Mode Switch Assistance to Maximize Human Intent Disambiguation

Deepak E. Gopinath[*][†], Brenna D. Argall[*][†][‡][§]
[*]Department of Mechanical Engineering, Northwestern University, Evanston, IL
[†]Rehabilitation Institute of Chicago, Chicago, IL
[‡]Department of Physical Medicine and Rehabilitation, Northwestern University, Chicago, IL
[§]Department of Electrical Engineering and Computer Science, Northwestern University, Evanston, IL
deepakgopinath@u.northwestern.edu
brenna.argall@northwestern.edu

***Abstract*—In this paper, we develop an algorithm for intent inference via goal disambiguation with a shared-control assistive robotic arm. Assistive systems are often required to infer human intent and this often is a bottleneck for providing assistance quickly and accurately. We introduce the notion of *inverse legibility* in which the human-generated actions are legible enough for the *robot* to infer the human intent confidently and accurately. The proposed disambiguation paradigm seeks to elicit legible control commands from the human by selecting control modes, for the robotic arm, in which human-directed motion will *maximally disambiguate* between multiple goals. We present simulation results which look into the robustness of our algorithm and the impact of the choice of confidence functions on the performance of the system. Our simulations results suggest that the choice of confidence function is a critical factor in determining the disambiguation algorithm's capability to capture human intent. We also present a pilot study that explores the efficacy of the algorithm on real hardware with promising preliminary results.**

I. Introduction

Assistive and rehabilitation devices such as powered wheelchairs, robotic arms and myoelectric prostheses play an important role in the lives of people with motor impairments. These devices help to increase the human's ability to perform activities of daily life and reduce their dependence on caretakers, and are crucial to revolutionizing the way people with motor impairments interact with society. As the field of assistive robotics progresses rapidly, the devices themselves become more capable and dextrous—and as a result also more complex, higher dimensional and harder to control.

To operate an assistive device, typically the human directly controls the device motion via a control interface. However, the more severe a person's motor impairment, the more limited are the control interfaces available for them to use. These interfaces (for example, a switch-based head array or Sip-N-Puff) are lower in dimensionality and bandwidth, and generally are able to operate on only a subset of the device control space at a given time. Therefore, we have a difficult situation, in which sophisticated assistive devices that require complex control strategies are paired with users with diminished ability to control them.

Due to the mismatch between the dimensionality of the control interface and controllable degrees-of-freedom (DoF) of the robotic device, control interfaces typically operate in *modes* which correspond to different partitions of the control space. By necessity, the more limited the control interface is, the greater number of modes there are. In order to have full control of the robot the user will have to switch between the different partitions and this is known as *mode switching* or *modal control* [19, 23].

It has been established that mode switching is expensive and as a result task performance is degraded [14]. Furthermore, it adds to the cognitive and physical burden of the human as each of these mode switches requires the user to shift their attention from the task to performing the mode switch. The introduction of *shared autonomy* to these systems helps to alleviate and address some of these issues by letting the system take partial responsibility of the task execution, thereby reducing the human effort in achieving a goal.

Any assistive autonomy system typically needs an idea of what it is the human is trying to do—either by explicit indication from the user of the task or goal [6], or by inferring the human's intent from their control signals or sensor data [22, 25]. The question of intent inference thus is key. In this paper, we develop an assistance paradigm that helps with intent inference, by selecting the control mode in which robot motion will *maximally disambiguate* human intent. The faster the autonomy is able to disambiguate intent, the earlier the robot is able to provide autonomy assistance—leading ideally to fewer mode switches and less burdensome executions.

Within the field of Human-Robot Interaction (HRI) a *legible*

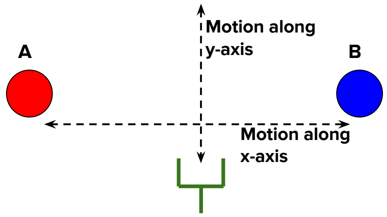


Fig. 1. Illustration of goal disambiguation along various control dimensions. Any motion of the end effector (green) along the y-axis will not help the system to disambiguate the two goals (A and B). However, motion along the x-axis provides cues as to which goal.

motion is one which helps the observer (usually the human) decipher the intent behind the robot's action more *quickly* and *confidently* [11]. It can be the case that certain actions *by the human* might carry more information about the human's intent—which can then help the robot to draw useful and correct inferences more easily. Therefore, in this paper we propose a paradigm of *inverse legibility* in which the roles are switched and the human-generated actions *help the robot* to infer human intent confidently and accurately.

Consider the example illustrated in Figure 1. A human control command issued along the x dimension is more *intent expressive* and helps the robot to provide appropriate assistance more quickly and confidently. With the disambiguation assistance scheme developed in this work, we hope to elicit more legible *human* control commands by placing the user control in those modes that *maximally disambiguate* between the various goals in the scene.

In Section II we present an overview of relevant research in the area of shared-control in assistance systems focusing on mode switching, legibility and synergies in HRI. Section III describes the mathematical formalism for the algorithm and the metric used for goal disambiguation, and Section IV describes the details of our implementation. The simulation results are presented in Section V, and the pilot study in Section VI. Conclusions are provided in Section VII.

II. Related Work

This section provides a brief overview of related research in the areas of intent inference, robot assistance for modal control, legibility and cooperation in HRI.

