Noname manuscript No.
(will be inserted by the editor)

Learning from Demonstration for Locally Assistive Mobility Aids

James Poon · Yunduan Cui · Jaime Valls Miro · Takamitsu Matsubara

Received: date / Accepted: date

Abstract Active assistive systems for mobility aids are largely restricted to environments mapped *a-priori*, while passive assistance primarily provides collision mitigation and other hand-crafted behaviors in the platform's immediate space. This paper presents a framework providing active short-term assistance, combining the freedom of location independence with the intelligence of active assistance. Demonstration data consisting of on-board sensor data and driving inputs is gathered from an able-bodied expert maneuvring the mobility aid around a generic interior setting, and used in constructing a probabilistic intention model built with Radial Basis Function Networks. This allows for short-term intention prediction relying only upon immediately available user input and on-board sensor data, to be coupled with real-time path generation based upon the same expert demonstration data via Dynamic Policy Programming, a stochastic optimal control method. Together these two elements provide a combined assistive mobility system, capable of operating in restrictive environments without the need for additional obstacle avoidance protocols. Experimental results in both simulation and on the University of Technology Sydney semi-autonomous wheelchair in settings not seen in training data show promise in assisting users of power mobility aids.

J. Poon, J. V. Miro
Centre for Autonomous Systems, University of Technology Sydney, Australia
Tel.: +61-2-9514-2588
E-mail: james.poon@student.uts.edu.au, jaime.vallsmiro@uts.edu.au

Y. Cui, T. Matsubara
Intelligent System Control Laboratory, Nara Institute of Science and Technology, Japan
Tel.: +81-743-72-5350
Fax: +81-743-72-5359
E-mail: cui.yunduan.ck4@is.naist.jp, takam-m@is.naist.jp

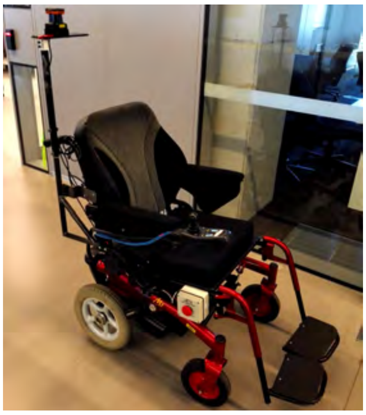


Fig. 1: University of Technology Sydney (UTS) semi-autonomous wheelchair.

1 Introduction

The global population is ageing quickly, with predictions that the worldwide proportion of people aged 60 and over is expected to double by 2050 [1]. As mobility aids promote independence and self-esteem in elderly and frail users, there is a strong motivation to develop intelligent systems capable of providing improved assistance to these groups [2]. One aim is to thus develop an assistive framework that can be incorporated into power mobility devices (PMDs) such as the one shown in Fig. 1. As devices such as power wheelchairs are large, heavy and powerful machines it is common for prospective users to meet a strict set of conditions before prescription is approved [3] even if they are otherwise capable of independently performing other routine tasks. In this light, the target end users of this work are those who are still capable of independence apart from mobility rather than those requiring constant oversight from a carer due to more complex healthcare and lifestyle needs.

Assistance for sensor-equipped PMDs can be broadly divided into two dominant flavors, referred to as reactive or active assistance. Most reactively assistive systems keep the user significantly in the loop, only intervening the user's input commands when collision is imminent or autonomous takeover is momentarily required, for example, in traversing a particularly narrow passage. In active assistance, the user is mostly removed from the control loop; their command signals are used to infer an intended destination, to which autonomous navigation algorithms then guide the PMD.

Reactive assistance primarily aims to mitigate collisions, as well as possibly accommodating hand-tooled heuristic behaviors such as driving parallel to walls. Earlier systems such as the Bremen Autonomous Wheelchair [4] or NavChair [5] provided safety mechanisms such as nullifying potentially hazardous input signals or incorporating obstacle avoidance behaviours adapted from algorithms developed for autonomous robots, e.g. the Vector Field Histogram [6] used recently in [7]. A more recent approach in reactively assistive PMDs is the weighted fusion of robot command signals with that from the user, as investigated by [8–10] among others. These

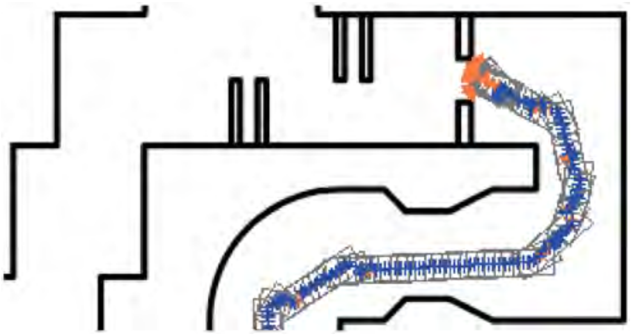


Fig. 2: An example of a reactive planner causing the platform (gray) to be trapped in local minima. Orange crosses represent a forward projection of user input, and blue crosses are navigational control points from the Dynamic Window Approach.

frameworks utilise a measure of “goodness” from the user’s behaviours to allocate authority over the final command signal, typically a weighted sum between the user’s input and command signals inferred from another algorithm. This measure typically considers metrics such as obstacle proximity, user input fluidity or navigational task relevance. The authors believe however that it is inherently safer for the user to merely provide suggestions to a system that is in total control at all times, as there is little promise of safety in the combined command signal even if both may be safe in isolation. The primary disadvantage to reactive assistance is that the user’s intended destination in most implementations is not considered. As a result the system is unable to actively assist the user in areas where pure obstacle avoidance would become trapped in local minima.

Active assistance tends to rely on *a-priori* maps of spaces that the system operates in. An early instance of this can be seen again on the NavChair [11], which seeks to change between modes such as doorway navigation and wall-following depending on its location and observed surroundings. Given data obtained from end-users or demonstrators, later works have approached inference of the likeliest intended target as a classification exercise. Such long-term “global” destination inference has already been previously addressed by several means including Hierarchical Hidden Markov Models [12], Gaussian Processes [13] and various Bayesian frameworks [14, 15] among other heuristic approaches [16–18]. Once an intention is deemed sufficiently probable, semi-autonomous navigation typically commences.

A motivating example for active assistance can be seen in Figure 2. User inputs are projected forward over a brief temporal window, yielding an approximate navigational goal for a reactive local controller; in this example, the Dynamic Window Approach [19]. These algorithms consider both the goal and immediately available sensor data to yield a safe control point and corresponding twist; however as they are limited solely to determining the next platform control signal with no consideration of longer-term path viability, the platform can become stuck in scenarios where comprehensive path planning is necessary for ensuring reliable traversal.

A significant limitation of active assistance is its restriction to working in areas having up-to-date occupancy maps. Instead it is arguably more desirable for a system to provide assistance ‘anywhere’, like reactively assistive frameworks are capable of, without the need for map-building by only relying upon immediately available on-board sensor data. In this “local” space one can infer immediate short-term destinations being points of interest that the user wishes to pass through or stop at at a given instant, to which short-term path planning can take place. It is these short-term destinations which are defined as intentions for the purpose of this work. The limited physical range of these short-term destinations in comparison to the long-term destinations in the aforementioned literature is not a significant issue, given user inputs and sensor data are readily available over the entire duration of travel. Although such an approach does not guarantee global long-term optimality, as postulated in [20] a destination inference and its corresponding path only have to be accurate for a brief portion of travel until the next path is available, resulting in an overall route that is an overall concatenation of many fragments of these short-term paths. While the aforementioned classification methodologies can function solely on low-dimensional pose information when considered alongside user input data, applying such an approach directly to sensor data is meaningless as there is little correlation between long-term destination and the immediately observable scene. However, the high dimensionality of typical mobile robot sensor data has, to the best of the authors’ knowledge, made short-term active assistance a largely unexplored area.

