggested that the robot would optimize a cost function capturing how the user intends it to move to achieve a goal.

Our user comments tend to suggest, however, that this might not be enough, for example 'assistance is good if you can tell that it [the robot] is doing the right thing'. In some cases, the robot's planned trajectory might have the additional burden of making its intent clear to the user, in other words being legible. We believe that if the robot's motion disambiguates between the possible goals in the scene, making it very clear that it predicted the correct intention, then this would have a positive effect on acceptance.

Optimizing for legibility. Legibility depends on the user's inference of the goal based on a snippet of the trajectory that the robot has executed thus far, ξ_r. One way to generate legible motion is to model this inference, and create motion that enables the right inference from the observer. Assume the user expects the robot to optimize the cost function C_r. Instead of directly optimizing C_r, the robot can explicitly optimize how easy it is for the user to infer its goal, G_r. A possible model is the one in (4), or its approximation in (9), enabling the robot to compute $P(G|\xi_r)$ for any G based on C_r. Here, we flip the robot and user roles: before, the robot was attempting to predict the user's goal; now the robot is attempting to model how the person would predict its goal, G_r.

Armed with a model, the robot can now produce legible trajectories (subject to the accuracy of the model's predictions) by optimizing a specific legibility criterion, as opposed to the function the user expects, C_r. A legible trajectory is one that enables quick and confident predictions. Criteria for legibility therefore track $P(G|\xi_r)$ for the

Fig. 11. A user with an unusual strategy for teleoperating. She moves the robot one joint at a time.

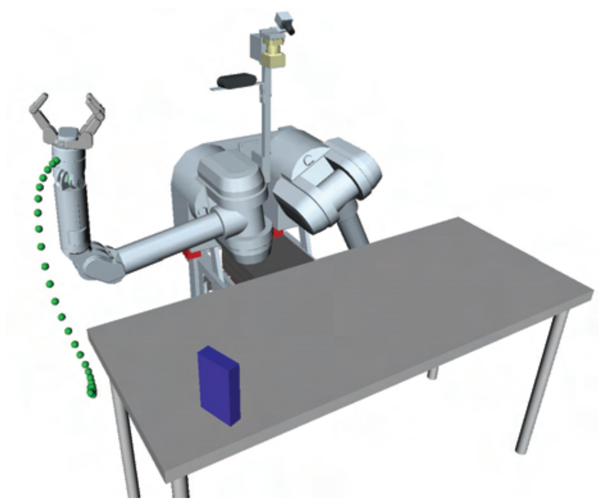


Fig. 12. A user's trajectory during the aggressive mode. Users can get careless with their input during aggressive arbitration, which can affect prediction.

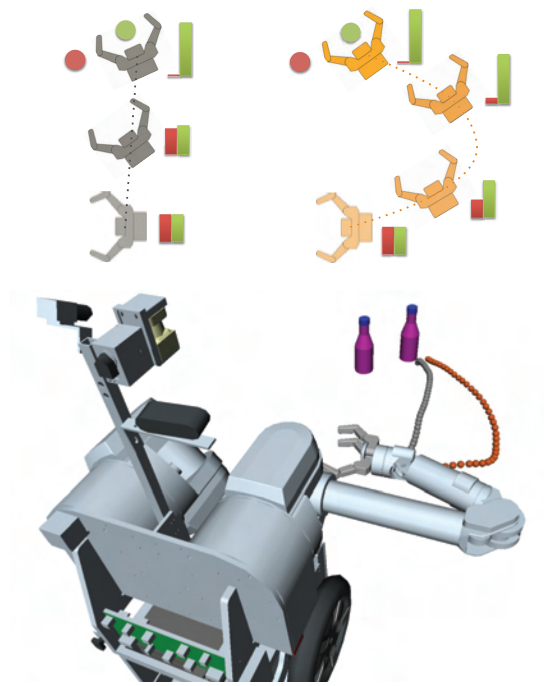


Fig. 13. Above: A hand trajectory that optimizes C_r (left); a hand trajectory that is legible (right). Below: similar trajectories in HERB. The trajectory directly optimizing C_r is more ambiguous, making it harder for the user to infer the robot's intent early on, as indicated by the green/red bars alongside the hands.

predicted goal G that the robot is aiming for. Figure 13 compares a trajectory with low-cost C_r with a trajectory with high legibility score in a criterion averaging the probabilities assigned to the robot's actual goal across the trajectory.