Shared-control assistance paradigms help to offload cognitive and physical burden [24] without requiring the user to relinquish complete control, and are usually preferred over fully autonomous assistive robotic systems for reasons of both robustness and user satisfaction. Often, shared-control systems require an estimate of the human's intent—their intended task, goal or motion, for example. Methods for intent inference have been extensively studied by roboticists and cognitive scientists alike and can be broadly classified into two categories: model-based and heuristic-based [4]. In model-based approaches the agent is typically modeled as a Partially Observable Markov Decision Process (POMDP) that acts according to a policy that maps states to actions. In such settings, intent inference reduces to solving the inverse problem of computing the posterior distribution over mental states conditioned on observed actions (Bayesian Inference) [2, 3]. Heuristic-based approaches instead seek to find direct mappings between some signals—such as low-level motion cues [5] or biological signals [7]—and the underlying human intention. Our approach builds on a heuristic-based intent inference framework. More specifically, we use a *confidence function* as a measure of the robot's estimate that a particular goal is indeed the user's intended goal. Confidence functions often depend on the human control command, autonomous policy, robot pose or goal locations, and often are used to dictate how much control lies with the robot versus with the human in the shared-control system.

The cognitive burden of shifting focus (*task switching*) from the task at hand to switching between control modes can result in a significant decrease in task performance regardless of the control modality [18]. Users in many cases find *modal control* and *mode switching* to be slow, difficult or burdensome [14]. Even a simple time-optimal automatic mode switching system can significantly improve user satisfaction while maintaining quality task performance [14]. However, it also is not always the case that users are trying to optimize for time or effort during task execution [13]. Our present system therefore does not make *a priori* assumptions regarding the optimizing principles at work when a user operates a robot.

The legibility and predictability of robot motion *to the human* is investigated [11], and different methods for generating legible robot motion are proposed [15]. We apply this concept of legibility however to the *human control* commands, such that the intent expressed in the human command is clear *to the robot*. We term this *inverse legibility*. Our assistance scheme is intended to bring out a more legible intent-expressive control command from the human, by placing the user control in that mode which can provide maximal goal disambiguation and improved legibility.

Eliciting legible commands from the user can also be thought of as an information acquisition problem. Information acquisition in robotic systems have been widely studied, primarily in the context of generating optimal control for mobile robot sensing systems. A typical approach is to model the problem as an optimal control problem with an associated reward structure that reflects some measure of information gain [1]. The problem of information gathering, for example, can be formulated as maximizing the ergodicity of a robot's trajectory with respect to an underlying information density map [16, 17] by evaluating the expected value of the Fisher information—the amount of information a single measurement reveals at a location.

Also related to our work is the idea of mutual cooperation between humans and robots, and the underlying synergies that are crucial for successful human-robot interaction. In order to overcome the communication bottleneck that exists between robots and people during human-robot interactions, different types of user interfaces have been developed that account for constrained capabilities of the robot [12]. A framework for *"people helping robots helping people"* has humans provide semantic information and judgments about the environment to the robot, which then utilizes them to improve its own capabilities [21]. A *symbiotic* human robot interaction scheme aims to overcome perceptual and cognitive limitations that robots might encounter while still allowing the robots to help humans [20]. From the robot's perspective the key concept behind our algorithm is the idea of *"Help Me, Help You"*—that is, if the human can "help" the robot by providing more legible control commands, then the robot in turn can assist the human more quickly and effectively.

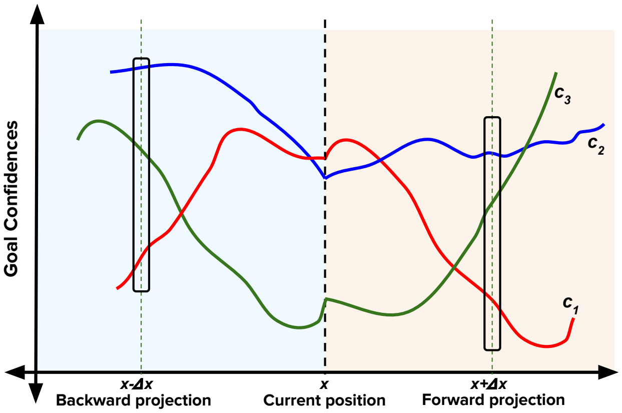


Fig. 2. Illustration of change in confidence with movement along a single control dimension. A very different distribution of confidences results from positive motion (beige) versus negative motion (blue). Snapshots after moving by amount Δx highlighted by rectangles.

III. ALGORITHM DESIGN

This section describes our algorithm that computes the control mode that is able to maximally disambiguate between human goals, thereby eliciting the most legible control command from the human. Section III-A outlines the mathematical notation used in the paper and Section III-B describes the formulation and computation of the metric used to disambiguate human intent.

A. Notation

Let $\mathcal{G}$ be the set of all candidate user goals with $n_g = |\mathcal{G}|$ and let g^i refer to the i^{th} goal where $i \in [1, 2, \ldots, n_g]$. A *goal* represents the intent of the human, and might be a task or location in the world, for example. The set of goals corresponds to an associated set of confidences denoted as $\mathcal{C}$, where c^i refers to the individual confidence associated with goal g^i—that is, the robot's confidence that g^i represents the human's intent. Let $\mathcal{K}$ be the control space in which the robot operates and k^i refer to an individual control dimension where $i \in [1, 2, \ldots, n_k]$. The cardinality of $\mathcal{K}$ depends on the robotic platform; for example, for a smart wheelchair $n_k = 2$ whereas for a six degrees-of-freedom robotic arm $n_k = 6$.

For control purposes, the set $\mathcal{K}$ is partitioned into subsets known as *modes*. Let $\mathcal{M}$ refer to the set of all modes that the control space $\mathcal{K}$ is partitioned into with $n_m = |\mathcal{M}|$. The number of modes n_m is specific to the control interface and mapping to $\mathcal{K}$. Furthermore, let m^i refer to the i^{th} mode where $i \in [1, 2, \ldots, n_m]$.