The contribution of this work in the literature covering shared control of PMD devices is in short-term intention estimation without depending upon a *a-priori* map, for enabling subsequent robot path planning and navigation in a non-reactive manner. As these intentions can encompass difficult locations such as doorways for which heuristics are designed to handle in reactively assistive systems, the inference of intentions allows the merging of collision avoidance and selective interaction with artefacts, bypassing the need for situational behaviors found in systems based upon obstacle avoidance. Our work was originally published as a conference paper [21]. This manuscript expands upon this preliminary work with the following additions:

1. Consideration of additional metrics [22,23] for a more thorough framework evaluation.
2. A human-centric analysis of the simulator used in both works against the real wheelchair.
3. Further experimentation in simulation with a larger user pool in a more complex test environment, and on the UTS wheelchair with a disabled volunteer.
4. A quantitative *a-posteriori* evaluation of the framework’s predictive accuracy on driving data by able users.

The remainder of this paper is arranged as follows. Section 2 documents a methodology to capture the local intention inference and path planning behaviors of able experts. Section 3 presents new experiments undertaken to evaluate the assistive framework both in simulation and on an instrumented mobility aid (Fig. 1), with a discussion of results and outcomes following in Section 4. Section 5 closes with conclusions and future work.

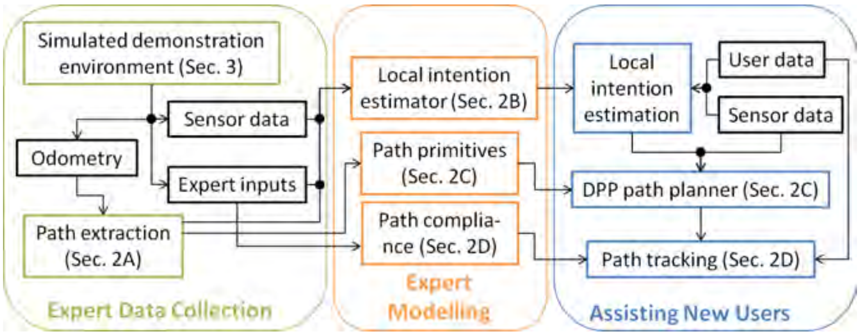

Simulated demonstration environment (Sec. 3)
Sensor data
Odometry
Expert inputs
Path extraction (Sec. 2A)
Expert Data Collection
Local intention estimator (Sec. 2B)
Path primitives (Sec. 2C)
Path compliance (Sec. 2D)
Expert Modelling
Local intention estimation
User data
Sensor data
DPP path planner (Sec. 2C)
Path tracking (Sec. 2D)
Assisting New Users

Fig. 3: Overview schematic of assistive framework.

2 Methodology

This section details the modelling of local intention estimation and path planning, as summarized in Figure 3. Training data comes from an able demonstrator driving the PMD around a simulated interior space while position, expert joystick inputs and sensor measurements from a planar laser scanner, are logged. A simulated environment is used here to allow for flexibility in creating environments suited to the capture of expert behaviours. As demonstrated in Section 3.1, the simulation behaves acceptably closely to the real PMD for data collection and user interaction.

For intention estimation a behavioural model is built yielding a likelihood distribution across discrete points in a moving window centralised around the PMD, trained only upon sensor data and user input. Fitting with the proposed definition of a destination as a point which the user intends to traverse to or through, this distribution then allows the likeliest destination point to be inferred. This is covered in Section 2.2.

From the same training data, a methodology for the learning of short-term path planning behaviors and user compliance is also proposed and detailed in Sections 2.3-2.4. Short local paths are combined with sensor data as a reward surface, which is then refined online via Dynamic Policy Programming (DPP) [24], a stochastic optimal control method. The final output of this planner is taken as the local path to be followed to its terminus, or user compliance indicates a loss of alignment with their true intention.

2.1 Defining Local Paths

In order to break down a continuous sequence of driving data, several criteria to terminate paths are proposed in Figure 4.

- Loss of visibility from starting position or exiting local window of starting position
- PMD turns $\geq 90^\circ$ relative to starting orientation

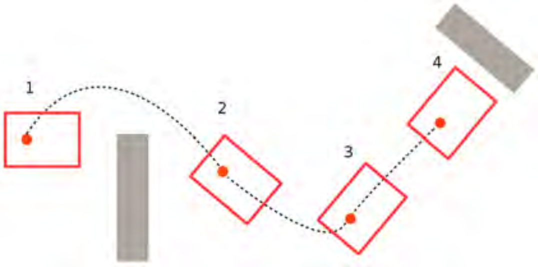


Fig. 4: Examples of local path termination along a path (black dashed line) taken by the PMD (red rectangle with circle at origin). Grey polygons represent obstacles. Termination condition from pose 1→2: Visibility loss due to obstacle. 2→3: PMD rotates $\geq 90°$. 3→4: Inflection in forward joystick axis due to stop.

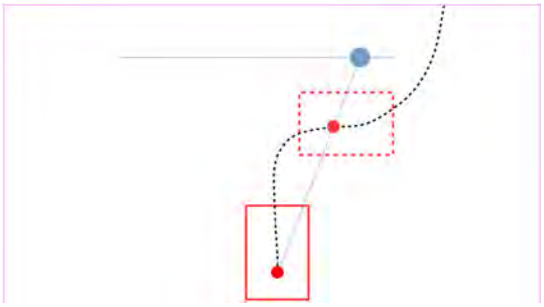


Fig. 5: Training data for the cell at the termination point (red circle in dashed PMD footprint) along the path taken by the PMD (black dash) is recorded, consisting of the expert joystick input at the origin (red circle in solid PMD footprint), and the range of the laser scanner beam (blue line capped with blue circle) covering the termination point. The grey line represents the edge of an obstacle perceived by the laser scanner, and the local planning window is bounded in magenta.

- Inflection in forward/reverse joystick axis

The first criteria ensures paths are terminated once immediately available sensor data can no longer provide information on the PMD's future surroundings, whereas the latter two indicate the user either intends to pursue a significantly different objective or they have reached their final destination.

In determining a suitable local window size for both the latter half of the first termination criterion and the subsequent path planning, the spatial requirements of typical PMDs are first considered. In Australia where this research was undertaken, an open circulation space of approximately 2m square is a requirement for building planners [25] for spaces such as landing areas near doors for disabled access bathrooms, so this is taken as a lower bound. On the basis that the framework is concerned with only an immediate path to track, it follows that the usefulness of path points diminishes as the path length increases; when the system is able to constantly replan paths at a reasonable speed, there is no need to plan beyond challenges in the immediate vicinity. Hence the requisite quantity of local space can be thought of as a softly defined region, bounded between the requisite minimum of planning space as implicated by the PMD's physical characteristics and an upper limit beyond which a path planning cycle becomes needlessly complex. In this work the front half of a 5m square moving window centred at the PMD is taken, allowing for an acceptable path planning time while also exceeding the lower bound for maneuvrability. With additional sensor coverage of the wheelchair's rear, the framework would also be readily applicable to reversing maneuvres.

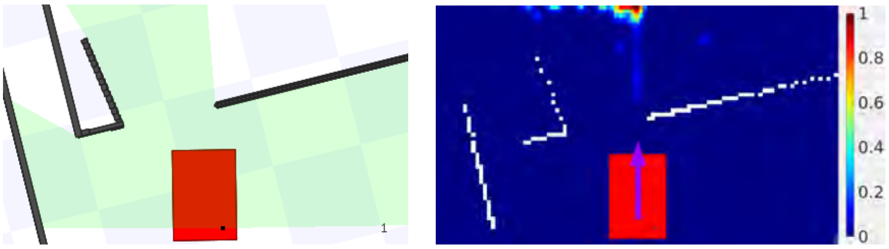
2.2 Local Intention Estimation

Here, an intention within the local window around the PMD is defined as the discrete cell with the highest likelihood of being a local path termination point. A cell size of 0.05m square is taken to allow for an acceptable path planning time (Sec. 2.3). Demonstration training data contains Cartesian pose information $\boldsymbol{X}_{1:N}$, expert actions $\boldsymbol{Y}_{1:N}$ of 2D Cartesian joystick positions, and sensor data $\boldsymbol{Z}_{1:N}$ containing polar co-ordinates $(r,\theta)_n^{1:|z|}$ of obstacle points across an on-board laser scanner's 180° horizontal field of view. For all instances of training data a local path is obtained as per the criteria above.

