■ Learning Assistance by Demonstration: Smart Mobility With Shared Control and Paired Haptic Controllers

Harold Soh

Singapore-MIT Alliance for Research and Technology (SMART)

and

Yiannis Demiris

Imperial College London

In this paper, we present a framework, probabilistic model, and algorithm for learning shared control policies by observing an assistant. This is a methodology we refer to as Learning Assistance by Demonstration (LAD). As a subset of robot Learning by Demonstration (LbD), LAD focuses on the assistive element by explicitly capturing how and when to help. The latter is especially important in assistive scenarios—such as rehabilitation and training—where there exists multiple and possibly conflicting goals. We formalize these notions in a probabilistic model and develop an efficient online mixture of experts (OME) algorithm, based on sparse Gaussian processes (GPs), for learning the assistive policy. Focusing on smart mobility, we couple the LAD methodology with a novel paired-haptic-controllers setup for helping smart wheelchair users navigate their environment. Experimental results with 15 able-bodied participants demonstrate that our learned shared control policy improved driving performance (as measured in lap seconds) by 43 s (a speedup of 191%). Furthermore, survey results indicate that the participants not only performed better quantitatively, but also qualitatively felt the model assistance helped them complete the task.

Keywords: shared control, learning by demonstration, assistive robotics, Gaussian processes, haptic control, smart mobility

1. Introduction

Providing proper assistance in a shared-control setting is a complex endeavor. By its nature, assistance is multi-faceted and dependent on a variety of factors including the task, the user's state, the environment, and the assistant's capabilities. Moreover, competing objectives may require subtleties with regard to the type and amount of assistance offered. For example, in educational and rehabilitation settings, the primary objective is long-term development rather than short-term task completion. These aspects have ramifications for the development of intelligent robotic assistants, such as smart wheelchairs that help users navigate their environment (Carlson & Demiris, 2008, 2010; Gomi & Griffith, 1998; Levine et al., 1999; Miller & Slack, 1995; Simpson et al., 2004; Soh & Demiris, 2013; Tsotsos et al., 1998; Yanco, 1998b).

Authors retain copyright and grant the Journal of Human-Robot Interaction right of first publication with the work simultaneously licensed under a Creative Commons Attribution License that allows others to share the work with an acknowledgement of the work's authorship and initial publication in this journal.

As a step toward resolving this challenging problem, we propose Learning Assistance by Demonstration (LAD), whereby we extract shared control policies by observing an assistant. This paper extends preliminary work (Soh & Demiris, 2013) with a mixture of experts model, experiments with paired haptic controllers, and a subsequent analysis of results.

In contrast to current task-centered LbD systems, LAD focuses on the assistive element. For example, to teach a smart wheelchair to navigate, existing LbD systems advocate providing driving demonstrations to the robot (Chow & Xu, 2006). However, in shared control scenarios, a smart wheelchair faces the difficult decisions of not only *how* best to assist but also *when* it should intercede (Demiris, 2009). Existing solutions primarily attempt to infer the user’s intent (Carlson & Demiris, 2008; Demiris, 2007; Vanhooydonck et al., 2010) using learned policies, failure models, and/or perspective-taking. Unfortunately, such approaches can be computationally demanding and difficult for system designers to tune, particularly when competing goals are present. In contrast, LAD is a direct approach; instead of deriving a “how to drive” policy, LAD extracts a policy for “how to help a user drive.”

This framework inherits many of the benefits accompanying LbD (Argall et al., 2009; Dillmann, 2004). Providing training examples is often more intuitive than hand-coding specific behaviors, particularly for complex tasks. As such, LAD enables rapid-prototyping and allows non-roboticists (such as occupational therapists) to participate in policy development. LAD also admits the use of robots incapable of solo task completion; to be an effective assistant, the robot need only be capable of providing contextual help when required. In other words, our approach recognizes the difference between conducting full open-heart surgery and handing a surgeon the proper surgical tool.

A key feature of the LAD model is that it explicitly captures both when and how to assist a user at a given task. Experts such as occupational therapists may have large amounts of tacit knowledge that can be difficult to formalize as control algorithms. As such, we develop a novel probabilistic online mixture of experts (OME) model to learn shared control policies from demonstration. At an algorithmic-level, the OME builds upon previous work (Soh & Demiris, 2013) and is composed of multiple online infinite echo-state Gaussian process (OIESGP) (Soh & Demiris, 2014b) sub-models. The method learns (and predicts) in an online manner (processing a sample or mini-batches at a time) on sequential sensory data. Thus, compared to existing work, the LAD model can be trained *during* the demonstration process, which facilitates iterative teaching.

To demonstrate the effectiveness of our approach, we focus on the problem of shared control for smart assisted mobility. In particular, we assume a specific limitation on the part of the wheelchair user and train the model to provide contextual assistance. Experiments involving both a simulated robot and a real-world smart wheelchair with human participants are presented. Inspired by the hand-over-hand approach used by occupational therapists, we introduce and implement a novel training instrument using paired haptic controllers. Our results show that the LAD OME model attains high prediction accuracy (AUC of 0.95) after only a single demonstration, which translates into improved task-performance; our LAD-enabled wheelchair decreased participant lap-driving times by 43.4 s (corresponding to a speedup of 191%).

The remainder of this paper is structured as follows: Background information on related work is given in Section 2; in Section 3, we define the LAD approach with an illustrative example; Section 4 describes our online LAD model, giving specifics on the base OIESGP learners and how the OME model is generated iteratively; Section 5 describes our simulation experiment and presents model accuracy results; and Section 6 extends this discussion to a real-world experiment on the ARTY smart wheelchair platform (Soh & Demiris, 2012) with the paired haptic controllers and human participants. We present both objective (lap-timings) and subjective results obtained from a participant survey. Finally, Section 7 concludes our paper with a summary and presents research questions arising from this work.

2. Related Work

While the LAD framework could be applied in different contexts, the focus of this paper is on providing smart mobility assistance for people with disabilities. Intelligent assisted mobility could benefit 61–91% of all wheelchair users (Simpson et al., 2008) and to meet this need, the research community has developed a variety of smart-wheelchair platforms; early prototypes include Tin Man II (Miller & Slack, 1995), Wheelesley (Yanco, 1998a,b), the TAO wheelchairs (Gomi & Griffith, 1998), Playbot (Tsotsos et al., 1998), and NavChair (Levine et al., 1999). For a more thorough discussion of smart wheelchairs platforms, readers are referred to (Simpson, 2005).

A significant amount of recent work has focused on deriving shared-control methods that provide safe navigation while ensuring that users remain active users (e.g., Carlson & Demiris, 2012; Parikh et al., 2007). For example, (Goil et al., 2013) “blended” planner-generated assistive and user controls using the task variance, which was learned offline via a Gaussian Mixture Model (GMM); thus, the task variance functioned as a heuristic for determining the degree of assistance. Instead of planner-based control, non-parametric machine-learning methods can also be trained to generate assistive control signals (e.g., the artificial neural network [ANN] used by Nguyen et al., 2013).

One particularly notable work was performed by (Vanhooydonck et al., 2010) using the Sharioto smart wheelchair to investigate adaptive intelligent assistance. The proposed framework estimated user intention with an implicit user model (represented by an ANN) and adapted user control signals only when the system determined that assistance was required. The system was successful both in simulation and during real-world experiments, but a key difficulty was user model training. The user was required to make several training and test runs on a standard course in order to correctly estimate intent and infer when assistance was required. Moreover, the system made use of deviations from an assumed “ideal path,” which the authors acknowledge can be difficult to define. In general, learning and determining intent and when to assist are inconvenient aspects of current algorithms.

By using the LbD framework, our LAD methodology side-steps these problems by leveraging on demonstrator’s capabilities to infer intent and decide when to provide assistance. Although LbD has previously been applied toward controlling smart wheelchairs (Chow & Xu, 2006), the focus was on transferring (expert) human *driving* skills. In contrast, we are interested in transferring *assistive* skills for shared control. This permits a degree of flexibility to the training environment since there is no longer a need for the algorithm designer to specify intentions or ideal paths. We argue that this permits a clearer and more intuitive separation of tasks (i.e., for the robot to learn the assistive policy from an expert such as an occupational therapist).

