Optimal Control Approach for Active Local Driving Assistance in Mobility Aids

Yunduan Cui[1], James Poon[2], Jaime Valls Miro[2], Takamitsu Matsubara[1], and Kenji Sugimoto[1]
[1] Nara Institute of Science and Technology (NAIST)
[2] University of Technology Sydney (UTS)

1. Introduction

This paper presents a system capable of providing active short-term driving assistance of robot wheelchair (Fig. 1). Combining the intelligence of active assistance with the freedom of location independence, A local intention estimator is proposed to predict the user's intention according to sensor data without map information. An optimal control method with smooth policy update is then applied to generate more intuitive and safer trajectories by smoothly combining the predicted user intention, current sensor data and expert training data from a human demonstrator.

Figure 1: Semi-autonomous wheelchair for assistive control system.

2. Local Intention Estimation

Conventional active navigational assistance with robot wheelchairs [1, 2, 3] tend to rely on *a-priori* map building of the entire space that the system is expected to operate in. Selected locations of interest in this space correspond to destinations that can be perceived as longer-term targets. The primary disadvantage of this approach is the requirement of maintaining considerable spatial maps to accomodate long-term assistive planning and the computational complexity associated to that. Moreover, it is effectively infeasible to attempt to map and provide targets for all the places a robot wheelchair user would go for any sizeable environment.

A system capable of operating in a short-term "local" frame of reference from the immediately available sensor data, yet still able to generalize beyond known regions and pre-established destinations within these appears an attractive proposition. This however raises the issue of dimensionality.

Since local intention estimation cannot be practically resolved as a multi-class classification exercise like in conventional intention estimation because of the high dimensionality of sensor data required to deal with no positional data and increase number of discrete likely intentions within a user's immediate space, the problem is hereby treated from an measurement element-wise point of view instead.

Demonstration data contains expert actions $\mathbf{m}_{1:D}$ of 2D Cartesian joystick positions, odometry information $\mathbf{o}_{1:D}$ containing the platform's Cartesian position, heading and linear and angular velocities, and sensor data $\mathbf{y}_{1:D}^{1:N}$ where $y^{1:N}$ denotes a sweep of distances to obstacles across the laser scanner's 180° horizontal field of view, downsampled to N evenly spaced measurements. The objective of this pre-processing step is thus deriving an element-wise training data container $\Omega_{1:N}$ where each element is allocated one range measurement within y, and also its corresponding angular window.

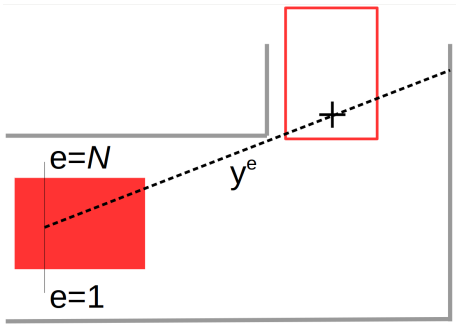

Figure 2: The element encompassing the end pose (+) receives a training set due to subsequent loss of line of sight.

Demonstration data is first decomposed into series of short trajectories which terminate when any of the following 'exit' criteria are met:

- Trajectory length exceeds short-term length bound

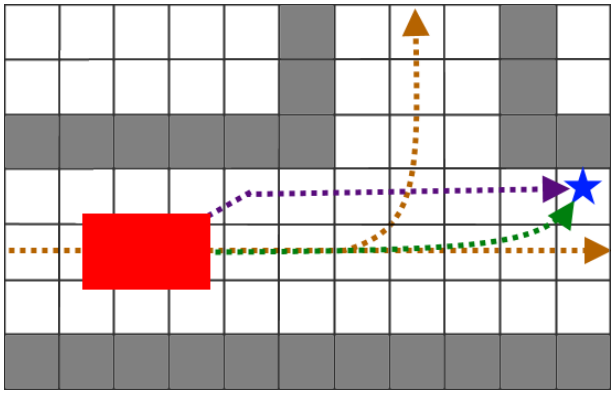

Figure 3: Illustration of the generated path based on the demonstration path (brown). When a new target is set as blue star, DPP generates safer, more intuitive trajectories in green (by DPP), compared with the figurative optimal geometric trajectory generated by Q-learning in purple.