Another quantity of interest is the spatial gradient of individual goal confidences for motions along the various control dimensions. More specifically, the gradient is denoted by $\frac{\partial c}{\partial x_k}$, where $c \in \mathcal{C}$ and x_k is the component of robot's position along control dimension k. Furthermore, since the confidence function in general can assume drastically different values upon moving in positive and negative directions within a given control dimension, the positive and negative gradients are explicitly denoted as $\frac{\partial c}{\partial x_k}^{+}$ and $\frac{\partial c}{\partial x_k}^{-}$ respectively. The formalism developed in Section III-B is agnostic to the particular form of confidence function. Additionally, an analytical closed-form expression for the gradient might not always be available, as confidence functions need not be continuous and differentiable. Even when available, such an expression might be expensive to compute. In such cases, the gradient can be numerically approximated, which we derive in Section III-B4.

We define a *disambiguation* metric $D_k \in \mathbb{R}$ for each control dimension $k \in \mathcal{K}$, which is a function of c and $\frac{\partial c}{\partial x_k}$. Analogous to the gradients, we explicitly define disambiguation metrics for both positive and negative motion directions as D_k^+ and D_k^-, respectively. We further define a disambiguation metric $D_m \in \mathbb{R}$ for each control mode $m \in \mathcal{M}$. The disambiguation metric D_m is a measure of how legible the user commands would be if the user were to control the robot in mode m. The higher the value, the easier it will be for the system to infer the human's intent.[1]

B. Disambiguation Metric

The disambiguation metric D_k encodes different aspects of how the goal confidences change upon moving along control dimension k. Figure 2 is an illustrative example which shows how goal confidences can vary as a function of position along a control dimension.

A proper design of D_k should take into account both immediate as well as long term benefits of moving in k. In order to compute the goal confidences after a small user-initiated robot motion we sample the confidence function in the neighborhood $x \pm \Delta x$ of the current pose x of the robot, where Δx is a small change along the control dimension.

An additional consideration is that a confidence function might depend on the user control command ($\boldsymbol{u}_h$). For such functions, during the sampling procedure the value of $\boldsymbol{u}_h$ is set as $\frac{\Delta x}{\Delta t}$—that is, the velocity that would result in movement from x to $x + \Delta x$ in one execution timestep Δt.

We identify four important components to inform the design of D_k. Each of the following computations is done in reference to the projection of this motion along control dimension k.

1) Maximum of confidences: The maximum of the goal confidences is a good measure of the system's overall certainty in accurately estimating human intent. A higher maximum implies that the robot has an even better idea of what the human is trying to do. The max Γ_k is computed as

$$\Gamma_k = \max_{1 \le i \le n_g} c^i_{\delta_x}$$

2) Difference between largest confidences: Since it is possible to have multiple[2] highly confident goals, accurate disambiguation also benefits from a large separation between the first and second most confident goals. This difference is denoted

[1] Going forward, subscript k will be dropped from x_k for notational brevity.

[2] Note that confidences are not normalized, since we do care about more than just their relative magnitudes (bullet 1).

by Ω_k and is computed as

$$\Omega_k = \max(\mathcal{C}_k) - \max(\mathcal{C}_k \setminus \max(\mathcal{C}_k))$$

where $\mathcal{C}_k$ is the set of projections of all goals confidences along a control dimension k.

3) Separation in confidences: If the difference between the largest confidences fails to disambiguate, then the *separation*, Λ_k, in the remaining goal confidences of a control dimension will further aid the disambiguation. At any point in space this can be computed as the *sum of pairwise distances* between the n_g confidences. Thus,

$$\Lambda_k = \sum_{p=1}^{n_g} \sum_{q=p}^{n_g} |c^p_{\delta_x} - c^q_{\delta_x}|$$

where δ_x indicates $x + \Delta x$ or $x - \Delta x$ depending on the direction of perturbation and $|\cdot|$ denotes the absolute value.

4) Gradients: The propensity for change and information gain upon the continuation of motion along control dimension k is encoded in the gradients $\frac{\partial c}{\partial x}$. The greater the difference between the individual confidence gradients, the greater will these confidences deviate from each other over time. Instead of using closed-form analytical gradients, we approximate the gradients numerically. Therefore,

$$\frac{\partial c}{\partial x} \approx c_{\delta_x} - c_x$$

where c_x denotes the confidence at location x. In order to quantify the "spread" of gradients we define a quantity Υ_k which is computed as

$$\Upsilon_k = \sum_{p=1}^{n_g} \sum_{q=p}^{n_g} \left| \frac{\partial c^p}{\partial x} - \frac{\partial c^q}{\partial x} \right|$$

Putting it all together: Γ_k, Ω_k, Λ_k and Υ_k are then combined to compute D_k as

$$D_k = \underbrace{w \cdot (\Gamma_k \cdot \Omega_k \cdot \Lambda_k)}_{\text{short-term}} + \underbrace{(1-w) \cdot \Upsilon_k}_{\text{long-term}} \qquad (1)$$

where w controls the relative contribution of short-term and the long-term benefit and is task-specific. (In our implementation, $w = 0.5$.) Equation 1 actually is computed twice, once in each of the positive ($\delta_x = x + \Delta x$) and negative directions ($\delta_x = x - \Delta x$), and the results are then summed. The computation of D_k (Equation 1) is performed for each $k \in \mathcal{K}$. Then, the disambiguation metric D_m for control mode m is calculated as

$$D_m = \sum_k D_k \ \ \forall \ \ k \in m. \qquad (2)$$

Lastly, the control mode with highest disambiguation capability m^* is given by

$$m^* = \operatorname*{argmax}_m D_m$$

and the control dimension with highest disambiguation capability k^* is given by

$$k^* = \operatorname*{argmax}_k D_k.$$

. Disambiguation mode m^* is the mode that the algorithm chooses *for* the human to better estimate their intent. Any control command issued by the user in m^* is likely to be more legible due to maximal goal confidence disambiguation.