A related strand of research in a different context is in learning collaborative tasks by demonstration (Calinon et al., 2009; Chernova & Veloso, 2008; Rozo et al., 2013). In particular, (Calinon et al., 2009) used hidden-Markov models (HMMs), Gaussian Mixture Regression (GMR), and haptic devices in a collaborative lifting task (lifting a beam and keeping it horizontal). Interestingly, the system learned not only leading and following behaviors, but also when to switch between the two. The key differences in our work, other than the intended application, are as follows: (i) assistance is not provided continuously but only at key points during execution and (ii) learning is performed using a mixture of sparse GPs, permitting fast, online training.

3. Learning Assistance by Demonstration: Problem Statement

The basic concept behind the LAD methodology is to learn an assistive (shared control) policy by observing an “expert assistant” (the demonstrator). As a guide, a system overview is illustrated in Fig. 1.

As a starting point, let us consider a specific intelligent wheelchair training scenario where an occupational therapist (U_A) is teaching a young child (U_P) with special needs to navigate an envi-

Figure 1. Learning-Assistance-by-Demonstration (LAD) System. Our model learns both *when* and *how* to assist iteratively from an assistant (demonstrator) who is helping a user accomplish a task. In this paper, we focus on extracting a shared-control policy using a paired-haptic controller setup with smart wheelchairs (Adapted from Soh & Demiris, 2013).

ronment using a smart wheelchair (R) (Soh & Demiris, 2013). At the same time, U_A wishes to train the smart wheelchair R to help the child when U_A is not present or engaged in another task. When U_A notices that U_P is unable to complete a particular sub-task, U_A engages an “assistance-mode” where U_A guides U_P appropriately with actions $\hat{\mathbf{a}}_t$ —this can be seen as the hand-over-hand assistance method (i.e., holding a child’s hand to demonstrate how the task should be completed). In one sense, U_A and U_P are sharing control of the wheelchair. R is able to observe these actions as well as states $\mathbf{x}_t$ when interventions occur and aims to extract a shared control policy $\pi_A(\mathbf{x}_t)$.

In general, we call U_A an assistant who is engaged in an *assistive task* (T_A) (i.e., helping a primary user U_P complete a primary task [T_P]). The robot, R , has to derive an assistive policy—a mapping $\pi_A(\mathbf{x}_t) : \mathcal{X} \rightarrow \mathcal{A}$ from states $\mathbf{x}_t \in \mathcal{X}$ to assistive actions $\mathbf{a}_t \in \mathcal{A}$. In this work, the system state comprises three basic elements $\mathbf{x}_t = (\mathbf{x}_{A,t}, \mathbf{x}_{U,t}, \mathbf{x}_{E,t})$ where $\mathbf{x}_{A,t}$, $\mathbf{x}_{U,t}$, $\mathbf{x}_{E,t}$ are the state of the assistant, the user, and the environment. In addition to user command velocities, the primary user state may contain other factors relevant to when and how to assist, such as user’s personality traits, physical/cognitive skills, biometric data (Urdiales et al., 2010), and driving capability. The actions performed by U_P and U_A , denoted as $\hat{\mathbf{u}}_t \in \mathcal{U}$ and $\hat{\mathbf{a}}_t \in \mathcal{A}$, respectively, are also observable and can be modeled as part of the associated state vectors. Note that LAD is distinct from typical LbD in that the latter seeks a primary task policy $\pi_P(\mathbf{x}_t) : \mathcal{X} \rightarrow \mathcal{U}$. In contrast, LAD seeks an assistive policy.

In this work, we assume a visible binary signal ($\hat{h}_t$), indicating when the assistance is provided. This signal plays an important role, because an essential element of our approach is the recognition that assistance may only be required at appropriate points during task execution. Constant inter-

vention can interfere with task completion or competing objectives. Using our wheelchair-training scenario above, the child may become frustrated with the activity if U_A does not provide sufficient assistance. On the other hand, over-assistance may lead to *learned helplessness* (Seligman, 1972) where the child learns to rely on assistance, which negatively impacts long-term development. As such, the central piece of the LAD system (Fig. 1) is the Online Probabilistic Assistive Modeling component, which consists of two sub-models: a “When-to-Help” model and a “How-to-Help” model.

4. An Online Probabilistic Model for Learning Shared Control

We adopt a probabilistic approach for LAD that models an assistant making two decisions at each time-step: whether help should be offered, and if so, what should be done. The assistive actions are continuous random variables $\mathbf{a}_t \in \mathbb{R}^D$, which, for example, can represent motor commands to an actuator. We also define a “null” or zero action $\mathbf{a}_0 \triangleq \mathbf{0}$ where no help is offered.

The model consists of two components: an “assistance component” and a “do-nothing component,” which generate assistive actions $\mathbf{a}_t$:

$$p(\mathbf{a}_t | \mathbf{x}_t, f) = \underbrace{p(h_t = 1 | \mathbf{x}_t) \mathcal{N}(\mathbf{a}_t | f(\mathbf{x}_t), \mathbb{V}[f(\mathbf{x}_t)])}_{\text{Assist}} + \overbrace{(1 - p(h_t = 1 | \mathbf{x}_t)) \delta_0(\mathbf{a}_t)}^{\text{Do not assist}}. \quad (1)$$

where δ_0 is the Dirac delta function centered at $\mathbf{a}_0$. In other words, the assistive actions are “drawn” from a two-component mixture model comprising the situations, where (i) the assistant intervenes, producing an assistive action using a latent function $f(\mathbf{x}_t)$, or (ii) does not assist, producing the null action $\mathbf{a}_0$ (allowing U_P to solve the sub-task on his own). As such, the discrete/binary random variable $h_t \in \{-1, 1\}$ —conditioned on the current state $\mathbf{x}_t$ —plays an important role: it determines when assistance is or should be provided.

In this paper, h_t and $\mathbf{a}_t$ are obtained using a GP binary discriminative classifier (GPC) and a GP regressor (GPR), respectively. These flexible non-parametric models are trained using the observed state $\mathbf{x}_t$ and assistive signals/actions ($\hat{h}_t$ and $\hat{\mathbf{a}}_t$). Unfortunately, a “full” training of the model using the standard GP formulation is computationally prohibitive, and we make use of two approximations: (i) We train the GP and GPC independently, and (ii) we use the OIESGP, a sparse iterative GP for spatio-temporal signals (Soh & Demiris, 2014b). Using the OIESGP as a basic unit, we introduce an online mixture model that improves modeling power while ensuring a low computational overhead.

4.1 Online Infinite Echo-State Gaussian Process

This section presents a brief overview of the OIESGP (Soh & Demiris, 2014b), a recently proposed online learner for multi-variate time series. For conciseness, we restrict our discussion to the main aspects of the OIESGP as relevant to the current study and refer readers to (Soh & Demiris, 2014b) for more details. The OIESGP combines a sparse online Gaussian Process (SOGP) (Csató & Opper, 2002) with a recursive automatic relevance determination (ReARD) kernel. Additionally, it uses stochastic natural gradient descent (SNGD) (Amari, 1998) for hyperparameter adaptation.

Unlike standard GPs, the OIESGP operates in an online manner (i.e., processing a sample or mini-batches at a time) on sequential data. This aspect is particularly relevant for on-demand, real-world applications, such as robotics. In terms of computational costs, the method uses a fixed maximum computational and storage overhead that can be defined based on available resources; this is achieved by using a set of basis vectors (BVs), $\mathcal{B}$, or inducing input locations (Quiñonero Candela & Rasmussen, 2005).

Figure 2. Kernel clustering using Euclidean centers produced a linear and incorrect partitioning of the space, which was resolved by using the kernel centers in the projected space.