Rather than attempting to match instances of $z_{1,\dots,N}$ in their entirety to the new z^* as this would be easily overfitted, individual laser scan beams are instead considered in conjunction with user input. This is done with the aim of both mitigating effects from the curse of dimensionality, and to allow for improved tolerance to large variation in environmental structure while retaining some information on the relationship between $\boldsymbol{Z}$ and the sense of space the expert afforded the PMD. For each path indexed $n = 1,2,\dots,N$ a training data tuple for the cell corresponding to goal $\boldsymbol{G}_n = [x^g, y^g]$, the Cartesian point at which the path terminated, is recorded. A tuple consists of $\{\boldsymbol{Y}_n, r_n^i\}$ where i is the beam index determined by $\arg\min_{i=1,2,\dots,|\boldsymbol{z}|} ||\theta_n^i - \text{arctan2}(y^g, x^g)||$. A summary of nomenclature can be found in Table 5.

Only the edges of obstacles within the local window perceived by $\boldsymbol{z}$ are considered, to remove reliance upon obstacle points from future measurements. Each grid cell then has its own one-class classifier built in the form of a Radial Basis Function Network [26] utilising a strong 0 prior, as the data only contains positive examples. Each classifier is designed to take a training tuple as input, and is trained to yield an intention likelihood between 0 and 1 following the additive approach in [27]. Taking all classifiers into consideration and normalizing across the grid cells thus yields an intention likelihood estimate $P(\boldsymbol{g}^*|\boldsymbol{y}^*, \boldsymbol{z}^*)$ given new joystick input $\boldsymbol{y}^*$ and laser range information $\boldsymbol{z}^*$. To reduce computational cost in experiments, only classifiers for cells which recieved training data were queried. The final output of the intention estimator is the position $\boldsymbol{g}^*$ of the cell with the highest likelihood, which serves as an objective in the subsequent path planning step.

An illustrative example of the intention estimator is shown in Figure 6.



(a) Simulated wheelchair (birds-eye view).

(b) Resultant intention heatmap from forward joystick input.

Fig. 6: Example of intention estimation. Fig. 6a shows a simulated PMD (red) with a laser scan (green) observing nearby obstacles (black). Fig. 6b shows the resultant intention probability distribution given a forward joystick input (purple arrow) on the PMD, with perceived obstacles in white. Path planning continues in Fig. 8.

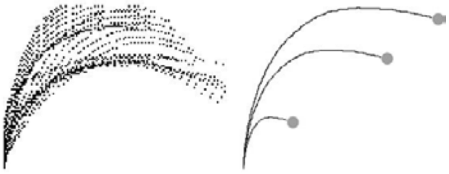
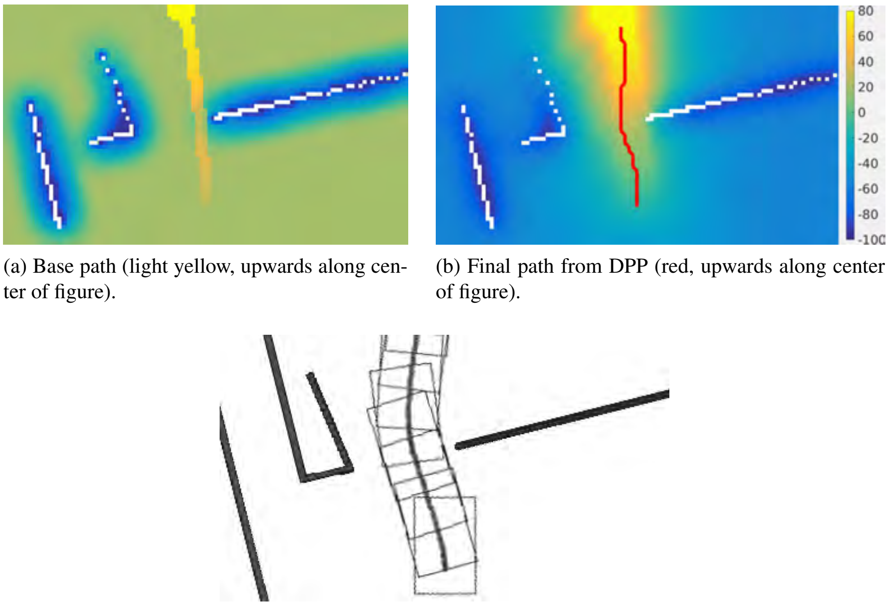


Fig. 7: Paths from training data sharing a similar endpoint orientation are gathered (left); their resulting primitive can then be spatially scaled to arbitrary positions (right, gray circles).

2.3 Local Path Planning

Due to our significantly truncated operating window and rapid rate of re-planning, here path primitives are utilized for planning due to their simplicity and speed. This is in contrast to the more exhaustively complete longer-term path planning algorithms detailed in [28] and other literature.

Local paths from training data with similar endpoint orientations are gathered and a primitive is obtained as a least-squares solution. As opposed to the velocity domain-based arcs in [23], the primitives in this paper are in real space and are formulated similarly to the approach in [29]. However the environment is not immediately considered, as the framework operates solely within visible space and considers the local occupancy map later. In this work 17 discretized endpoint orientions are taken for creating expert-styled path primitives; this number was determined as the limit beyond which no training data would be available for a given orientation. Then for a goal point $\boldsymbol{g}^*$, a path primitive is selected based on the nearest average end point and spatially scaled (Fig. 7) to reach it for a natural baseline. As this inferred path is not guaranteed to be safe, it instead serves as the basis for real-time path generation with consideration to the local occupancy map.



(a) Base path (light yellow, upwards along center of figure).

(b) Final path from DPP (red, upwards along center of figure).

(c) Resultant PMD movement from path tracking.

Fig. 8: Example of path planning with DPP, from the intention resultant from Fig. 6. A path primitive (Fig. 8a) is set as the baseline policy with a high reward (light yellow) while perceived obstacles (white) from the PMD's point of view are inflated and given a low reward (dark blue cells). Gradient ascent across the reward landscape after DPP optimization yields the final DPP path (red) in Fig. 8b, with resultant PMD movement shown in Fig. 8c.

2.3.1 Dynamic Policy Programming

Stochastic optimal control learns a Markov Decision Process (MDP) defined by a 5-tuple $(\mathcal{S},\mathcal{A},\mathcal{T},\mathcal{R},\gamma)$. $\mathcal{S}$ is a finite set of states, $\mathcal{A}$ is a finite set of actions, $\mathcal{T}^{a}_{ss'}$ is the transition probability from state s to state s' under the action a, $r^{a}_{ss'} = \mathcal{R}(s,s',a)$ is the reward from state s to state s' under the action a. $\mathcal{S}$ is defined as the local window around the PMD discretized into a grid-world for path planning, and $\mathcal{A}$ as a list of 9 actions that can be taken at each grid cell: moving to any of its 8 immediate neighbors or remaining in place. $\gamma \in (0,1)$ is the discount factor. The policy $\pi(a|s)$ denotes the probability of taking the action a under the state s. The value function $V_{\pi}(s) = \lim_{k\to\infty} \mathbb{E}_{\pi}\big[\sum_{k=1}^{K} \gamma^{k-1} r_{s_{t+k}} | s_t = s\big]$ is the expected return when the process starts in s and the decision maker follows the policy π. The solution of MDP is an optimal policy π^* that attains the maximum expected reward:

$$\pi^* = \arg\max_{\pi} \sum_{\substack{a\in\mathcal{A}\\ s'\in\mathcal{S}}} \pi(a|s)\mathcal{T}^{a}_{ss'}\big(r^{a}_{ss'} + \gamma V^*(s')\big), \forall s \in \mathcal{S}. \tag{1}$$

DPP builds a new value function by adding the Kullback-Leibler divergence to the reward function $\mathcal{R}$ as a penalty. The Kullback-Leibler divergence between the policy π and the baseline policy $\bar{\pi}$, and the new value function V are defined as:

$$g_{\bar{\pi}}^{\pi}(s) = KL(\pi \| \bar{\pi}) = \sum_{a \in \mathcal{A}} \pi(a|s) \log \left(\frac{\pi(a|s)}{\bar{\pi}(a|s)} \right), \tag{2}$$

$$V_{\bar{\pi}}(s) \triangleq \lim_{k \to \infty} \mathbb{E}_{\pi} \left[\sum_{k=1}^{K} \gamma^{k-1} \left(r_{s_{t+k}} - \frac{1}{\eta} g_{\bar{\pi}}^{\pi}(s_{t+k-1}) \right) \bigg| s_t = s \right] \tag{3}$$

where $\mathbb{E}_{\pi}$ denotes the expectation over transition model $\mathcal{T}$ and the current policy π. $\eta \in (0,1)$ is a constant that controls the Kullback-Leibler divergence term.