IV. IMPLEMENTATION

This section describes the details of our implementation. Section IV-A discusses various confidence functions used to infer user intent, followed by the details of the underlying shared-control system in Section IV-B.

A. Confidence Function

The algorithm proposed in this paper requires that the confidence measure varies as a function of $\boldsymbol{x}$, so that $\frac{\partial c}{\partial x}$ is well-defined and exists. The choice of confidence function is up to the system designer and numerous options exist.

We implement two confidence functions in this work. A simple proximity-based confidence function used extensively in the literature [8, 9, 10] is

$$\textbf{C1}: \quad c(\boldsymbol{x}, \boldsymbol{x_g}) = \max(0, 1 - \frac{\|\boldsymbol{x} - \boldsymbol{x}_g\|}{r})$$

where $\boldsymbol{x}$ is the current position of the robot, $\boldsymbol{x}_g$ is the location of goal g, r is the radius of a sphere beyond which the confidence is always 0 and $\|\cdot\|$ is the Euclidean norm. We refer to this confidence function as **C1**.

A weakness of confidence measure **C1** is that it considers only current position and ignores any cues regarding human intent present in the control command itself. A confidence function that instead incorporates the human's control command will contain more information content. One such function aims to capture the "directedness" of the human control command towards a goal position

$$\textbf{C2}: \quad c(\boldsymbol{x}, \boldsymbol{x_g}, \boldsymbol{u}_h) = \boldsymbol{u}_h \cdot (\boldsymbol{x}_g - \boldsymbol{x})$$

where $\boldsymbol{u}_h$ is the human control command. We refer to this confidence function as **C2**.

B. Control Sharing Paradigm

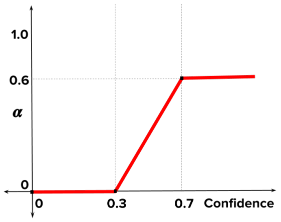


Fig. 3. A prototypical arbitration function.

In our implementation, the proposed disambiguation assistance paradigm augments a blending-based shared-control system, in which the final control command issued to the robot is a blended sum of the human control command and an autonomous robot policy. The control signal from the robot autonomy is generated by a function $f_r(\cdot) \in \mathcal{F}_r$,

$$\boldsymbol{u}_r \leftarrow f_r(\boldsymbol{x})$$

where $\mathcal{F}_r$ is the set of all control behaviors corresponding to different tasks. Specifically, let $\boldsymbol{u}_{r,g}$ be the autonomous control policy associated with goal g. The final control command $\boldsymbol{u}$, issued to the robot then is given as

$$\boldsymbol{u} = \alpha \cdot \boldsymbol{u}_{r,g^*} + (1 - \alpha) \cdot \boldsymbol{u}_h$$

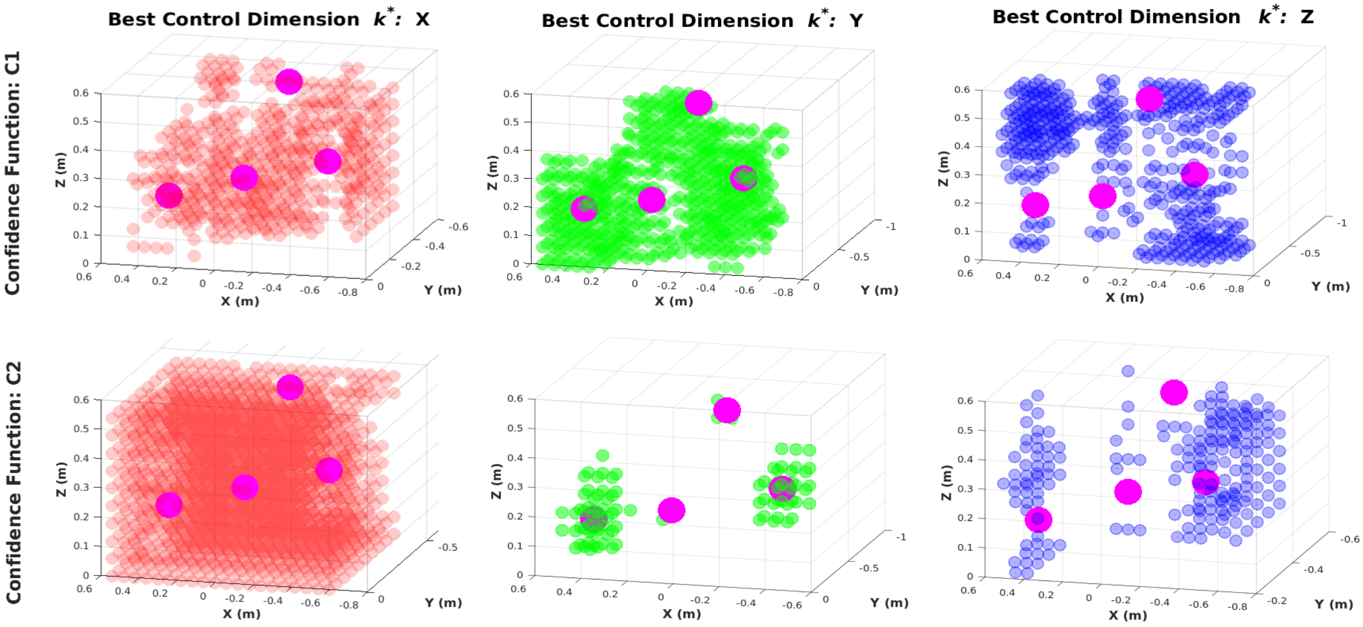


Fig. 4. Control dimensions best able to disambiguate intent. Using confidence functions **C1** (top row) and **C2** (bottom row). Left column: k^* is X. Middle Column: k^* is Y. Right Column: k^* is Z. Magenta spheres indicate the goal locations (intent).

where g^* is the most confident goal. The blending factor α is a function of the system's confidence (Figure 3).