The ReARD kernel belongs to the class of recursive kernels (Hermans & Schrauwen, 2011) and incorporates automatic relevance detection (ARD) (Neal, 1996):

$$\kappa_t^{\text{ARD}}(\mathbf{x}, \mathbf{x}') = \underbrace{\exp\left(-\frac{1}{2}(\mathbf{x}_t - \mathbf{x}_t')^T \mathbf{M}(\mathbf{x}_t - \mathbf{x}_t')\right)}_{\text{"Spatial Similarity"}} \overbrace{\exp\left(\frac{\kappa_{t-1}^{\text{ARD}}(\mathbf{x}, \mathbf{x}') - 1}{\sigma_\rho^2}\right)}^{\text{"Temporal Similarity"}} \quad (2)$$

where $\mathbf{M} = \text{diag}(\mathbf{l})^{-2}$, a diagonal $d \times d$ matrix where $\mathbf{l} = [l_i]_{i=1}^d$. The ReARD kernel is anisotropic in that the kernel function’s responsiveness to input dimension k is inversely related to l_k . The parameter σ_ρ can be regarded as a “temporal lengthscale” as it weights the past—the larger σ_ρ is, the less relevant the past recursive kernel value will be. In the OIESGP, these parameters are optimized in an online manner, which simplifies hyperparameter specification while post-learning inspection can potentially identify feature relevance.

Although demonstrated to be effective with a variety of tasks (Soh & Demiris, 2014a,b), the OIESGP (with hyperparameter optimization) requires periodic kernel matrix inversions of order $O(|\mathcal{B}|^3)$. When the basis vector set $\mathcal{B}$ is allowed to grow large (to increase modeling power), the inversions can induce a pronounced slowdown in the algorithm. We circumvent this issue by introducing a ‘mixture of experts’ model, where each OIESGP expert maintains a small $\mathcal{B}$.

4.2 Mixtures of OIESGP Experts

Mixture of experts (ME) models (Jordan & Jacobs, 1994) consider that the probability of the observed output comes from a linear mixing of K base experts:

$$p(y_t | \mathbf{x}_t, \boldsymbol{\Theta}) = \sum_{i=1}^K \pi_k p_k(y_t | \mathbf{x}_t, \boldsymbol{\theta}_k) \quad (3)$$

where Θ represent all the model parameters, θ_k are the parameters for the k -th GP, and π_k are the mixing weights, $\pi_k = p(z_t = k \mid \mathbf{x}_t, \zeta)$ where ζ are the *gating function* parameters and z_t are the indicator variables assigning samples to experts.

We present an online approximation using OIESGP experts where we apply online kernel k-means to partition the space. For an *offline* LAD mixture model, we may instead apply the GP mixtures proposed in (Rasmussen & Ghahramani, 2002) and (Tresp, 2001). Our online approach differs from LWPR (Vijayakumar et al., 2005) and LGP (Nguyen-Tuong et al., 2009), which can produce pathological clusterings due to the use of Euclidean cluster centers. Fig. 2 shows a concrete example of this, where we applied online clustering on data generated from a simple switching dynamical system (with two modes of different radii and switching between the modes every 50 time-steps). Here, kernel clustering using the *centers in the projected space* generated a more accurate and intuitively appealing partitioning of the space, given the underlying system.

4.2.1 Kernel k-means Clustering The widely-used k-means algorithm is an iterative method that clusters samples $\mathbf{x}_i$ into K clusters to minimize the loss function,

$$\mathcal{L}(M, Z) = \sum_i \|\mathbf{x}_i - \mathbf{m}_{z_i}\|^2 \quad (4)$$

where $M = \{\mathbf{m}_k\}$ for $k = 1, \dots, K$ is the set of the cluster centers and $Z = \{z_i\}$ is the set of assignment variables indicating to which cluster the sample $\mathbf{x}_i$ belongs. The solution to this optimization problem is achieved by iteration of two steps:

1. Assign the points to the closest cluster: $z_i = \arg \min_k \|\mathbf{x}_i - \mathbf{m}_k\|^2$
2. Re-compute the cluster centers: $\mathbf{m}_k = \sum_{\mathbf{x}_i \in \mathcal{C}_k} \mathbf{x}_i / |\mathcal{C}_k|$, where $\mathcal{C}_k$ is the set of samples assigned to cluster k .

In kernel k-means, the overall structure of the algorithm remains unchanged. The difference is in the distance computation; instead of using the Euclidean distance in input space, we find the distance to the cluster center in projected space:

$$d(\mathbf{x}_i, \mathbf{m}_k^\phi) = \|\phi(\mathbf{x}_i) - \mathbf{m}_k^\phi\|^2 \quad (5)$$

where the cluster center is given by:

$$\mathbf{m}_k^\phi = \frac{\sum_{\mathbf{x}_i \in \mathcal{C}_k} \phi(\mathbf{x}_i)}{|\mathcal{C}_k|}. \quad (6)$$

Using the property of Mercer kernels, we can apply kernel evaluations instead of dot products:

$$d(\mathbf{x}_i, \mathbf{m}_k^\phi) = k(\mathbf{x}_i, \mathbf{x}_i) - \frac{2 \sum_{\mathbf{x}_j \in \mathcal{C}_k} k(\mathbf{x}_i, \mathbf{x}_j)}{|\mathcal{C}_k|} + \frac{\sum_{\mathbf{x}_j, \mathbf{x}_l \in \mathcal{C}_k} k(\mathbf{x}_j, \mathbf{x}_l)}{|\mathcal{C}_k|^2} \quad (7)$$

4.2.2 Online Clustering for OIESGP experts When data arrives sequentially, we can approximate the full kernel k-means solution in a greedy fashion to partition the space incrementally. Here, we process samples $\mathbf{x}_t$ one at a time to update a set of OIESGP experts. Given the maximum number of experts $K_{\max}$, the model experts m_k for $k = 1, \dots, K_t$ where $K_t \leq K_{\max}$ and where $K_{\max}$ is user-defined. Each expert has a basis vector set $\mathcal{B}_k$. Similar to LWPR and LGP, we employ a cluster creation threshold w_{gen} to determine when a cluster should be spawned; however, the number of experts is bounded by $K_{\max}$. Updating the model proceeds as follows:

1. Compute the distances, $d_{k,t}$, to each of the k centers via Eq. 7 using $\mathcal{B}_k$ in place of $\mathcal{C}_k$.
2. If $\exp(-d_{k,t}) < w_{\text{gen}}$ for all k and $K_t \leq K_{\max}$, we create a new expert with the sample $\mathbf{x}_t$, increment K_t , and proceed to the next sample.

3. Otherwise, we assign $\mathbf{x}_t$ to the closest expert (i.e., $z_t = \arg \min_k d_{k,t}$ and update the OIESGP m_k with the sample).

Although the distance computation in Eq. 7 appears to be on the order of $O(|\mathcal{B}_k|^2)$, it is straightforward to maintain and update the last term on the RHS for each cluster (on the order of $O(|\mathcal{B}_k|)$ for each sample). Then, calculating the distance to each cluster takes $O(|\mathcal{B}_k|)$ for each expert, and hence, $O(K_{\max}|\mathcal{B}_k|)$ in total. In summary, the total distance computation time grows linearly with the total number of basis vectors stored across all experts and is bound by the total number of stored BVs.

Training can be performed online as the demonstration is being conducted. In our experiments, the When-to-Help model is continuously trained throughout the demonstration with both positive and negative observed samples $(\mathbf{x}_t, \hat{h}_t)$. The How-to-Help model is only trained when assistance is offered ($\hat{h}_t = 1$) with the observed sample pairs $(\mathbf{x}_t, \hat{\mathbf{a}}_t)$. Note that both models are trained independently in that the likelihood of h_t is not considered when training the regressor; this allows for fast online training at the expense of bias.