According to [24], the action preferences function [30] for all state action pairs $(s,a) \in \mathcal{S} \times \mathcal{A}$ in the k-th iteration are defined as $\Psi_k(s,a) = \frac{1}{\eta} \log \bar{\pi}^{k-1}(a|s) + \sum_{s' \in \mathcal{S}} \mathcal{T}_{ss'}^{a} \big(r_{ss'}^{a} + \gamma V_{\bar{\pi}}^{k-1}(s') \big)$. It represents the closed form of the optimal policy π^* following:

$$V_{\bar{\pi}}^{k+1}(s) = \frac{1}{\eta} \log \sum_{a \in \mathcal{A}} \exp(\eta \Psi_k(s,a)), \tag{4}$$

$$\bar{\pi}^{k+1}(a) = \frac{\exp\big(\eta \Psi_k(s,a)\big)}{\sum_{a' \in \mathcal{A}} \exp\big(\eta \Psi_k(s,a')\big)}. \tag{5}$$

The optimal action preferences function determines DPP's optimal policy according to Eq. (5). The update recursion of Ψ follows:

$$\Psi_{k+1}(s,a) = \Psi_k(s,a) - \mathcal{M}_{\eta}\Psi_k(s) + \sum_{s' \in \mathcal{S}} \mathcal{T}_{ss'}^{a}(r_{ss'}^{a} + \gamma \mathcal{M}_{\eta}\Psi_k(s'))) \tag{6}$$

where $\mathcal{M}_{\eta}\Psi_k(s)$ is the the Boltzmann soft-max operator:

$$\mathcal{M}_{\eta}\Psi(s) = \sum_{a \in \mathcal{A}} \frac{\exp\big(\eta \Psi(s,a)\big)\Psi(s,a)}{\sum_{a' \in \mathcal{A}} \exp\big(\eta \Psi(s,a')\big)}. \tag{7}$$

Following Eq. (6), DPP updates the action preferences function Ψ to iteratively improve its value function to the optimal one while considering the smoothness in the policy update (controlled by η). A summary of nomenclature can be found in Table 6.

2.3.2 Path Planning via DPP

In order to use the inferred local path within DPP, it is taken to serve as the baseline reward (Eq. 5). Cells along the path receive increasingly positive reward, whereas cells near obstacles perceived by the laser scanner receive an increasingly negative reward. DPP then optimizes its policy grid-world with smooth updates giving consideration to the baseline policy, and the resultant path drawn from the final policy is then followed. An example of path planning can be seen in Figure 8.

Given initial action preferences $\Psi_0(\cdot,\cdot)$, DPP parameters γ, η and the number of iterations K, the process of the path generator is summarised in Algorithm 1.

Algorithm 1: DPP path generator

```
input  : $\Psi_0(\cdot,\cdot)$, $\gamma$, $\eta$, $K$
/* DPP loop                                              */
for $k = 1,2,3,...,K-1$ do
    for $(s,a) \in \mathcal{S} \times \mathcal{A}$ do
        calculate $s^{'}$, the next state of $s$ under action $a$
        $\mathcal{T}_k \Psi_k(s,a) = r^a_{ss^{'}} + \gamma \mathcal{M}_\eta \Psi_k(s^{'})$
        $\Psi_{k+1}(s,a) = \Psi_k(s,a) + \mathcal{T}_k \Psi_k(s,a) - \mathcal{M}_\eta \Psi_k(s)$

/* generate path                                         */
$i = 0$; $s_0 \leftarrow$ position of occupancy grid origin
while $s_i$ is within occupancy grid do
    $a_i = \arg\max_{a \in \mathcal{A}} \Psi_{k+1}(s,a)$
    calculate $s_{i+1}$, the next state of $s_i$ under action $a_i$
    $i = i + 1$
output: path $\{s_1, s_2, ..., s_{i-1}\}$
```

2.4 User Path Compliance

Path tracking via Pure Pursuit [31] commences once a local path has been transformed into global co-ordinates. This controller was chosen primarily for its simplicity and speed, although it can be readily substituted by others such as DWA [19]. The magnitude of joystick displacement scales the output linear and angular velocities, allowing the user to always control the rate of PMD motion. Pure Pursuit derives linear and angular velocities from a control point located a fixed lookahead distance (0.8m for the UTS wheelchair) further ahead along the path. Given this point is always within the PMD's reference frame, it serves as a suitable objective to gauge the relevance of the remainder of the current tracking path against.

In a similar manner to the construction of the intention estimation model, a compliance model is built for potential control points within the lookahead distance while only considering joystick input, resulting in a compliance estimate in range (0,1) given a control point and a^*. Either reaching the end of the path or insufficient compliance results in a new path request.

3 Experiments

This section details the experiments undertaken in further evaluating the assistive framework in both simulation and on the UTS wheelchair. Ethical approval [1] was granted prior to experimentation, as was consent from all participants.

Training data was provided by an able expert in a Stage (`wiki.ros.org/stage`) simulation of the UTS wheelchair driving inside the home environment depicted in Fig 9. The driver was tasked with navigating throughout the entirety of the environment's accessible space while maintaining safe distances to obstacles and walls. PMD odometry, user input and simulated laser scanner data were logged at 10 Hz.

[1] UTS HREC 2012000400: "Applying Assistive Robotic Technology to Disabled Care Practice at Greystanes Disability Services"

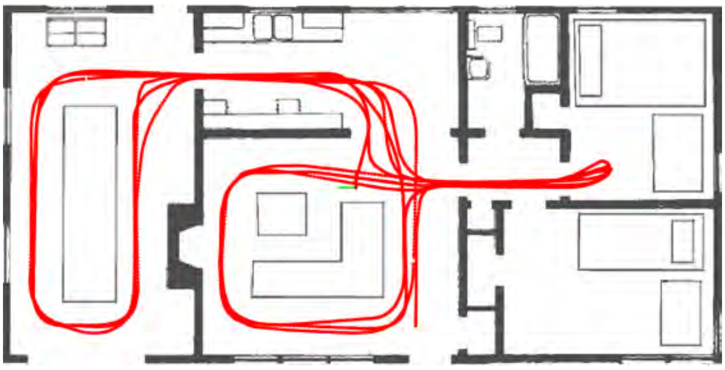


Fig. 9: Training data (red) from an able demonstrator in a 20 × 10m simulated home environment.