6.3. User reaction

Sharing control raises opportunities stemming from having a user in the loop. Key among them is that users directly react to what the robot does.

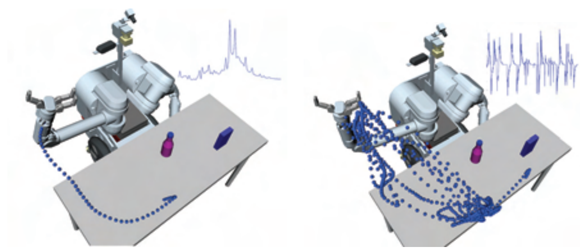
Detecting 'aggressive and wrong'. Being aggressive and wrong results in large penalties in time and user preference. Fortunately, it is also a state that can easily be identified and remedied by the user. Because prediction affects the user's behavior, when the robot eagerly starts heading towards the wrong target, the user rapidly attempts to get back control by moving against the robot's policy, as in Figure 14. This, in turn, decreases the predictor's confidence, causing the robot to begin following the user. This state can be detected early (Figure 15) by comparing the user's trajectory in the right and wrong cases, along with the dot product between the robot's policy and its actual velocity.

Provoking 'aggressive and wrong'. While the robot can capitalize on the user's reaction to infer that it has made the wrong prediction, it could also purposefully probe the user to trigger a reaction, as an information-gathering action in a game-theoretic setup. In doing so, the robot could become more confident in its prediction earlier on. How much users enjoy the experience when the robot acts in this manner, however, remains an open research question.

7. Limitations

Our work is limited in many ways, from formalism to analysis. Our formalism is restricted to the shared autonomy paradigm requiring a user to provide continuous input, and

Fig. 14. Users have strong reactions when the robot does the wrong thing. As soon as this user realizes that the robot will grasp the bottle, he moves his arm towards the right to show the robot that this is not his intended goal.



(a) Hard, Right, Aggressive (b) Hard, Wrong, Aggressive

Fig. 15. A comparison of the user input trajectories for right vs wrong. The graphs show the dot product between the robot's policy and the robot's actual velocity.

having a robot modify and play out this input; we do not focus on different, albeit useful, paradigms in which, for example, the robot is given high-level commands, or the user is in charge of setting the arbitration level. The formalism is also one possible interpretation of assistance, not *the* interpretation; it provides but one interpretation of prior work, while there can be many others.

Our analysis is done with a limited number of users, in a typical lab setting that is likely to bias the way they act and respond to the system, and with a simple system that enabled us to control for all the factors we wanted to manipulate. Our user pool was not comprised of the usual robot operators, for example users trained in disaster relief or users with disabilities. Our prediction analysis used data biased by the study setup. Therefore, our results throughout the paper should be interpreted as useful trends rather than irrefutable facts.

8. Conclusion

In this work, we presented an analysis of systems that share control with their users in order to assist them in completing the task. We proposed a formalism for assistance, namely policy blending, which we hope will provide a common ground for future methods and comparisons of such systems. We investigated aggressiveness, prediction correctness, and task difficulty as factors that affect assistance performance and user preferences, analyzing their interaction in a user study. Our data suggest that arbitration should take into account the robot's confidence in itself (i.e. in the correctness of the predicted policy) and in the user (i.e. in how easy the task is for the user). Our study identified potential challenges of such systems, which form interesting avenues for future work, such as behavior adaptation to the user and intent expression. Furthermore, unlike for typical intent prediction tasks, the robot's prediction directly affects the user's behavior. This gives the learner the opportunity to improve its predictions by explicitly incorporating the user's reaction, and even by taking purposeful information-gathering actions to disambiguate among its hypotheses.

Notes

1. In the case of velocity inputs, the robot applies its motion model to obtain the configuration U.
2. Although considering current velocity in amnesic prediction can help with this, this information is still local. As in Figure 3, incorporating global knowledge from the trajectory can be beneficial.

Funding

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.

Acknowledgments

We thank G Gordon, J Forlizzi, C Rose, A Collet, C Dellin, M Cakmak, MK Lee, M Vazquez, and the members of the Personal Robotics Lab for very helpful discussion and advice.

References

Aarno D, Ekvall S and Kragic D (2005) Adaptive virtual fixtures for machine-assisted teleoperation tasks. In: *IEEE international conference on robotics and automation*.