The robot control command $\boldsymbol{u}_{r,g}$ is generated using a simple potential field based dynamical system which is defined in all parts of the state space. In our implementation, each goal is a position in space ($\boldsymbol{x}_g$). Every goal g is associated with a potential field P_g which treats g as an attractor and all the other goals in the scene as repellers. For potential field P_g, the attractor velocity is given by

$$\dot{\boldsymbol{x}}_{attract} = \boldsymbol{x}_g - \boldsymbol{x}$$

where $\boldsymbol{x}_g$ is the location of g. The repeller velocity is given by

$$\dot{\boldsymbol{x}}_{repel} = \sum_{i \in \mathcal{G} \setminus g} \frac{\boldsymbol{x} - \boldsymbol{x}_i}{\eta(\|\boldsymbol{x} - \boldsymbol{x}_i\|^2)}$$

where $\dot{\boldsymbol{x}}$ indicates the velocity of the robot in the world frame and η controls the magnitude of the repeller velocity. Therefore,

$$\boldsymbol{u}_{r,g} = \dot{\boldsymbol{x}}_{attract} + \dot{\boldsymbol{x}}_{repel}$$

Additionally, P_g operates in the full six dimensional Cartesian space and treats position and orientation as independent potential fields.

V. SIMULATION ANALYSIS

In this section we qualitatively look at whether our disambiguation algorithm selects modes that are useful for intent disambiguation for a given goal configuration. We also evaluate the impact of certain simplification assumptions in our algorithm and of the choice of confidence functions within a simulated environment.

A. *Choice of Confidence Functions*

In order to qualitatively assess the correctness of our algorithm, we perform simulations in which k^* is computed at 2000 uniformly sampled points in the workspace of a robotic arm (a simulated version of the robotic arm described in Section VI-A), approximated as a $1.2 \times 0.6 \times 0.7m^3$ volume in front of the robot. Confidence functions **C1** ($r = 0.3m$) and **C2** are evaluated using a goal configuration shown in Figure 6 (middle column). Since the target orientations are the same for all goals, disambiguation only happened in the translational dimensions and therefore is reduced to a 3-D problem.

Figure 4 shows the results of the simulation. For the goal distribution used in the simulation, the goal positions are spread out maximally along the x and z axes. Intuitively the system will be able to more quickly infer the humans intent if the human control command is either along the x or the z axes. Table I further reports the number of times the algorithm picked each of the three control dimensions, for each confidence function.

These results shed light on the efficacy of a confidence function in properly capturing human intent. The choice of confidence functions can greatly affect the computation of k^* and m, and so the power of our disambiguating algorithm is intimately linked to the inference power of different choices of confidence function.

Under confidence function **C1** the information is equally spread throughout all control dimensions (Table I) because **C1** contains less information content with respect to the users selected motion and therefore also their intended goal. Furthermore, **C1** had "null" spaces where all confidences are identically equal to zero—and therefore neither disambigua-

Best control dimension distribution				
Confidence Function	**X**	**Y**	**Z**	**NULL**
C1	579	615	446	360
C2	1711	93	196	0

TABLE I
BEST CONTROL DIMENSION DISTRIBUTION FOR TWO DIFFERENT CONFIDENCE FUNCTIONS.

tion nor intent inference are possible.

By contrast, using **C2**, x is identified as the preferred dimension in 1711 out of 2000 samples, and z in 196 of the remaining 289 samples, which indicates that the confidence function along with our algorithm is able to select the disambiguating dimensions over 95% of the time. The algorithm picked y only when the robot is directly in front of a goal.

B. Characterization of Simplifying Assumption

In our algorithm, the computation of D_m (Equation 2) only considers motion projected along perpendicular vectors: the axes of each dimension k^i of mode m. However, the user can generate a control command in any arbitrary direction within the control mode, and so the robot can move along any vector spanned by the control dimensions in m. In order to assess the impact of this simplification, we perform simulations in which m^* is computed for 500 uniformly spaced locations in the robot workspace. At each of those points, 100 random control commands feasible in m^* are generated and applied to perturb the robot. Finally, at each of these perturbed positions the best control mode is once again computed.

If the best mode in the perturbed position is indeed mode m^*, then the simplification does not adversely affect the identification of the disambiguating mode. Table II summarizes the number of times a match occurs for different configurations of the workspace (with n_g = $3, 4$ and 5). While the simplification does hold for 85-90% of off-axis motions, we also do observe a trend where performance drops as the number of goals increases. Intuitively this makes sense because disambiguation between goals will become harder with a larger number of goals in the scene.