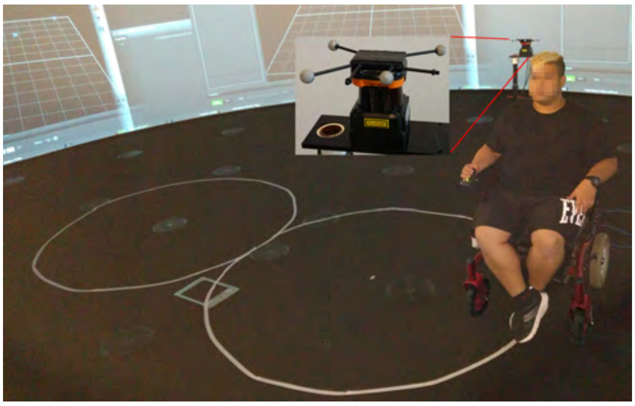
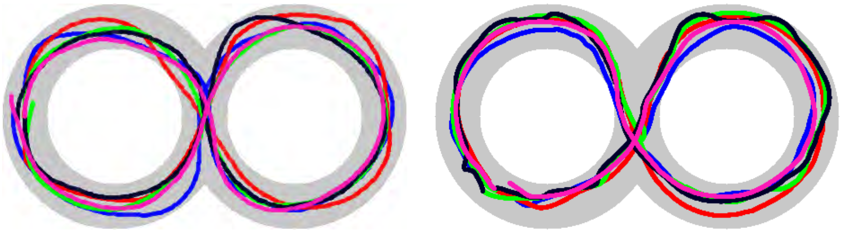


Fig. 10: Real experiment in the UTS Data Arena, overlaid with a close-up of a 6-DOF tracking marker on top of the wheelchair's laser scanner.

Experimentation was conducted on an Intel i7 Ubuntu laptop with 16 GB RAM and Nvidia 980M GPU. An average intention inference and path generation cycle took 0.46 ± 0.06 seconds, excluding minor latencies arising from MATLAB/ROS (`www.ros.org`) communications. For path planning a reward of -100 was allocated to obstacle cells. Cells near obstacles received a negative reward from Gaussian blurring of the obstacle reward map with a std. dev. of 2 cells when all other cells were set to 0. Cells along the path were then allocated a linearly increasing reward from 10 to 100 from beginning to end. DPP learning parameters γ, η and K were set to 0.99, 0.001 and 100 respectively. The compliance threshold was set to 0.9.



(a) Paths taken by users in simulation.

(b) Paths taken by users in the UTS Data Arena.

Fig. 11: Resultant Figure-8 paths from simulation and in the UTS Data Arena. The gray circles have a thickness of 0.5m.

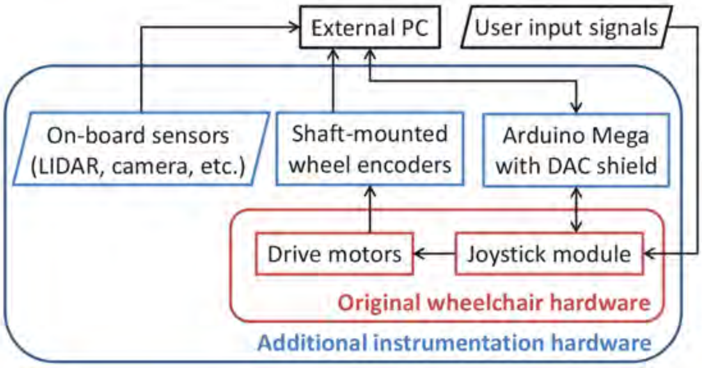

External PC
User input signals
On-board sensors (LIDAR, camera, etc.)
Shaft-mounted wheel encoders
Arduino Mega with DAC shield
Drive motors
Joystick module
Original wheelchair hardware
Additional instrumentation hardware

Fig. 12: High level schematic of the UTS instrumented wheelchair.

3.1 Comparison of Simulation to Real PMD

Five able users drove around a 4m by 2m Figure-8 in both a computer simulation and on the real PMD within the UTS Data Arena (Fig. 10), an augmented reality/motion capture facility capable of real-time millimeter accuracy 6-DOF marker tracking which serves as a localization ground truth. The UTS wheelchair (schematic in Fig. 12) is an Invacare Roller M1 fitted with a Hokuyo UTM-30LX laser scanner, shaft-mounted wheel encoders, and an Arduino to both read and write joystick signals. For additional information about modelling of the PMD's control, see [32]. The resultant paths are shown in Figure 11. The simulation GUI measured $26 \times 13cm$, and the users were free to adjust the computer's position before attempting the task. Driving metrics from these experiments are presented in Table 1. The closenesses in task completion time and input magnitude, and the low path deviation, indicate a strong similarity in both user performance and task relevance. A similar level of steering entropy and angular jerk was also observed, representing a similar level of task effectiveness between both experiments. Given the similarities in the results from both experiments the authors put forward that the simulation is a viable evaluation setting for the wheelchair in terms of human factors. With other phenomena such as wheel slippage across varying ground surfaces beyond the scope of such evaluations, it can be seen that users perform similarly in terms of task completion and efficiency.

Table 1: Driving metrics from the Figure-8 experiments.

Metric	Simulated	Real
Distance travelled (*m*)	13 ± 0.4	12.7 ± 0.8
Avg. path deviation (*m*)	0.05 ± 0.01	0.07 ± 0.006
Steering entropy	0.56 ± 0.03	0.61 ± 0.04
Avg. angular jerk (rad/s^3)	0.01 ± 0.01	0.02 ± 0.01
Avg. forward joystick signal	0.46 ± 0.1	0.41 ± 0.1
Time taken (*s*)	42 ± 8	46 ± 6

3.2 Simulation Experiment

The assistive framework is first assessed in simulation with 10 able-bodied test users, utilizing a course (Fig. 13) seen in [33]. This course significantly differs to the home environment from which training data was gathered for model building purposes.

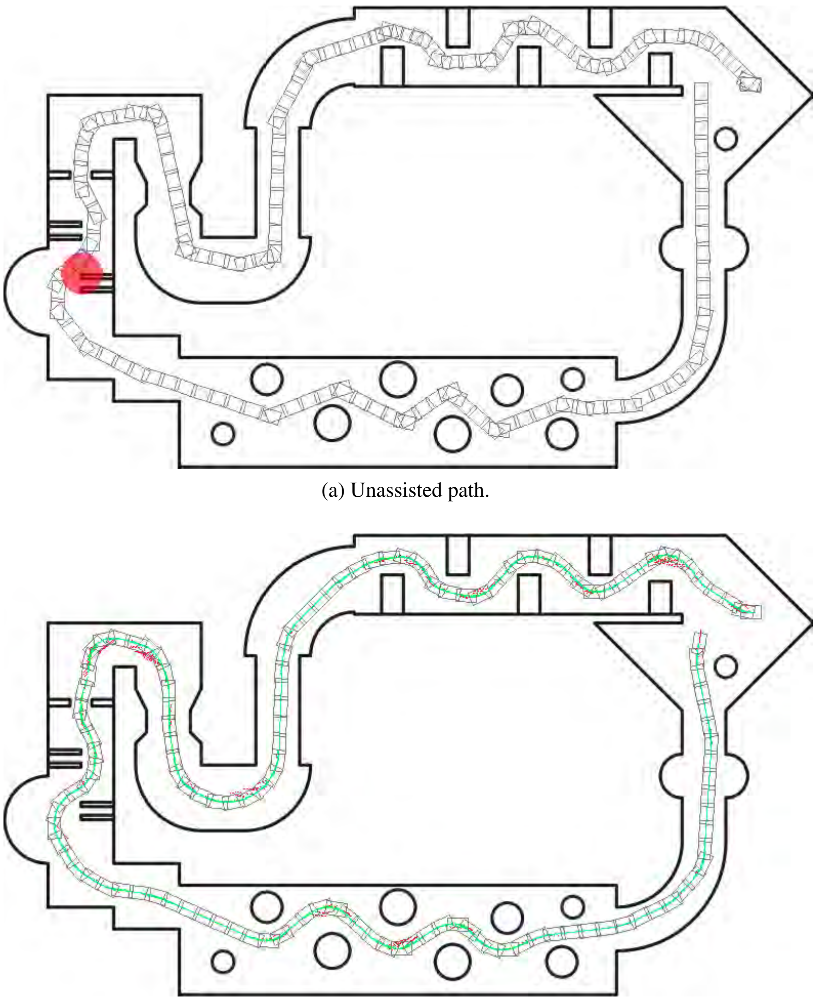
PMD users suffering from limb paresis or other motor skill losses may only be able to provide rough indications [33] of desired direction. To imitate this coarseness in able users, their joystick input signals were discretized amongst 5 evenly spaced joystick orientations. Each test user drove around the test course twice with such hampered input signals; without assistance, then with the assistive framework in operation.