4.3 Shared Control Policy

We can apply the trained models directly in a shared control policy $\pi_A(\mathbf{x}_t)$. To make a prediction, we used a simple winner-takes-all scheme:

$$\mu_t = \sum_k \delta_{k,z_t} \mu_{k,t} \quad (8)$$

where $\mu_{k,t}$ is the predicted mean of expert k using input $\mathbf{x}_t$, δ_{k,z_t} is the Kronecker delta, and $z_t = \arg \min_k d_{k,t}$. Preliminary tests indicated that this approach worked effectively for both the How-to-Help regressor and When-to-Help classifier (Soh & Demiris, 2013). For assistive control, we applied a threshold W_h and assistance ($\hat{h}_t = 1$) was offered whenever the probability of assistance $p(h_t|\mathbf{x}_t) \geq W_h$. The threshold W_h is a user-defined parameter that controls the minimum level of confidence before offering assistance. The assistive control signals, $\tilde{\mathbf{a}}_t$, are taken as the prediction of the OIESGP experts using Eq. 8 above. More complex policies can be derived to make use of the assistive action distribution variance, and this will be explored in future work.

5. Simulation Experiment: Smart Mobility

Our simulation study was designed as a feasibility test to determine if that the proposed model was capable of learning from a set of demonstrations to provide assistance at correct moments during execution. Our first-cut scenario is simple but inspired from preliminary wheelchair trails with a young child having difficulty controlling his powered wheelchair during turns. Here, the wheelchair user is assumed to lack the fine motor control skill required to make sharp right turns, motivating the need for assistance to prevent user frustration. To encourage the development of wheelchair driving skills, no assistance was given during soft right turns (where the desired turning angle was less than 60 degrees), left turns or translational (forward/backward) movements. The straightforward nature of this scenario allowed us to more easily and precisely analyze what kind of assistance was offered and if it was given at the correct time. The performance of our model was evaluated using the model accuracy and the lap time (in seconds).

5.1 Experimental Setup

Our simulation was developed using the Robot Operating System (ROS) (Quigley et al., 2009) and the Stage simulator (Gerkey et al., 2003). Our “test subject” (i.e., primary user) was an autonomous differential-drive robot driver (RD) driving in the environment shown in Fig. 3a. We used the ROS

Figure 3. (a) Driving course used in our simulation and preliminary trials. Our simulated robot tasked to drive from S to F, passing through and making two sharp right turns at R₁ and R₂ (Adapted from Soh & Demiris, 2013). (b) Driving course used in our haptic-control LAD experiment. Participants were tasked to drive from S to F, passing through and making four sharp right turns, labeled R₁ through R₄. The demonstrator was located at D.

navigation stack¹ to develop RD’s basic behavior: it drove laps starting at S, through R₁ and R₂, and finishing back at F. RD avoided obstacles using the Dynamic Window Approach (DWA) (Fox et al., 1997) and drove around the track with maximum forward and rotational velocities of 0.7 m/s and 0.5 rad/s respectively. For the purposes of this test, RD always made right turns at R₁ and R₂. With this setup, RD completed a lap in 44.0 s on average (the baseline). We simulated the right-turn impairment by scaling right turns to a maximum of 0.15 rad/s—this negatively impacted RD’s average lap performance, increasing lap times to 61.6 s.

5.2 Assistance Demonstration

Demonstrations were provided by a human assistant using a joystick controller. Assistance, in the form of a control takeover, was only given during right turns at R₁ and R₂ and withheld during all other times during the lap run. As stated before, the learning algorithm was able to observe the assistance signal h_t and the assistive command velocities $\hat{\mathbf{a}}_t = (\hat{a}_{x,t}, \hat{a}_{\theta,t})$. With demonstrator assistance, RD was able to complete the lap in approximately 46.1 s, only slightly higher than the baseline (without impairment).

5.3 Model Setup and Parameters

As stated in Section 3, the system state is a tuple consisting of the primary user, assistant, and environmental states, $\mathbf{x}_t = (\mathbf{x}_U, \mathbf{x}_A, \mathbf{x}_E)$. In this experiment, we represented RD’s state by its current and desired (command) velocities, $\mathbf{x}_U = (v_{x,t}, v_{\theta,t}, u_{x,t}, u_{\theta,t})$. The command velocities served as proxies for intent, which is a latent internal state. Note that when the simulated impairment was applied, the model only observed *scaled* right turns (0.15 rad/s) instead of the full desired rotational velocity of 0.5 rad/s—this complicates learning and prediction since soft and hard right turns are more difficult to distinguish.

¹See <http://www.ros.org/wiki/navigation> for more information.

Table 1: Assistive Model Parameters. The same parameters were used for the simulation and the real-world trials.

Parameters	When-To-Help	How-To-Help
Capacity (C)	50	50
Recursion Depth (τ)	10	10
Temporal scale (σ_ρ^2)	1.01	1.01
Lengthscale (l_U)	1.0	0.1
Lengthscale (l_A)	1.0	0.1
Lengthscale (l_E)	2.0	2.0
Signal Variance (σ_f^2)	0.1	0.05
Noise (σ_n^2)	0.1	0.01
Assistance Threshold (W_h)	0.5	—
Expert Creation Threshold (w_{gen})	0.4	0.3
Maximum Number of Experts (K_{max})	50	50

To simplify our state representation, we drop $\mathbf{x}_A$ since both U_A and U_P are the same object in this simulation and assistant-specific features were not relevant for this specific scenario. As the environmental state, we used forward laser scan readings $\mathbf{x}_E = \mathbf{s}_t = (s_{1,t}, s_{2,t}, \dots, s_{M,t})$ where $M = 15$ segments. The benefit of using laser scan readings instead of localized positions was that the latter could potentially limit model generalizability, since assistance would be tied to particular map coordinates instead of observed environmental features.

Although this simple representation was sufficient for this experiment, our model is capable of accommodating more complex states. For example, the state of the user can include parameters concerning the user’s current skill-level, disability, and personality scores. As such, it may be possible to use models trained on one user with another user with similar traits, and for the model to “interpolate” user traits—future experiments could be designed to validate this aspect.

Finally, the parameters for the probabilistic model, including the GPC and GPR parameters, are given in Table 1. To investigate the differences between the OME and when only one OIESGP regressor and classifier is used (as was the condition in Soh & Demiris, 2013), we ran additional trials with $K_{\text{max}} = 1$.

5.4 Simulation Results

Ten supervisor demonstrations were recorded and combined into a single dataset to train and test our model. Each “run” consisted of two phases. In the first phase, training was conducted in an online fashion (i.e., the model was trained/tested at each time step for nine demonstrations). In the second “test” phase, the last remaining trial was used for testing only (without training). Each run was repeated fifteen times with the ten demonstrations randomly permuted.

Performance results are shown in Fig. 4, which summarizes the AUC and the RMSE scores. The AUC scores (Fig. 4a) for our trained model were high (0.97–0.99), indicating strong classifier performance; in particular, high true positive to false positive ratios. We also see that a general increasing trend as more demonstrations were provided—the median AUC scores were 0.990 by the end of the first phase and 0.993 for the final test trial. Finally, the AUC scores were higher for the OME model than for the single OIESGP model (0.985 for the test phase)—the number of experts “grown” for the classifier varied between 9 and 11 depending on the trial.

In terms of the assistive control accuracy, Figs. 4b and 4c show that our model achieved low

Figure 4. Performance measures summarizing LAD model accuracy. The When-to-Help model achieved a high AUC score (0.97–0.99), which progressively improved as more samples were provided. This was higher than the single-expert model (0.95–0.98), showing the advantage of the OME model. In addition, the How-to-Help model attains relatively low (≈ 0.07) errors for the turning command velocities (Adapted from Soh & Demiris, 2013).

Figure 5. Lap times in seconds. The control-limited simulated driver (RD) completed a lap in 61.6 s. With human demonstrator assistance, RD’s lap time was 46 s. Our model was similarly able to improve lap times to 46.4 s (corresponding to a speedup of 133%).