C. Discussion

Our simulation results indicate a strong correlation between the intent inference power of a given confidence function and the disambiguation power of our algorithm. It is unsurprising that confidence functions which are information-poor approximations of human intent also perform less robustly when disambiguating between those approximations. Moreover, the algorithm could be used to pre-compute the most informative modes ahead of time, which then might be called on-demand during the task execution—which could prove helpful for complex tasks and/or limited interfaces that require more information from the human for disambiguation.

n_g	3	4	5
Accuracy%	89.24	87.09	86.11

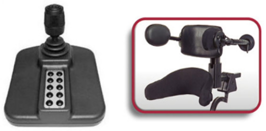
TABLE II
DISAMBIGUATION ACCURACY FOR OFF-AXIS MOTIONS

VI. PILOT STUDY

We next explore the use and utility of our disambiguation approach in a pilot study. Four subjects participated in the pilot study (3 male, 1 female), and all were lab members.

A. Hardware

The experiments were performed using the MICO robotic arm (Kinova Robotics, Canada), which is a 6-DoF robotic arm specifically designed for assistive purposes. The software system was implemented using the Robot Operating System (ROS) and data analysis was performed using MATLAB.[3]



Control Mappings		
Mode	**Head Array**	**Joystick**
1	v_x	v_x, v_y
2	v_y	v_x, v_z
3	v_z	ω_z, ω_y
4	ω_z	ω_x
5	ω_y	—
6	ω_x	—

Fig. 5. A 2-axis joystick (left) and switch-based head array (center) and their operational paradigms (right). v and ω indicate the translational and rotational velocities of the end-effector, respectively.

The human control command $\boldsymbol{u}_h$ was captured using two different control interfaces: a 2-axis joystick and a 1-D head array, shown in Figure 5. For both interfaces, the control interface signals were mapped to Cartesian velocities of the end-effector of the robot. Additionally, the interfaces were used to perform mode switches and to request mode switch assistance.

In detail, the joystick generated continuous signals and was capable of controlling a maximum of two control dimensions at a time. Different control modes could be accessed using the buttons on the interface. The head array generated 1-D discrete control signals, and consisted of three switches operated by the head and embedded within the back and sides of the headrest. When used for controlling a robotic arm, the switch at the back was used to cycle between the control modes, and the switches on the left and right sides controlled the motion of the end-effector in positive and negative directions along the control dimension of the selected control mode.

In using two interfaces, our aim was to observe whether differences in interface dimensionality and continuity correlated with any differences in task performance or mode switching behavior.

[3] Additionally, for the 6-D robot arm implementation, the computation of **C2** is split into translational and rotational components. (In Section V, only translational components were considered.) Here confidence function **C2** is computed as $c(\boldsymbol{x}, \boldsymbol{u}_h, \boldsymbol{u}_{r,g}) = \boldsymbol{u}_h^{trans} \cdot (\boldsymbol{x}_g - \boldsymbol{x})^{trans} + \boldsymbol{u}_h^{rot} \cdot \boldsymbol{u}_{r,g}^{(rot)}$ where $trans$ refers to the translational and rot the rotational parts of the full control space.

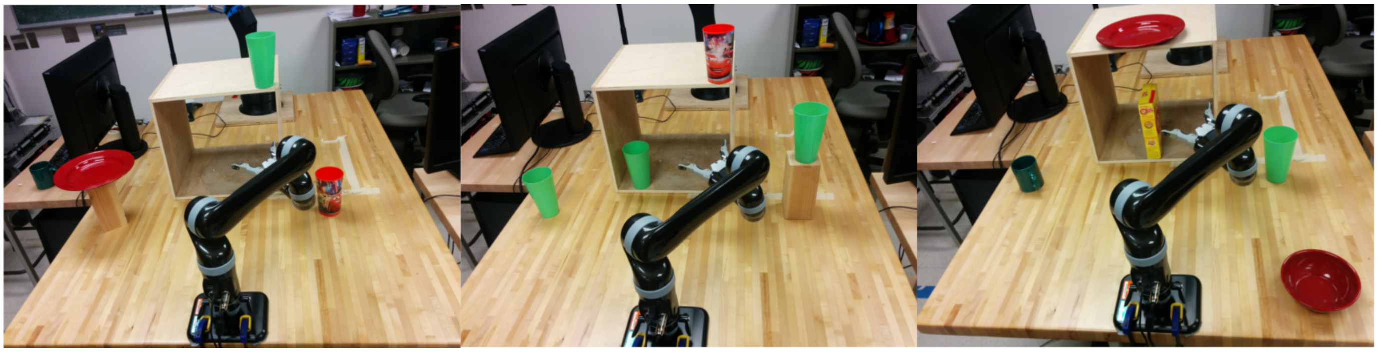


Fig. 6. Pilot study tasks. *Left to right:* Training, Testing (Easy), Testing (Hard).

B. Assistance Paradigms

Three kinds of mode switching paradigms were evaluated. Note that the blending assistance (as described in Section IV-B) was always running for all three paradigms. Since under the blending paradigm the amount of assistance directly depended on the level of confidence, this meant that if the intent inference improved as a result of disambiguation, more assistance would be provided by the robot.

Manual: The trial started in a randomized initial mode and during task execution the user manually performed all subsequent mode switches.

Disambiguation: The disambiguation system was activated right at the beginning of a trial. The algorithm identified the "best mode" m^* and started the trial in mode m^*. All subsequent mode switches were performed manually by the user. Furthermore, the user was required to first move in the selected mode before manually switching the mode.

On-Demand: The user could request mode switch assistance at any time during the task execution. This paradigm was exploratory and sought to find underlying patterns in assistance request behavior.

C. Tasks, Metrics and Protocol

Tasks: Training and Testing tasks were developed for the pilot study (Figure 6). **Training**: The user operated the robotic arm to perform simple reaching motions to three different goal locations. The primary purpose was for the user to become accustomed to the operation of the interfaces, the blending-based assistance and the experiment protocol. **Testing**: The user operated the robotic arm under two scenarios of varying difficulty. *Easy*: Four objects, all with the same grasp orientations. *Hard*: Five objects, all with different grasp orientations.