3.3 Real Experiment

A 63 year old female volunteered to evaluate the framework on the UTS wheelchair [2]. Due to complications from back injuries, she is unable to walk without heavy reliance upon dual walking canes and has been considering PMD prescription. Her disabilities in conjunction with relative inexperience pose a significant challenge in maneuvring the PMD safely.

Experimentation took place on the publicly accessible campus of a college neighboring UTS with which the volunteer was already familiar, following a 10 minute acclimation period after which the test user deemed herself sufficiently confident. As shown in Fig. 14, the route featured several doorways and narrow corridors, as well as large open areas. Foot traffic at the time was sufficiently sparse that no moving entities had an effect on the user or PMD behavior; this is reasonable given both the tendency of pedestrians to avoid PMDs, and the framework's significantly truncated operating envelope.

[2] Video available at `http://youtu.be/EPjV2dCe46E`.



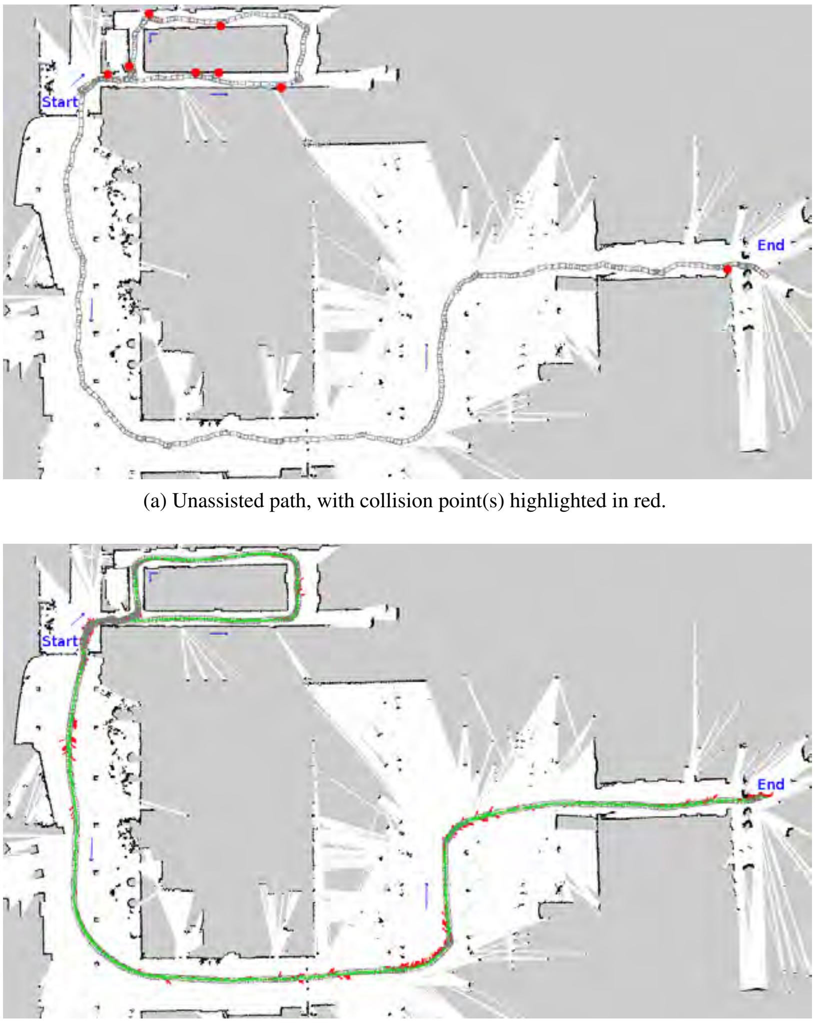
(a) Unassisted path.

(b) Assisted path.

Fig. 13: Comparison between unassisted and assisted paths (gray) of an able volunteer with simulated input disability, driving counter-clockwise in a $21 \times 37m$ simulated environment starting from the top-right corner. Additional details for each figure are available in the captions of Figure 14.

4 Results

For quantitative comparison, all 20 simulated runs were ended upon re-entering the starting ‘diamond’. From [22] several task metrics are evaluated (Table 2) that do not consider deviation from some optimal route, to uphold the belief that users should



(a) Unassisted path, with collision point(s) highlighted in red.

(b) Assisted path with this paper's framework.

Fig. 14: Comparison between unassisted and assisted paths (gray) taken by the disabled volunteer on the UTS wheelchair, with rectangular frames representing PMD footprint. On the right, compliant DPP path points are drawn in green, with noncompliant remainders of paths in red. The area shown in these figures measures approx. $97 \times 58m$. White regions represent free space, black regions represent perceived obstacles and grey regions are unknown due to occlusions. PMD localization and map building via Hector SLAM [34].

not be penalized for particular driving preferences such as keeping to the right of corridors.

Table 2: Driving metrics from simulation experiment.

Metric	Unassisted	Assisted
Avg. angular jerk (rad/s^3)	0.07 ± 0.03	0.03 ± 0.02
Time taken (s)	166 ± 25	146 ± 16
Distance travelled (m)	96 ± 1	97 ± 2
Total collisions	5	0
Steering entropy	0.57 ± 0.02	0.55 ± 0.03

Table 3: Driving metrics from real experiment.

Metric	Unassisted	Assisted
Avg. angular jerk (rad/s^3)	0.03 ± 0.03	0.025 ± 0.03
Time taken (s)	566	252
Distance travelled (m)	238	220
Total collisions	8	0
Steering entropy	0.57	0.6

The same metrics extracted from the real experiment follow in Table 3. From the simulation experiments, the angular jerk decreased slightly with the assistive framework; the paths taken by the PMD also appear smoother, particularly noticeably in the slalom portions on the left sides of Figures 13a-13b. As the real test course largely consisted of straight corridors, the improvements in platform jerk was less noticeable in this experiment compared to the change seen in the simulated experiments.

The distances travelled in both experiments are similar with or without assistance, however the course completion times taken by the users in simulation were noticeably shorter with a reduction of over 10%. The difference in completion times is far larger in the real experiment due to several factors. Firstly the user lost speed scraping along the walls on several occasions, which also resulted in several stops for recovery. By contrast, such contact is not handled by the simulator and no hindrance was incurred as a result. Secondly when unassisted the user drove at an average linear velocity of 0.6 m/s when moving freely compared to 0.85 m/s while assisted, whereas from the simulation experiments less disparate average linear velocities of 0.59 ± 0.07 and 0.67 ± 0.07 m/s were observed. There were also no collisions in both simulation and real experiments when users drove with assistance, whereas without assistance several collision events occurred.

4.1 Steering Entropy

Introduced in [35] to assess driver workload, steering entropy has also been taken as a measure of task effectiveness [23]. For each input steering angle u_t in a time series, an input prediction error e_t is taken as the difference between u_t and a second order Taylor approximation $\hat{u}_t$:

$$\begin{aligned}\hat{u}_t &= u_{t-1} + (u_{t-1} - u_{t-2}) \\ &\quad + \frac{1}{2}\big((u_{t-1} - u_{t-2}) - (u_{t-2} - u_{t-3})\big).\end{aligned} \tag{8}$$

A frequency distribution P of all e is then discretized into 9 bins [35], of which the Shannon entropy H is taken as the steering entropy of the user over the entire time series:

$$H = \sum_i -P_i \times log_9 P_i. \tag{9}$$

The steering entropy observed between the able users with simulated disabilities and the disabled volunteer are very similar at approximately 0.57 (Tables 2-3), indicating that the simulated disability provided a level of task impedance comparable to the disabled volunteer's various health issues. However this does not imply that the her disabilities were accurately replicated; rather that the able users experienced a similar difficulty in providing task effective inputs.