RMS errors and generated assistive actions similar to the demonstrated controls. We note that the rotational command velocities were larger than the linear controls since the help was a right turn and therefore principally rotational. Nonetheless, we see a falling trend in the error as more demonstrations were provided to the model. Only one expert was generated during the online clustering, and thus, the RMSE scores between the OME and single OIESGP were very similar. Finally, Fig. 4c reveals an interesting aspect of the model: It appears to “anticipate” when assistance should be offered. In the seconds leading up to the assistance, we observe a rise in the model’s predicted $p(h_t)$ and a decrease in the variance $\mathbb{V}[f(\mathbf{x}_t)]$.

Fig. 5 shows RD’s lap times for the baseline (normal driving), the limited-control scenario (with simulated disability), assistance by the human demonstrator, and assistance by our trained LAD model. After only a single demonstration, RD’s lap time improved to 46.5 s (a speedup of 133% over the limited control scenario), which is similar to the human demonstrator’s performance (Kolmogorov-Smirnov test with $p = 0.67$, $k_s = 0.30$).

5.4.1 Feature Relevance The lengthscales obtained after the training laps (Fig. 6) show an indication of feature relevance. The When-to-Help component exhibited more change in lengthscales values, and the most relevant features (with the largest decrease) were the wheelchair velocities and the front-right laser reading. This was surprising since we expected the user input to have the most relevance. We posit that since the classifier was only able to observe the scaled command values, the wheelchair velocities (over time, with the laser scan values) were found more informative of when to start and stop assistance.

5.4.2 Computational Times The How-to-Help component took an average of 0.021 s ($SD = 0.0072$ s) per iteration, with a maximum of 0.0695 s when the GP was being rebuilt using new hyperparameters. The When-To-Help classifier’s processing time was similar: an average of 0.0128 s ($SD = 0.0145$) per iteration, with a peak 0.0730 s. As such, our LAD model was capable of operation at ≈ 10 Hz. Note that using the mixture of 10 experts (with a maximum of 50 BVs each) was computationally cheaper than using a single model with 500 BVs. On a separate test, the single model variant with $|\mathcal{B}| = 500$ required approximately 2.5 s per iteration.

Post-Learning Lengthscales for the LAD Simulation Experiment

Figure 6. Post-learning lengthscales for LAD simulation experiment. The When-to-Help component manifested more change in lengthscales values, and features with the largest decrease (over the starting values) were the wheelchair velocities and the front laser-right reading.

6. Real-World Experiment with ARTY and Haptic Controllers

This section describes results of a real-world experiment using the Assistive Robotic Transport for Youngsters (ARTY) smart wheelchair platform (Fig. 7) (Soh & Demiris, 2012) with 15 human participants and paired haptic controllers. The closest related work (Marchal-Crespo et al., 2010) evaluated a robotic wheelchair trainer using a high-grade haptic device and found that the trainer enhanced motor learning by able-bodied children as well as a child with cerebral palsy. Here, we propose paired haptic controllers that allow both the trainer and user to feel each other's actions. In effect, both trainer and user are participating in a shared control session during the demonstration (while simultaneously training the wheelchair to assist). In this section, we describe our experimental platform, setup, and obtained findings.

6.1 Robot Platform: ARTY Smart Wheelchair

Our experimental platform was ARTY, which was designed to widen access to independent mobility for children with disabilities. The base model of ARTY is the Skippi, an electronic powered indoor/outdoor chair (EPIOC) that fulfilled several basic requirements (as set out in Orpwood et al., 2005, i.e., it was colorful, worked both indoors and outdoors with a good range [30 km], was easily transportable with adjustable seats, and had high-capacity batteries that lasted more than a day). Furthermore, interfacing with the controller-area network (CAN) system² used in the Skippi

²CAN is a message-based protocol originally designed for automotive systems. Currently, it is used in a wide range of devices from EPIOCS to humanoid robots, such as the iCub (Metta et al., 2008).

Figure 7. The Assistive Robotic Transport for Youngsters (ARTY) Smart Wheelchair (Adapted from Soh & Demiris, 2012).

permitted electronic control of the wheelchair.

Processing on ARTY is performed on a distributed system comprising two parts: (i) the “lower-level” Atom-powered mini-PC that integrates sensory information and (ii) the tablet PC that performs higher-level path planning and obstacle avoidance. As such, ARTY’s sensors—three Hokuyo URG-04LX laser scanners, five sonars, and bump sensors—are connected directly to the mini-PC. By splitting the processing tasks, we were able to improve response-time performance and accommodate future expansion. Furthermore, the tablet PC allowed users to change the wheelchair’s basic parameters via a natural touch-based interface.

ARTY’s software system (high-level schematic as shown in Fig. 8) was developed using the Robot Operating System (ROS) (Quigley et al., 2009), and hence, is composed of interacting ROS nodes. For example, each laser sensor is managed by its own ROS node that publishes range information as messages; these range messages are later combined into a coherent obstacle map.

The central ROS node is the shared-control system (SCS). The SCS is responsible for producing appropriate control messages, taking into account the user’s control command messages (obtained via the joystick or haptic device nodes), sensory information, and the LAD signals. Additionally, it employs a limited form of DWA³ to slow the wheelchair (by scaling user-commanded velocities) to prevent hard collisions with obstacles. This safeguarding mechanism prevented damage to the wheelchair and potential injury to participants. Finally, the control messages generated by the SCS are read by the motor access/control (MAC) node that translates ROS velocity commands into CAN messages understood by the Skippi electronic system.

6.1.1 Paired Haptic Controllers The haptic controller used in this study was the low-cost⁴ Novint Falcon, a three-DOF joystick with 400 dpi (≈ 0.06 mm) position resolution (Fig. 9). The Falcon is capable of providing a maximum force of approximately 9 N, via three armed motors, updated at 1 kHz (Novint Technologies, 2013). In comparison, the professional-grade ForceDimension Delta 6

³DWA has been used extensively in robot navigation, with collaborative adaptations made for use in smart wheelchairs, e.g., Sharioto (Demeester et al., 2007) and the Bremen Wheelchair Roland III (Rofer, 1998).

⁴Recommended retail price of $\approx$ USD 250.00.

Figure 8. High-level schematic of ARTY's Software Components (ROS Nodes) (Adapted from Soh & Demiris, 2012). In addition to the sensor and wheelchair control nodes, the system accommodates LAD via the demonstrator and user input nodes for alternative inputs such as haptic devices.

controller provides 20 N of force with < 0.01 mm resolution, but costs approximately USD 35,000.

We used two Falcon controllers⁵, with the user unit (U) fixed on the smart wheelchair as shown in Fig. 9 and demonstrator unit (D) placed on a table marked by D in Fig. 3b. Each controller was driven by a three-dimensional PD controller:

$$\mathbf{u}_t^f = \mathbf{K}_p \mathbf{e}_t + \mathbf{K}_d \frac{d}{dt} \mathbf{e}_t, \quad (9)$$

where $\mathbf{u}_t^f$ is the control signal applied to the Falcon, $\mathbf{K}_p$ and $\mathbf{K}_d$ are the 3×3 position and derivative diagonal gain matrices, and $\mathbf{e}_t = \mathbf{b}_t - \mathbf{c}_t$ is the error at time t (i.e., the difference between the desired $[\mathbf{b}_t]$ and current $[\mathbf{c}_t]$ positions). Hence, each PD controller was parametrized by the set $\theta^f = \{\mathbf{K}_p, \mathbf{K}_d\}$.

For LAD, the PD controllers operated in one of two modes: standard (S) or assistance (A) takeover with a corresponding parameter set for each controller, denoted by θ_U^S , θ_U^A , θ_D^S , and θ_D^A , where the subscript and superscript label the unit and mode respectively. These parameters were manually tuned and are shown in Table 2.

⁵The two units used were donated by Novint Technologies.

Figure 9. Placement of the Novint Falcon Haptic Controller on the ARTY wheelchair. The user provides an input by moving the ball grip with standard axes as shown. For driving the wheelchair, only the x and y axes are relevant.

Table 2: PD parameters for controlling the user and demonstrator Falcon haptic devices. Note that the gain variables are diagonal matrices, and as such, only the diagonal elements are shown.