Metrics: *Task completion time* is the amount of time a user spent in accomplishing a task. *Mode switches* refers to the number of times the user switched between various modes while performing the task.

Protocol: A within-subjects study was conducted using a full factorial design in which the manipulated variables were the tasks, control interfaces and assistance paradigms. Each task consisted of two phases.

In Phase I, each user performed the task using both interfaces under the *Manual* and *Disambiguation* paradigms. The trials were balanced and the control interfaces and the paradigms were randomized to eliminate ordering effects. Participants were not aware of whether the starting mode was randomized or m^* (that is, of whether the paradigm for that trial was *Manual* or *Disambiguation*). The starting positions of the robot also were randomized to avoid bias. Three trials were collected for each permutation of manipulated variables. In Phase II, the user performed the same task using both interfaces and the *On-Demand* paradigm, and two trials were collected for each task-interface combination.

D. Pilot Study Results

An improvement in task performance in terms of a decrease in the number of mode switches was observed across both interfaces. Statistical significance in Figure 7 was determined by a two-sided Wilcoxon Rank-Sum Test, where (*) indicates $p < 0.05$ and (***) indicates $p < 0.001$.

Mode Switches: Figure 7 (top row) reveals a general trend of a decrease in the number of mode switches when disambiguation assistance was employed. This indicates that upon starting in a mode identified by the algorithm, the number of subsequent mode switches performed by the user was reduced.

Somewhat surprisingly, this difference was significant only when using the joystick. The head array moreover required more mode switches to complete the task. We also observe a much larger spread across trials and users when using the head array.

Task Completion Time: In Figure 7 (bottom row), a statistically significant decrease in task completion times between the *Manual* and *Disambiguation* paradigms was observed when using the joystick. The task completion times were comparable between the two paradigms when using the head array.

This difference between interfaces in task success might be explained by the fact that a *single* mode switch assistance at the beginning of the trial probably did not have a measurable impact in reducing the subsequent number of mode switches needed when using the head array—which required many more mode switches to achieve the task (Figure 7, top row).

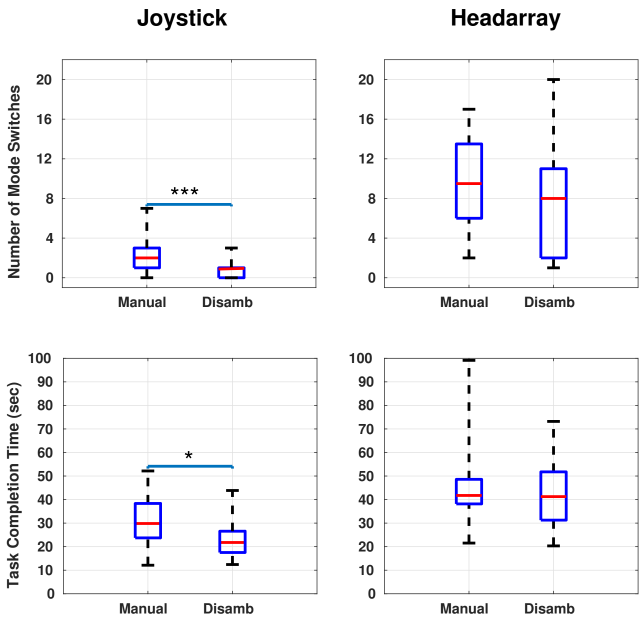


Fig. 7. Comparison of Disambiguation and Manual paradigms. Operation using joystick (left column) and head array (right column) interfaces. Evaluation of mode switches (top row) and completion time (bottom row). Box plots show the median and the quartiles.

Such a conclusion would have a utility implication for our algorithm: that as the number of required mode switches increases, so too must the number of disambiguation operations, if disambiguation assistance is useful.

<u>*On-Demand*</u>: The number of disambiguation requests under the *On-Demand* paradigm are reported in Table III. Although the subjects demonstrated a wide range of disambiguation request behaviors, we are able to observe a general trend of an increase in disambiguation requests with an increase in task difficulty. This shows that users were keen to explore the *On-Demand* option when the tasks became more difficult.

E. Future Work

Post-experiment feedback from the users also revealed that the subjects found the disambiguation assistance to be counter-intuitive at times in the *On-Demand* paradigm. This might be attributable to two main limitations of the current algorithm.

First, when the user requests assistance (in the *On-Demand* paradigm) at any point in time and space, the algorithm discards all information contained in the history of user commands from the start of the trial until that point. That is, the disambiguation algorithm lacks *memory* and reasons about the user's intent solely based on information that is available locally. However, it might be useful to bias the computation of the disambiguating control mode by incorporating information from the past history of the robot trajectory and control commands, by having a non-uniform prior over the intended goals. This would also likely improve the robustness and efficacy of the algorithm, and result in higher user acceptance.

	Joystick		Head Array	
Subject	*Easy*	*Hard*	*Easy*	*Hard*
1	1	0	5	6
2	1	1	3	6
3	2	2	4	5
4	2	5	17	7

TABLE III
NUMBER OF DISAMBIGUATION REQUESTS

Second, the algorithm only tries to maximize the utility value *for* the robot. However, it also is important to take into account the user's preference and ability to operate in the mode selected by the algorithm. Concepts from decision theory might be used to enhance the current framework.

Our pilot study revealed some promising trends and therefore a more extensive user study with motor-impaired subjects will be conducted in the future, to evaluate the utility of the disambiguation assistance system and further explore and understand the disambiguation request patterns of users.