4.2 Assistive Framework Metrics

For evaluating predicted paths only an initial segment truncated by the path's first incompliant point is considered: given a path point p generated from time-step t, PMD positions $\boldsymbol{X}^*_{1:T}$ and user inputs $\boldsymbol{Y}^*_{1:T}$, incompliance is defined as the compliance estimator falling below the experimental threshold for (p',$\boldsymbol{Y}^*_{t:T}$), where p' is p w.r.t. $\boldsymbol{X}^*_{t:T}$. Table 4 documents several metrics directly concerning predictive performance from both simulated (14,664 samples) and real (2,517 samples) experiments. Paths can be evaluated in isolation due to the lack of reliance upon past or expected future data.

Table 4: Framework metrics from both experiments.

Metric	Simulated	Real
Path compliance (%)	64 ± 33	56 ± 40
PMD path RMSE (m)	0.04 ± 0.03	0.1 ± 0.07

Although the observed percentage-wise path compliance and utilisation of each path is arguably equal to or greater than in the work by [20], it is worth noting that this does not necessarily translate into superior overall performance given that both works can always provide new paths at a given instant. The RMSE between compliant path points and PMD positions indicates reliable performance of Pure Pursuit.

4.3 User Impressions

When asked about her experience, the disabled volunteer stated that she received the impression that the PMD was “pulling” towards empty spaces, most noticeably in restrictive areas, and felt more confident knowing her inputs were buffered. The authors were also informed that the integrated joystick was awkwardly positioned, and that she felt it negatively impacted her performance. This has been previously documented in [36], and the authors are currently exploring alternatives.

4.4 Evaluation of Generalization in Path Prediction

In order to evaluate alignment between predicted paths and natural human behaviour, the framework was run offline (Fig. 15) for all $10,948$ driving samples from [13] without retraining from the environment simulated in Figure 9. These samples were collected from several able users freely driving the UTS wheelchair to various long-term destinations. For each path point planned over the course of a single driving sequence, an error is recorded as the minimum distance from the path point to the nearest recorded PMD position in the sequence. In total $132,917$ path points were generated, with a mean error of $0.068 \pm 0.081m$.

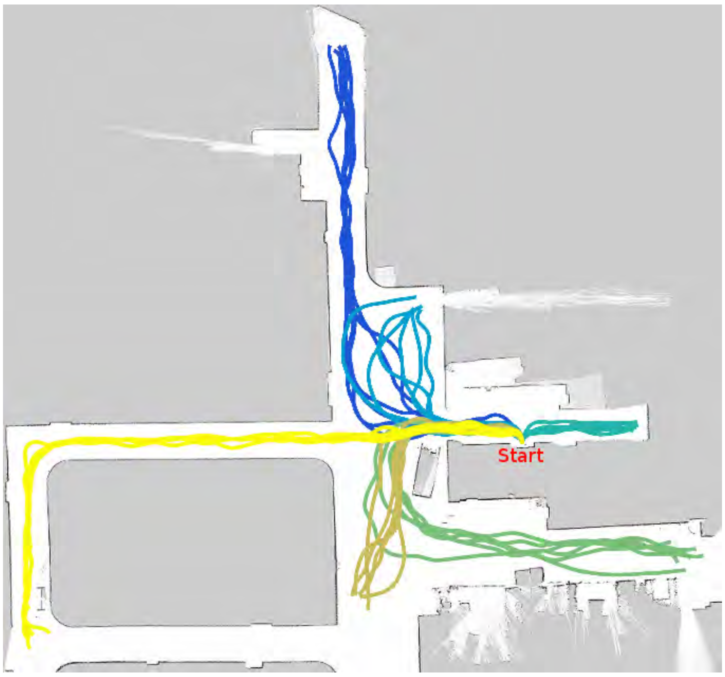
With a null hypothesis of an accuracy of $0.1m$, $p < 0.01$ is obtained. $0.1m$, equivalent to the width of two cells in the path planning grid-world, is taken here as it is the average path tracking error encountered during the real experiment. This result indicates that the framework is capable of capturing user intentions, and can also produce natural human-like paths under conditions not seen in training data.

5 Conclusions

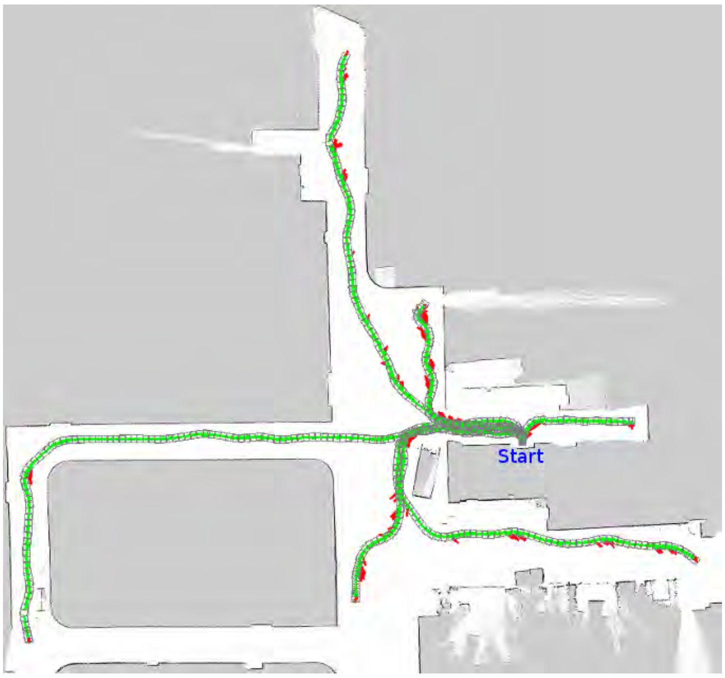
Besides their amenability to be modelled upon demonstration data, a significant advantage of actively assistive mobility systems is the ability to naturally encompass maneuvres which would fail under the strict collision avoidance protocols that form a core component of passively assistive systems. This paper presents a framework allowing for the utilisation of expert demonstration data to serve as the basis of short-term intention estimation and path planning behaviors for users of sensor-equipped powered mobility aids, in order to provide active navigational assistance in areas that have not been explored and mapped *a-priori*.

Testing in environments not explored in training data by both a disabled volunteer on the UTS semi-autonomous wheelchair, and in simulation by able users subject to a comparable level of impedance, reveal strong correlations between desired wheelchair behavior and predicted paths. Although trials with disabled individuals were limited to a single user, the comprehensive testing with a typical end-user suited for PMD prescription demonstrates how intelligent PMD assistance would have significant benefits.

Avenues for further investigation include the application of deep learning methodologies towards both intention estimation and path planning as these techniques have



(a) Paths taken by experienced users on the UTS wheelchair.



(b) Examples of predicted paths along a route to each destination.

Fig. 15: 10,948 driving samples (left) were recorded from a range of experienced users in [13]. The framework is then run offline for all samples (examples on right, with additional details available in the caption of Figure 14) to evaluate the ability to accurately plan paths close to those taken by experienced users driving freely. The area shown in these Figures measures approx. $60 \times 55m$.

been proven for building complex models given adequate data, as well as further experimentation with a greater test user base.

Appendix: Nomenclature

Table 5: For Section 2.2.

Symbol	Definition
$\boldsymbol{X}$	Cartesian PMD pose data
$\boldsymbol{Y}$	2D joystick input signal data
$\boldsymbol{Z}$	polar planar laser scanner data
$\boldsymbol{G}$	termination points of local paths

Table 6: For Section 2.3.

Symbol	Definition
$\mathcal{S}$	finite set of discrete states
$\mathcal{A}$	finite set of discrete actions
$\mathcal{T}^{a}_{ss'}$	transition likelihood from state s to s' via action a
$\mathcal{R}(s,s',a)$	reword obtained by transitioning from s to s' via a
γ	discount factor $\in (0,1)$
$\pi(a\|s)$	policy representing probability of a being performed at s
$V_{\pi}(s)$	expected reward returned from following π starting at s
K	number of iterations
η	parameter $\in (0,1)$ controlling policy update smoothness
$\Psi_k(s,a)$	action preference function at iteration k
$\mathcal{M}_{\eta}\Psi_k(s)$	Boltzmann softmax operator of Ψ_k at s for all $a \in \mathcal{A}$

Acknowledgment

This work was supported by the 2015-2017 JSPS Bilateral Joint Research Projects (Open Partnership) for collaboration between NAIST and UTS. The authors also wish to express their sincere gratitude to all who took part in the experiments.