Controller	Mode	$\mathbf{K}_p(x, y, z)$	$\mathbf{K}_d(x, y, z)$
User	Standard	(15, 5, 10)	(0.1, 0.1, 0.1)
	Assistance	(50, 50, 50)	(0.1, 0.1, 0.1)
Demonstrator	Standard	(30, 10, 20)	(0.1, 0.1, 0.1)
	Assistance	(15, 5, 10)	(0.1, 0.1, 0.1)

During standard operation, the user drove the wheelchair without assistance; the desired state of the user Falcon controller was the center state $\mathbf{b}_t^U = (0, 0, 0)$. For the demonstrator controller, the desired state was the current state of the user's controller $\mathbf{b}_t^D = \mathbf{c}_t^U$. In other words, the demonstrator felt the user's controlling signals. During assistance, the signal flow was flipped and the user felt the demonstrator's control. More precisely, the demonstrator's Falcon had desired state $\mathbf{b}_t^D = (0, 0, 0)$ and the user's Falcon had desired state $\mathbf{b}_t^U = \mathbf{c}_t^D$. In preliminary tests, we experimented with bi-directional control during the assistance phase but this resulted in a coarse feel to the controller. We posit that this was due to communication delays, and future work would look into resolving this issue. The operating mode was controlled by the demonstrator using the When-to-Help h_t signal, toggled via a button press on the demonstrator Falcon grip.

6.1.2 Limited Control with Haptic Feedback To simulate turning control loss, we limited both the turning capability of the wheelchair as before and also set the user's Falcon to provide a greater counteracting force when the user attempted right turns. This was achieved by setting the y -axis component of the positional gain to a larger value $\mathbf{K}_{p,y} = 100$ when the ball grip position was negative. During assistance takeover, the original gain value was restored.

Survey Questions:

1. I found navigating the obstacle course with the wheelchair easy.
2. I found navigating the obstacle course frustrating.
3. I performed well on the task.
4. The task was difficult at times.
5. I found the driving assistance to be helpful.
6. The driving assistance interfered with my driving in a negative way.
7. The driving assistance enabled me to complete the task faster.
8. I found the driving assistance to be timely.
9. The driving assistance negatively impacted my driving ability.
10. I would have completed the task easily even without the driving assistance.
11. I liked having driving assistance for this task.

Figure 10. Survey questions for the LAD Experiment. Participants were asked to indicate how much they agreed with each of the eleven statements on a 5-point Likert scale. For the first two trials where assistance was not given, surveys consisted of only questions 1-4.

6.2 Experimental Setup

Our experimental route was similar to the simulation setup but with four hard right turns instead of two (labeled as R_1 to R_4 in Fig. 3b). We invited 15 participants (6 female, 9 male) ages 23-38 (mean: 27.3, SD : 4.22) to drive the wheelchair along the specified track (from S to F) a total of five times. Of the 15 participants, only four had previous experience using the Falcon haptic controller. As such, before beginning the trial, the participants drove around the experimental area and along the route until they felt confident about their driving ability.

During the first timed lap, full control was conferred, allowing us to obtain the baseline performance of an average user. The control limitation was applied during the latter four laps. Human demonstrator assistance was offered during the third lap, while the model was simultaneously trained. For the last two laps, each participant was provided with human or model assistance, the ordering of which was randomized to account for driver improvement. Eight of the fifteen participants underwent the model assistance first. After each trial, the participants were given a survey (shown in Fig. 10) to complete, indicating how much they agreed with eleven statements on a 5-point Likert scale.

6.3 Results: Lap Performance and Survey Results

The box-plot shown in Fig. 11 summarizes the lap completion times under the different conditions. Similar to the simulation experiment, we observed that the control limitation hampered performance, doubling lap times to an average of 90.97 s. With demonstrator assistance, the participants were able to complete the task with performance similar to the baseline (average lap times of 43.30 s and 43.43 s for the learning and test trial, respectively). Most importantly, participants using LAD-model assistance attained similar performance (average lap time of 47.49 s) to demonstrator assistance and the baseline, corresponding to a speedup of 191% over the limited-control condition.

Fig. 12 illustrates the empirical distributions of when assistance was offered. We see that both human and model assistance distributions are highly similar; the model only provided assistance at the sharp right turns (marked R_1 to R_4) despite the user attempting right turns at other portions of the course (e.g., as indicated by the path driven between R_3 and R_4). These results show that the model provided *contextual* assistance, based not only on the user's desired movements but also the environment (via the laser readings).

Figure 11. Lap times in seconds for all fifteen participants categorized by mode. The control limitation increased lap times from an average of 43.04 s ($SD = 8.32$ s) to 90.97 s ($SD = 26.13$ s). Demonstrator assistance enabled drivers to complete the task more quickly, reducing lap times back to the baseline (i.e., 43.30 s [$SD = 5.64$] and 43.43 s [$SD = 6.89$] for the learning trial and test trial, respectively). The model assistance (after a single demonstration) performed similarly to the demonstrator assistance, with slightly higher average lap times of 47.49 s ($SD = 10.99$).

Figure 12. Normalized smoothed density plots illustrating when assistance was offered by the human demonstrator and LAD model. Driving paths are shown as blue lines. Both distributions are highly similar—assistance was only offered at the marked sharp right turn points—indicating the learned model was able to capture when to appropriately assist. In particular, note the lack of assistance during the right turns made between R_3 and R_4 .

Survey Responses for LAD Experiment (Q1 to Q4)

Figure 13. Survey responses comparing the different lap conditions (Q1 to Q4). Participants found the course to be more difficult and frustrating under the limited-control condition. Responses during assistance (both human and model) were similar to the baseline condition (without control limitation).

Survey Responses for LAD Experiment with ARTY and Haptic Controllers (Q1 to Q11)

Figure 14. Survey responses from 15 participants for the experiment with haptic controllers, comparing the two laps with human assistance (both demonstration/learning and test laps) and the LAD-model assistance. The distributions are similar, indicating that participants felt they were appropriately helped in all three laps (irrespective of whether it was human or model-based assistance).

From the survey results for the first two laps (Fig. 13), we observed that more participants found navigating the course difficult and frustrating (Q1 and Q2) during limited control as compared to the other conditions. Specifically, 33% of the participants disagreed (or strongly disagreed) with Q1 and 40% agreed that the task was frustrating under limited control. Additionally, participants felt that they performed better on the task for the baseline (Q3) while the task was more difficult under limited control (Q4). To summarize, the control limitation was sufficient to make the task more challenging for our able-bodied participants. Notably, the response distributions for the assistance laps were comparable to the baseline, suggesting that the help offered alleviated the difficulty effected by the limited control.

Fig. 14 shows the responses for laps when assistance was offered. The response distributions between the human and LAD-model assistance laps are similar, which suggests that to the participants, the model assistance performed comparably to the human demonstrator—using a χ^2 goodness-of-fit test, we were unable to reject the hypothesis that the distributions were the same for all questions (at the Bonferroni-corrected $\alpha = 0.05$ level).

Focusing on the responses during the model assistance laps, we found participants answered positively with regards to the assistance; 14 out of the 15 participants found the assistance to be helpful (Q5) and timely (Q8), enabling them to complete the task faster (Q7). The remaining one participant was neutral to both statements. All participants who drove with model assistance preferred help for the task (Q11)⁶. These responses were also largely consistent, with only a minority (2 participants⁷) agreeing that the assistance negatively affected in their task (Q6 and Q9). In summary, participants not only performed quantitatively better with assistance (with lower lap completion times), but also felt that the human/model assistance helped them complete the task.

7. Conclusions

This paper proposed and discussed LAD as applied to the problem of deriving shared control policies for smart mobility assistance. We have contributed a probabilistic model that captures both when and how to assist, and an efficient mixture of OIESGP experts method for online training. We demonstrated the efficacy of our approach on a task requiring assistance during “hard” right turns. In both simulation and real-world experiments, our LAD-enabled smart wheelchair learned to provide contextual assistance where needed, speeding up lap times by almost double (i.e., an average speedup of 191% after only one demonstration). Survey responses by 15 human participants support the notion that the LAD-model performed similarly to the human demonstrator and participants appreciated the assistance given, which enabled them complete the driving task more effectively. These results show that it is feasible to (1) learn an assistive model on-the-fly and (2) execute the model to provide assistance similar to a human demonstrator.