Abbeel P and Ng AY (2004) Apprenticeship learning via inverse reinforcement learning. In: *International conference on machine learning*.

Aigner P and McCarragher B (1997) Human integration into robot control utilising potential fields. In: *IEEE international conference on robotics and automation*.

Anderson SJ, Peters SC, Iagnemma K, et al. (2010) Semi-autonomous stability control and hazard avoidance for manned and unmanned ground vehicles. In: *Proceedings of the 27th army science conference*.

Bruemmer DJ, Dudenhoeffer DD and Marble J (2002) Dynamic autonomy for urban search and rescue. In: *2002 AAAI mobile robot workshop*, Canada.

Crandall J and Goodrich M (2002) Characterizing efficiency of human robot interaction: A case study of shared-control teleoperation. In: *IEEE/RSJ international conference on intelligent robots and systems*.

Debus T, Stoll J, Howe R, et al. (2000) Cooperative human and machine perception in teleoperated assembly. In: *International symposium on experimental robotics*.

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*. pp. 327–361, Cambridge, MA, USA: MIT Press.

Desai M and Yanco HA (2005) Blending human and robot inputs for sliding scale autonomy. In: *International workshop on robot and human interactive communication*, pp. 537–542.

Dey D, Liu TY, Hebert M, et al. (2012) Contextual sequence prediction with application to control library optimization. In: *Proceedings of robotics: Science and systems VIII*.

Dias MB, Kannan B, Browning B, et al. (2008) *Sliding autonomy for peer-to-peer human-robot teams.* Technical report CMU-RI-TR-08-16, Robotics Institute, Pittsburgh, PA.

Dragan A and Srinivasa S (2012) Online customization of teleoperation interfaces. In: *International symposium on human and robot communication (Ro-Man)*.

Dragan A, Gordon G and Srinivasa S (2011a) Learning from experience in manipulation planning: Setting the right goals. In: *Proceedings of the international symposium on robotics research*.

Dragan A, Ratliff N and Srinivasa S (2011b) Manipulation planning with goal sets using constrained trajectory optimization. In: *IEEE international conference on robotics and automation*.

Enes A and Book W (2010) Blended shared control of Zermelo's navigation problem. In: *American Control Conference*, pp. 4307–4312.

Fagg AH, Rosenstein M, Platt R, et al. (2004) Extracting user intent in mixed initiative teleoperator control. In *Proceedings of the American Institute of Aeronautics and Astronautics intelligent systems technical conference*.

Gerdes JC and Rossetter EJ (2001) A unified approach to driver assistance systems based on artificial potential fields. *Journal of Dynamic Systems, Measurement, and Control* 123(3): 431–438.

Glynn SJ and Henning RA (2000) Can teams outperform individuals in a simulated dynamic control task. *Proceedings of the human factors and ergonomics society annual meeting*.

Goertz R (1963) Manipulators used for handling radioactive materials. In: Bennett EM (ed) *Human Factors in Technology*. New York, NY: McGraw-Hill.

Hauser K (2012) Recognition, prediction, and planning for assisted teleoperation of freeform tasks. In: *Proceedings of robotics: Science and systems*, Sydney, Australia.

Jetchev N and Toussaint M (2009) Trajectory prediction: Learning to map situations to robot trajectories. In: *Proceedings of the 26th annual international conference on machine learning*, New York, NY, pp. 449–456.

Kalakrishnan M, Chitta S, Theodorou E, et al. (2011) STOMP: Stochastic trajectory optimization for motion planning. In: *IEEE international conference on robotics and automation*.

Kim D-J, Hazlett-Knudsen R, Culver-Godfrey H, et al. (2011) How autonomy impacts performance and satisfaction: Results from a study with spinal cord injured subjects using an assistive robot. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans* 42(1): 2–14.

Kofman J, Wu X, Luu T, et al. (2005) Teleoperation of a robot manipulator using a vision-based human-robot interface. *IEEE Transactions on Industrial Electronics* 52(5): 1206–1219.

Kortenkamp D, Keirn-Schreckenghost D and Bonasso R (2000) Adjustable control autonomy for manned space flight. In: *IEEE aerospace conference proceedings*.

Kragic D, Marayong P, Li M, et al. (2005) Human-machine collaborative systems for microsurgical applications. *International Journal of Robotics Research* 24: 731–741.