VII. CONCLUSIONS

In this paper, we have presented an algorithm for *intent disambiguation assistance* with a shared-control robotic arm. We also introduced the notion of *inverse legibility*, in which the human-generated actions are legible enough *for* the robot to infer the human intent confidently and accurately. The goal of the algorithm developed in this paper is to seek legible control commands from the human by placing control in those modes able to *maximally disambiguate* between various goals. Preliminary pilot study results suggested that the proposed disambiguation paradigm to be promising. Our simulation work evaluated the impact of the simplifying assumptions and of different confidence functions on intent disambiguation. In our future work, as informed by the pilot study, we plan to enhance the algorithm and extend the framework into an automated mode switch assistance system.

ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation under Grant CNS 15544741. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the aforementioned institutions.

REFERENCES

[1] Nikolay Atanasov, Jerome Le Ny, Kostas Daniilidis, and George J. Pappas. Information acquisition with sensing robots: Algorithms and error bounds. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2014.

[2] Chris L. Baker, Joshua B. Tenenbaum, and Rebecca R. Saxe. Goal inference as inverse planning. In *Proceedings of the Cognitive Science Society*, 2007.

[3] Chris L. Baker, Rebecca R. Saxe, and Joshua B. Tenenbaum. Action understanding as inverse planning. *Cognition*, 113(3):329–349, 2009.

[4] Chris L. Baker, Julian Jara-Ettinger, Rebecca R. Saxe, and Joshua B. Tenenbaum. Rational quantitative attribution of beliefs, desires and percepts in human mentalizing. *Nature Human Behaviour*, 1:0064, 2017.

[5] Clark H. Barrett, Peter M. Todd, Geoffrey F. Miller, and Philip W. Blythe. Accurate judgments of intention from motion cues alone: A cross-cultural study. *Evolution and Human Behavior*, 26(4):313–331, 2005.

[6] Young Sang Choi, Cressel D. Anderson, Jonathan D. Glass, and Charles C. Kemp. Laser pointers and a touch screen: intuitive interfaces for autonomous mobile manipulation for the motor impaired. In *Proceedings of the International SIGACCESS Conference on Computers and Accessibility*, 2008.

[7] John P. Donoghue. Connecting cortex to machines: recent advances in brain interfaces. *Nature Neuroscience*, 5: 1085–1088, 2002.

[8] Anca D. Dragan and Siddhartha S. Srinivasa. Assistive teleoperation for manipulation tasks. In *Proceedings of ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 2012.

[9] Anca D. Dragan and Siddhartha S. Srinivasa. *Formalizing assistive teleoperation*. MIT Press, 2012.

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

[11] Anca D. Dragan, Kenton CT. Lee, and Siddhartha S. Srinivasa. Legibility and predictability of robot motion. In *Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 2013.

[12] Ian J. Goodfellow, Nate Koenig, Marius Muja, Caroline Pantofaru, Alexander Sorokin, and Leila Takayama. Help me help you: Interfaces for personal robots. In *Proceedings of ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 2010.

[13] Deepak Gopinath, Siddarth Jain, and Brenna D. Argall. Human-in-the-loop optimization of shared autonomy in assistive robotics. *IEEE Robotics and Automation Letters*, 2(1):247–254, 2017.

[14] Laura V. Herlant, Rachel M. Holladay, and Siddhartha S. Srinivasa. Assistive teleoperation of robot arms via automatic time-optimal mode switching. In *Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 2016.

[15] Rachel M. Holladay, Anca D. Dragan, and Siddhartha S. Srinivasa. Legible robot pointing. In *The IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, 2014.

[16] Lauren M. Miller and Todd D. Murphey. Trajectory optimization for continuous ergodic exploration. In *American Control Conference (ACC)*, 2013.

[17] Lauren M. Miller, Yonatan Silverman, Malcolm A. MacIver, and Todd D. Murphey. Ergodic exploration of distributed information. *IEEE Transactions on Robotics*, 32(1):36–52, 2016.

[18] Stephen Monsell. Task switching. *Trends in cognitive sciences*, 7(3):134–140, 2003.

[19] Marnix Nuttin, Dirk Vanhooydonck, Eric Demeester, and Hendrik Van Brussel. Selection of suitable human-robot interaction techniques for intelligent wheelchairs. In *Proceedings of the IEEE International Workshop on Robot and Human Interactive Communication*, 2002.

[20] Stephanie Rosenthal, Joydeep Biswas, and Manuela Veloso. An effective personal mobile robot agent through symbiotic human-robot interaction. In *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems*, 2010.

[21] Alexander Sorokin, Dmitry Berenson, Siddhartha S. Srinivasa, and Martial Hebert. People helping robots helping people: Crowdsourcing for grasping novel objects. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2010.

[22] Alireza Tavakkoli, Richard Kelley, Christopher King, Mircea Nicolescu, Monica Nicolescu, and George Bebis. A vision-based architecture for intent recognition. *Advances in Visual Computing*, pages 173–182, 2007.

[23] Katherine Tsui, Holly Yanco, David Kontak, and Linda Beliveau. Development and evaluation of a flexible interface for a wheelchair mounted robotic arm. In *Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 2008.

[24] Ivan Volosyak, Oleg Ivlev, and Axel Graser. Rehabilitation robot FRIEND II-the general concept and current implementation. In *Proceedings of the International Conference on Rehabilitation Robotics (ICORR)*, 2005.

[25] Glenn Wasson, Pradip Sheth, Majd Alwan, Kevin Granata, Alexandre Ledoux, and Cunjun Huang. User intent in a shared control framework for pedestrian mobility aids. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2003.