We believe LAD to be fertile ground for future research. Moving forward, it is important to validate the model on more complex scenarios (e.g., passing through doorways, curb handling) with the target population. As a next-step, we seek to evaluate the system using real-world trials with end-users in a rehabilitation center or hospital. Furthermore, additional experiments are needed to resolve outstanding issues such as model generalizability (in particular, to test how well policies learned in one location extend to other environments) and possible model bias due to class label imbalance.

Although the Falcon was fully capable of providing haptic feedback, we observed that its force

⁶The one participant who disagreed with Q11 during the *human assistance* lap drove too close to the wall during the return leg from R₃ and R₄ and was momentarily immobile due to the obstacle avoidance method. Assistance was not offered, because it was possible for the participant to resolve the situation.

⁷The two participants who provided inconsistent responses were not native English speakers and may have misunderstood the statement.

capabilities were not sufficient to overcome the forces issued by our able-bodied participants. As such, future studies may look into using higher-grade haptic devices. It would also be interesting to apply our model to other robotic platforms such as humanoids, which would enable assistance with object manipulation tasks. In some circumstances, the robot’s physical embodiment and observations may be different from that of the human demonstrator and thus, require alternative training methodologies. To derive a proper policy, a correspondence mapping between the robot and demonstrator needs to be identified (Argall et al., 2009). Finally, we would like to highlight open questions arising from the LAD approach:

- **How can the model learn with latent assistive signals?** Thus far, we have assumed that h_t is visible but in certain scenarios, h_t is hidden or latent, making the training process more difficult. One potential solution is to rely on proxy observables and estimate the probability of assistance. Then, the How-to-Help model can be trained via a sampling process or by weighting the training instances with the probability that assistance is offered.

- **What if there are multiple assistants and multiple users?** For simplicity, we have limited our model description to a single user and single assistant. However, this can be extended to teams of assistants and users.

- **How can the model adapt to improvements on the part of the assisted user?** A core issue with assistance is that the user and assistant are likely to change and develop over time. For example, our wheelchair driver would hopefully improve with gained experience. Moreover, the attending occupational therapist may also learn when and how to better help. The simplest, albeit less scalable, solution is simply for the demonstrator to provide additional demonstrations. A potentially more interesting solution may come from merging our approach with ideas from human-robot cross-training for collaborative robots (Nikolaidis & Shah, 2013).

- **Can we do more with the learned policies?** This work illustrated how LAD can be applied to derive shared control policies but the learned policy can also be used for analyzing how humans assist under different circumstances. Further analysis may supplement other published findings (Dragan & Srinivasa, 2012), leading to general concepts for improving shared control in assistive devices.

Acknowledgements

The authors would like to thank members of the Personal Robotics Lab at Imperial College London for their assistance in conducting the experiments. Thank you to Novint Technologies for donating the two Falcon controllers used in the experiments. This work was supported in part by the EU FP7 ALIZ-E project (248116), and H. Soh is supported by a SMART Scholar grant.

References

- Amari, S. I. (1998). Natural gradient works efficiently in learning. *Neural Computation*, 10(2), 251–276. doi:10.1162/089976698300017746
- Argall, B. D., Chernova, S., Veloso, M., & Browning, B. (2009, May). A survey of robot learning from demonstration. *Robotics and Autonomous Systems*, 57(5), 469–483. doi:10.1016/j.robot.2008.10.024
- Calinon, S., Evrard, P., Gribovskaya, E., Billard, A., & Kheddar, A. (2009). Learning collaborative manipulation tasks by demonstration using a haptic interface. In *Proceedings of the International Conference on Advanced Robotics (ICAR)* (pp. 1–6).
- Carlson, T., & Demiris, Y. (2008). Human-wheelchair collaboration through prediction of intention and adaptive assistance. *IEEE International Conference on Robotics and Automation*, 3926–3931. doi:10.1109/ROBOT.2008.4543814
- Carlson, T., & Demiris, Y. (2010). Increasing robotic wheelchair safety with collaborative control: Evidence from secondary task experiments. In *Proceedings of the IEEE International Conference on Robotics and Automation* (pp. 5582–5587). doi:10.1109/ROBOT.2010.5509257

- Carlson, T., & Demiris, Y. (2012, June). Collaborative control for a robotic wheelchair: Evaluation of performance, attention, and workload. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 42(3), 876–888. doi:10.1109/TSMCB.2011.2181833
- Chernova, S., & Veloso, M. (2008, December). Teaching collaborative multi-robot tasks through demonstration. In *Proceedings of the 8th IEEE-RAS International Conference on Humanoid Robots* (p. 385–390). doi:10.1109/ICHR.2008.4755982
- Chow, H., & Xu, Y. (2006). Learning human navigational skill for smart wheelchair in a static cluttered route. *IEEE Transactions on Industrial Electronics*, 53(4), 1350–1361. doi:10.1109/TIE.2006.878296
- Csató, L., & Oppen, M. (2002, March). Sparse on-line Gaussian processes. *Neural Computation*, 14(3), 641–668. doi:10.1162/089976602317250933
- Demeester, E., Hüntemann, A., Vanhooydonck, D., Vanacker, G., Van Brussel, H., & Nuttin, M. (2007, December). User-adapted plan recognition and user-adapted shared control: A Bayesian approach to semi-autonomous wheelchair driving. *Autonomous Robots*, 24(2), 193–211. doi:10.1007/s10514-007-9064-5
- Demiris, Y. (2007). Prediction of intent in robotics and multi-agent systems. *Cognitive Processing*, 8(3), 151–158. doi:10.1007/s10339-007-0168-9
- Demiris, Y. (2009). Knowing when to assist: Developmental issues in lifelong assistive robotics. In *Proceedings of the International Conference of IEEE Engineering in Medicine and Biology Society* (pp. 3357–3360). doi:10.1109/IEMBS.2009.5333182
- Dillmann, R. (2004). Teaching and learning of robot tasks via observation of human performance. *Robotics and Autonomous Systems*, 47, 109–116. doi:10.1016/j.robot.2004.03.005
- Dragan, A. D., & Srinivasa, S. S. (2012). Assistive teleoperation for manipulation tasks. In *Proceedings of the 7th Annual ACM/IEEE International Conference on Human-Robot Interaction* (pp. 123–124). doi:10.1145/2157689.2157716
- Fox, D., Burgard, W., & Thrun, S. (1997). The dynamic window approach to collision avoidance. *IEEE Robotics Automation Magazine*, 4(1), 23–33. doi:10.1109/100.580977
- Gerkey, B. P., Vaughan, R. T., & Howard, A. (2003). The Player/Stage project: Tools for multi-robot and distributed sensor systems. In *Proceedings of the 11th International Conference on Advanced Robotics* (pp. 317–323). doi:10.1.1.8.3914
- Goil, A., Derry, M., & Argall, B. (2013, June). Using machine learning to blend human and robot controls for assisted wheelchair navigation. In *Proceedings of the 2013 IEEE International Conference on Rehabilitation Robotics (ICORR)* (p. 1–6). doi:10.1109/ICORR.2013.6650454
- Gomi, T., & Griffith, A. (1998). Developing intelligent wheelchairs for the handicapped. In *Assistive Technology and Artificial Intelligence* (Vol. 1458, pp. 150–178). doi:10.1007/BFb0055977
- Hermans, M., & Schrauwen, B. (2011). Recurrent kernel machines: Computing with infinite echo state networks. *Neural Computation*, 24(1), 104–133. doi:10.1162/NECO_a.00200
- Jordan, M. I., & Jacobs, R. A. (1994). Hierarchical mixtures of experts and the EM algorithm. *Neural Computation*, 6(2), 181–214. doi:10.1162/neco.1994.6.2.181
- Levine, S. P., Bell, D. A., Jaros, L. A., Simpson, R. C., Koren, Y., & Borenstein, J. (1999). The NavChair Assistive Wheelchair Navigation System. *IEEE Transactions on Rehabilitation Engineering*, 7(4), 443–451. doi:10.1109/86.808948
- Marchal-Crespo, L., Furumasa, J., & Reinkensmeyer, D. J. (2010). A robotic wheelchair trainer: Design overview and a feasibility study. *Journal of NeuroEngineering and Rehabilitation*, 7, 40. doi:10.1186/1743-0003-7-40
- Metta, G., Sandini, G., Vernon, D., Natale, L., & Nori, F. (2008). The iCub humanoid robot: An open platform for research in embodied cognition. *8th Workshop on Performance Metrics for Intelligent Systems*, 50–56. doi:10.1145/1774674.1774683
- Miller, D. P., & Slack, M. G. (1995). Design and testing of a low-cost robotic wheelchair prototype. *Autonomous Robots*, 2(1), 77–88. doi:10.1007/BF00735440
- Neal, R. (1996). *Bayesian learning for neural networks*. Lecture Notes in Statistics (No. 118). New York, NY: Springer Verlag. doi:10.1007/978-1-4612-0745-0
- Nguyen, A. V., Nguyen, L. B., Su, S., & Nguyen, H. T. (2013). Shared control strategies for human-machine interface in an intelligent wheelchair. In *Proceedings of the Engineering in Medicine and Biology Society*