Leeper AE, Hsiao K, Ciocarlie M, et al. (2012) Strategies for human-in-the-loop robotic grasping. In: *Proceedings of human–robot interaction conference*.

Li M and Okamura A (2003) Recognition of operator motions for real-time assistance using virtual fixtures. In: *Proceedings of haptic interfaces for virtual environment and teleoperator systems conference*.

Loizou S and Kumar V (2007) Mixed initiative control of autonomous vehicles. In: *IEEE international conference on robotics and automation*.

Marayong P, Li M, Okamura A, et al. (2003) Spatial motion constraints: Theory and demonstrations for robot guidance using virtual fixtures. In: *IEEE international conference on robotics and automation*.

Marayong P, Okamura AM and Bettini A (2002) Effect of virtual fixture compliance on human-machine cooperative manipulation. In: *IEEE/RSJ international conference on intelligent robots and systems*.

Nudehi S, Mukherjee R and Ghodoussi M (2005) A shared-control approach to haptic interface design for minimally invasive telesurgical training. *IEEE Transactions on Control Systems Technology* 13(4): 588–592.

OpenNI (2010) 3D sensing framework. Available at: http://www.openni.org.

Platt JC (1999) Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In: *Advances in Large Margin Classifiers*. pp. 61–74, Cambridge, MA, USA: MIT Press.

Ratliff N, Bagnell JA and Zinkevich M (2006) Maximum margin planning. In: *International conference on machine learning*.

Ratliff N, Zucker M, Bagnell J, et al. (2009) Chomp: Gradient optimization techniques for efficient motion planning. In: *IEEE international conference on robotics and automation*.

Rosenberg L (1993) Virtual fixtures: Perceptual tools for telerobotic manipulation. In: *IEEE virtual reality annual international symposium*.

Schmidt CF and D'Addamio J (1973) A model of the commonsense theory of intention and personal causation. In: *Proceedings of the 3rd international joint conference on artificial intelligence*.

Sellner BP, Simmons R and Singh S (2005) User modelling for principled sliding autonomy in human-robot teams. In: *Proceedings of the NRL multirobot workshop 2005*.

Shen J, Ibanez-Guzman J, Ng TC, et al. (2004) A collaborative-shared control system with safe obstacle avoidance capability. In: *IEEE conference on robotics, automation and mechatronics*.

Smith C, Bratt M and Christensen H (2008) Teleoperation for a ball-catching task with significant dynamics. *Neural Networks* 21(4): 604–620.

Todorov E and Li W (2005) A generalized iterative LQG method for locally-optimal feedback control of constrained nonlinear stochastic systems. In: *Proceedings of the 2005 American Control Conference*.

Toussaint M (2009) Robot trajectory optimization using approximate inference. In: *Proceedings of the 26th annual international conference on machine learning.*

Vasquez D, Fraichard T, Aycard O, et al. (2005) Intentional motion on-line learning and prediction. *Machine Vision and Applications* 19(5–6): 411–425.

Waern A (1996) *Recognising Human Plans: Issues for Plan Recognition in Human-Computer Interaction.*

Weber C, Nitsch V, Unterhinninghofen U, et al. (2009) Position and force augmentation in a telepresence system and their effects on perceived realism. In: *Third joint EuroHaptics conference and symposium on haptic interfaces for virtual environment and teleoperator systems*, pp. 226–231.

You E and Hauser K (2011) Assisted teleoperation strategies for aggressively controlling a robot arm with 2D input. In: *Proceedings of robotics: Science and systems*.

Yu W, Alqasemi R, Dubey R, et al. (2005) Telemanipulation assistance based on motion intention recognition. In: *IEEE international conference on robotics and automation*.

Ziebart BD, Maas A, Bagnell JA, et al. (2008) Maximum entropy inverse reinforcement learning. In: *Proceedings of the Association for the Advancement of Artificial Intelligence*.

Ziebart BD, Ratliff N, Gallagher G, et al. (2009) Planning-based prediction for pedestrians. In: *IEEE/RSJ international conference on intelligent robots and systems*.

Zollner R, Rogalla O, Dillmann R, et al. (2002) Understanding users intention: Programming fine manipulation tasks by demonstration. In: *IEEE/RSJ international conference on intelligent robots and systems*, pp. 1114–1119.