References

1. United Nations: World population ageing 2015. Technical report, United Nations, New York (2015)
2. Tegart: Smart technology for healthy ageing. Technical report, ATSE (2010)
3. Queensland Department of Health: Mass 24 - home access checklist - power-drive wheelchairs (pwc). `https://www.health.qld.gov.au/mass/documents/form-mass24-home-access-safety.pdf` (2016) Accessed: 2016-09-12.
4. Lankenau, A., Meyer, O., Krieg-Bruckner, B.: Safety in robotics: the bremen autonomous wheelchair. In: Advanced Motion Control, 5th International Workshop on. (Jun 1998) 524–529

5. Simpson, R.C., Levine, S.P., Bell, D.A., Jaros, L.A., Koren, Y., Borenstein, J.: Navchair: An assistive wheelchair navigation system with automatic adaptation. Assistive Technology and Artificial Intelligence: Applications in Robotics, User Interfaces and Natural Language Processing (1998) 235–255
6. Borenstein, J., Koren, Y.: The vector field histogram-fast obstacle avoidance for mobile robots. IEEE Transactions on Robotics and Automation **7**(3) (Jun 1991) 278–288
7. Ashley, D., Ashley, K., Alqasemi, R., Dubey, R.: Semi-autonomous mobility assistance for power wheelchair users navigating crowded environments. In: 2017 International Conference on Rehabilitation Robotics (ICORR). (July 2017) 1025–1030
8. Devigne, L., Narayanan, V.K., Pasteau, F., Babel, M.: Low complex sensor-based shared control for power wheelchair navigation. In: 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). (Oct 2016) 5434–5439
9. Goil, A., Derry, M., Argall, B.D.: Using machine learning to blend human and robot controls for assisted wheelchair navigation. In: Rehabilitation Robotics, IEEE International Conference on. (June 2013) 1–6
10. Urdiales, C., Peula, J.M., Fdez-Carmona, M., Barrué, C., Pérez, E.J., Sánchez-Tato, I., del Toro, J.C., Galluppi, F., Cortés, U., Annichiaricco, R., Caltagirone, C., Sandoval, F.: A new multi-criteria optimization strategy for shared control inwheelchair assisted navigation. Autonomous Robots **30**(2) (2011) 179–197
11. Simpson, R.C., Levine, S.P.: Automatic adaptation in the navchair assistive wheelchair navigation system. IEEE Transactions on Rehabilitation Engineering **7**(4) (Dec 1999) 452–463
12. Patel, M., Miro, J.V., Dissanayake, G.: A probabilistic approach to learn activities of daily living of a mobility aid device user. In: Robotics and Automation, IEEE International Conference on. (2014) 969–974
13. Matsubara, T., Miro, J., Tanaka, D., Poon, J., Sugimoto, K.: Sequential intention estimation of a mobility aid user for intelligent navigational assistance. In: Robot and Human Interactive Communication, 24th IEEE International Symposium on. (Aug 2015) 444–449
14. Huntemann, A., Demeester, E., Poorten, E., Van Brussel, H., De Schutter, J.: Probabilistic approach to recognize local navigation plans by fusing past driving information with a personalized user model. In: Robotics and Automation, IEEE International Conference on. (May 2013) 4376–4383
15. Escobedo, A., Spalanzani, A., Laugier, C.: Using social cues to estimate possible destinations when driving a robotic wheelchair. In: Intelligent Robots and Systems, IEEE/RSJ International Conference on. (Sept 2014) 3299–3304
16. Carlson, T., Demiris, Y.: Collaborative control for a robotic wheelchair: Evaluation of performance, attention, and workload. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) **42**(3) (June 2012) 876–888
17. Derry, M., Argall, B.: Automated doorway detection for assistive shared-control wheelchairs. In: Robotics and Automation, IEEE International Conference on. (May 2013) 1254–1259
18. Narayanan, V.K., Spalanzani, A., Babel, M.: A semi-autonomous framework for human-aware and user intention driven wheelchair mobility assistance. In: Intelligent Robots and Systems, IEEE/RSJ International Conference on. (Oct 2016) 5775–5780
19. Fox, D., Burgard, W., Thrun, S.: The dynamic window approach to collision avoidance. Robotics Automation Magazine, IEEE **4**(1) (Mar 1997) 23–33
20. Huntemann, A., Demeester, E., Vanacker, G., Vanhooydonck, D., Philips, J., Brussel, H.V., Nuttin, M.: Bayesian plan recognition and shared control under uncertainty: assisting wheelchair drivers by tracking fine motion paths. In: Intelligent Robots and Systems, IEEE/RSJ International Conference on. (2007) 3360–3366
21. Poon, J., Cui, Y., Miro, J.V., Matsubara, T., Sugimoto, K.: Local driving assistance from demonstration for mobility aids. In: Robotics and Automation, IEEE International Conference on. (May 2017) 5935–5941
22. Urdiales, C., Fdez-Carmona, M., Peinado, G., Sandoval, F.: Metrics and benchmarking for assisted wheelchair navigation. In: Rehabilitation Robotics, IEEE International Conference on. (2013)
23. Yang, X., Sreenath, K., Michael, N.: A framework for efficient teleoperation via online adaptation. In: Robotics and Automation, IEEE International Conference on. (May 2017) 5948–5953
24. Azar, M.G., Gómez, V., Kappen, H.J.: Dynamic policy programming. The Journal of Machine Learning Research **13**(1) (2012) 3207–3245
25. Standards Australia: As 1428.1-2009-design for access and mobility-general requirements for access-new building work. Technical report (2009)
26. Orr, M.: Introduction to radial basis function networks. Technical report, University of Edinburgh, Scotland (1996)

27. Hagan, M.T., Demuth, H.B., Beale, M.H., De Jesus, O.: Neural Network Design (2nd Edition). (2014)
28. Jahanshahi, H., Sari, N.N.: Robot path planning algorithms: A review of theory and experiment. CoRR **abs/1805.08137** (2018)
29. Ballesteros, J., Urdiales, C., Velasco, A.B.M., Ramos-Jimnez, G.: A biomimetical dynamic window approach to navigation for collaborative control. IEEE Transactions on Human-Machine Systems **PP**(99) (2017) 1–11
30. Sutton, R.S., Barto, A.G.: Reinforcement learning: An introduction. MIT press, Cambridge (1998)
31. Coulter, R.: Implementation of the pure pursuit path tracking algorithm. Technical report, Carnegie Mellon University, Pittsburgh (1990)
32. Poon, J., Miro, J.V.: Learning by demonstration for co-operative navigation with assistive mobility devices. In: Robotics and Automation, Australasian Conference on. (Dec 2015)
33. Vanhooydonck, D., Demeester, E., Hüntemann, A., Philips, J., Vanacker, G., Van Brussel, H., Nuttin, M.: Adaptable navigational assistance for intelligent wheelchairs by means of an implicit personalized user model. Robotics and Autonomous Systems **58**(8) (Aug 2010) 963–977
34. Kohlbrecher, S., Meyer, J., Stryk, O.V., Klingauf, U.: A flexible and scalable slam system with full 3d motion estimation. In: Safety, Security and Rescue Robotics, IEEE International Symposium on, IEEE (November 2011)
35. Nakayama, O., Futami, T., Nakamura, T., Boer, E.R.: Development of a steering entropy method for evaluating driver workload. Technical report, SAE International (1999)
36. Esquenazi, A.: Why joystick location matters for power wheelchair users. `http://www.homecaremag.com/news/why-joystick-location-matters-power-wheelchair-users` (2016) Accessed: 2018-06-22.