- Conference (EMBC) (pp. 3638–3641). doi:10.1109/EMBC.2013.6610331
- Nguyen-Tuong, D., Seeger, M., & Peters, J. (2009). Model learning with local Gaussian process regression. *Advanced Robotics*, 23(15), 2015–2034. doi:10.1163/016918609x12529286896877
- Nikolaidis, S., & Shah, J. (2013). Human-robot cross-training: Computational formulation, modeling and evaluation of a human team training strategy. In *Proceedings of the 8th ACM/IEEE International Conference on Human-robot Interaction* (pp. 33–40). doi:10.1109/hri.2013.6483499
- Novint Technologies. (2013). *Falcon technical specifications*. Retrieved from <http://www.novint.com/index.php/novintxio/41>
- Orpwood, R., Halsey, S., Evans, N., & Harris, A. (2005). Development of a powered mobility vehicle for nursery-aged children. In A. Pruski & H. Knops (Eds.), *Assistive technology: From virtuality to reality: AAATE 2005* (pp. 540–545). Amsterdam, The Netherlands: IOS Press.
- Parikh, S. P., Grassi, V., Kumar, V., & Okamoto, J. (2007). Integrating human inputs with autonomous behaviors on an intelligent wheelchair platform. *Intelligent Systems, IEEE*, 22(2), 33–41. doi:10.1109/MIS.2007.36
- Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., ... Ng, A. (2009, May). ROS: an open-source Robot Operating System. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA) Workshop on Open Source Software* (Vol. 3). Kobe, Japan.
- Quiñero Candela, J., & Rasmussen, C. E. (2005, December). A unifying view of sparse approximate Gaussian process regression. *Journal of Machine Learning Research*, 6, 1939–1959.
- Rasmussen, C. E., & Ghahramani, Z. (2002). Infinite mixtures of Gaussian process experts. *Advances in Neural Information Processing Systems*, 2, 881–888.
- Rofer, T. (1998). Architecture and applications of the Bremen Autonomous Wheelchair. *Information Sciences*, 126(1-4), 365–368. doi:10.1016/S0020-0255(00)00020-7
- Rozo, L., Calinon, S., Caldwell, D. G., Jimenez, P., & Torras, C. (2013). Learning collaborative impedance-based robot behaviors. In *Proceedings of the AAAI Conference on Artificial Intelligence* (pp. 1422–1428). Bellevue, WA, USA.
- Seligman, M. E. (1972). Learned helplessness [Review]. *Annual Review of Medicine*, 23(5), 407. doi: 10.1146/annurev.me.23.020172.002203%7D
- Simpson, R. C. (2005). Smart wheelchairs: A literature review. *Journal of Rehabilitation Research and Development*, 42(4), 423–436. doi:10.1682/JRRD.2004.08.0101
- Simpson, R. C., Lopresti, E., Hayashi, S., Nourbakhsh, I., & Miller, D. (2004). The smart wheelchair component system. *Journal of Rehabilitation Research and Development*, 41(3B), 429–442. doi:10.1682/jrrd.2003.03.0032
- Simpson, R. C., LoPresti, E. F., & Cooper, R. A. (2008). How many people would benefit from a smart wheelchair? *Journal of Rehabilitation Research and Development*, 45(1), 53–71. doi:10.1682/JRRD.2007.01.0015
- Soh, H., & Demiris, Y. (2012). Towards early mobility independence: An intelligent paediatric wheelchair with case studies. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) Workshop on Progress, Challenges and Future Perspectives in Navigation and Manipulation Assistance for Robotic Wheelchairs*. Vilamoura, Portugal.
- Soh, H., & Demiris, Y. (2013, November). When and how to help: An iterative probabilistic model for learning assistance by demonstration. In *Proceedings of the 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (p. 3230–3236). doi:10.1109/IROS.2013.6696815
- Soh, H., & Demiris, Y. (2014a, May). Incrementally learning objects by touch: Online discriminative and generative models for tactile-based recognition. *IEEE Transactions on Haptics*, 7, 512–525. doi:10.1109/TOH.2014.2326159
- Soh, H., & Demiris, Y. (2014b, June). Spatio-temporal learning with the online finite and infinite echo-state gaussian processes. *IEEE Transactions on Neural Networks and Learning Systems*, 26. doi:10.1109/TNNLS.2014.2316291
- Tresp, V. (2001). Mixtures of Gaussian processes. In *Advances in Neural Information Processing Systems* (pp. 654–660). Cambridge, MA: MIT Press.
- Tsotsos, J. K., Verghese, G., Dickinson, S., Jenkin, M., Jepson, A., Milios, E., ... Mann, R. (1998). PLAYBOT: A visually-guided robot for physically disabled children. *Image & Vision Computing, Special Issue on Vision*

- for the Disabled*, 16(4), 275-292. doi:10.1016/s0262-8856(97)00088-7
- Urdiales, C., Fernandez-Espejo, B., Annicchiarico, R., Sandoval, F., & Caltagirone, C. (2010). Biometrically modulated collaborative control for an assistive wheelchair. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 18(4), 398-408. doi:10.1109/TNSRE.2010.2056391
- Vanhooydonck, D., Demeester, E., Hüntemann, A., Philips, J., Vanacker, G., Van Brussel, H., & Nuttin, M. (2010). Adaptable navigational assistance for intelligent wheelchairs by means of an implicit personalized user model. *Robotics and Autonomous Systems*, 58(8), 963-977. doi:10.1016/j.robot.2010.04.002
- Vijayakumar, S., D'souza, A., & Schaal, S. (2005). Incremental online learning in high dimensions. *Neural Computation*, 17(12), 2602-2634. doi:10.1162/089976605774320557
- Yanco, H. A. (1998a). Integrating robotic research: A survey of robotic wheelchair development. In *Proceedings of the AAAI Spring Symposium on Integrating Robotic Research*. Palo Alto, CA.
- Yanco, H. A. (1998b). A robotic wheelchair system: Indoor navigation and user interface. *LNCS Assistive Technology and Artificial Intelligence: Applications in Robotics User Interfaces and Natural Language Processing*, 1458, 256-268. doi:10.1007/bfb0055983

Authors' names and contact information: Harold Soh, Future Urban Mobility, Singapore-MIT Alliance for Research and Technology, Singapore. Email: haroldsoh@smart.mit.edu. Yiannis Demiris, Personal Robotics Laboratory, Intelligent Systems and Networks, Dept. of Electrical and Electronic Engineering, Imperial College London, UK. Email: y.demiris@imperial.ac.uk.