- No line of sight to start point of trajectory
- Instance of inflection in forward/reverse joystick axis

Then for each demonstration data instance t along all trajectories, the element Ω_e whose angular coverage encompasses the trajectory's termination point at $\mathbf{o}_t$ receives a training set of $(\mathbf{m}_t, \mathbf{y}_t^e) = 1$. After populating Ω, each element then has a Radial Basis Function Network [5] trained as a one-class classifier aiming to infer the likelihood of an input belonging to a single class when given only positive training examples, as opposed to conventional discriminative classification. As the likelihood 1 provides a solution to the training data within Ω, a strong 0 prior is employed to reduce overfitting.

3. Path Generator

The path generator periodically generates assistive paths based on the environmental sensor information, the estimated user intention and expert demonstration. The current local occupancy grid is translated to a grid world with discrete states and actions. The obstacle areas detected by sensors are assigned a low reward, while an area along the direction of the estimated user intention prior to any detected obstruction is assigned a maximal reward. The path generator then selects trajectories which have a close termination point to this goal area from the expert demonstration. Dynamic Policy Programming (DPP) [4] is applied to generate a path smoothly mixed with expert demonstration while it is difficult to impart human behaviors to conventional optimal control algorithms (e.g., Q-learning [6]) without complicated and carefully hand-crafted reward functions (shown in Fig. 3).

DPP achieves smooth policy update by adding Kullback-Leibler divergence between the optimal policy π and the baseline policy $\bar{\pi}$ into the reward function as one regularization term to control the policy deviation. The Bellman equation of optimal value function then becomes:

$$V_{\bar{\pi}}^{*}(\boldsymbol{s}) = \max_{\pi} \sum_{\substack{\boldsymbol{a}\in\mathcal{A} \\ \boldsymbol{s}'\in\mathcal{S}}} \pi(\boldsymbol{a}|\boldsymbol{s}) \left[\mathcal{T}_{\boldsymbol{ss}'}^{\boldsymbol{a}} (r_{\boldsymbol{ss}'}^{\boldsymbol{a}} + \gamma V_{\bar{\pi}}^{*}(\boldsymbol{s}')) - \frac{1}{\eta} \log\left(\frac{\pi(\boldsymbol{a}|\boldsymbol{s})}{\bar{\pi}(\boldsymbol{a}|\boldsymbol{s})}\right) \right].$$

where $\mathcal{S} = \{\boldsymbol{s}_1, ..., \boldsymbol{s}_n\}$ is the state set, $\mathcal{A} = \{\boldsymbol{a}_1, ..., \boldsymbol{a}_m\}$ is the action set, $\mathcal{T}_{\boldsymbol{ss}'}^{\boldsymbol{a}}$ represents the transition probability from $\boldsymbol{s}$ to $\boldsymbol{s}'$ under the a and $r_{\boldsymbol{ss}'}^{\boldsymbol{a}} = \mathcal{R}(\boldsymbol{s}, \boldsymbol{s}', \boldsymbol{a})$ is the corresponding reward, $\gamma \in (0,1)$ is the discount factor. The policy $\pi(\boldsymbol{a}|\boldsymbol{s})$ denotes the probability of taking the action $\boldsymbol{a}$ under the state $\boldsymbol{s}$.

To get the solution of this Bellman equation, the action preferences [7] for all state action pairs $(\boldsymbol{s}, \boldsymbol{a}) \in \mathcal{S} \times \mathcal{A}$ in the $(t+1)$th are defined according to [4]:

$$\Psi_{t+1}(\boldsymbol{s}, \boldsymbol{a}) = \frac{1}{\eta} \log \bar{\pi}^{t}(\boldsymbol{a}|\boldsymbol{s}) + \sum_{\boldsymbol{s}'\in\mathcal{S}} \mathcal{T}_{\boldsymbol{ss}'}^{\boldsymbol{a}} (r_{\boldsymbol{ss}'}^{\boldsymbol{a}} + \gamma V_{\bar{\pi}}^{t}(\boldsymbol{s}')).$$

Instead of the optimal value function, DPP learns optimal action preferences to determine the optimal control policy throughout the state-action space. The DPP recursion $\Psi_{t+1}(\boldsymbol{s}, \boldsymbol{a}) = O\Psi_t(\boldsymbol{s}, \boldsymbol{a})$ calculated by:

$$\Psi_t(\boldsymbol{s}, \boldsymbol{a}) - \mathcal{M}_\eta \Psi_t(\boldsymbol{s}) + \sum_{\boldsymbol{s}'\in\mathcal{S}} \mathcal{T}_{\boldsymbol{ss}'}^{\boldsymbol{a}} (r_{\boldsymbol{ss}'}^{\boldsymbol{a}} + \mathcal{M}_\eta \Psi_t(\boldsymbol{s}')) \quad (1)$$

with the Boltzmann soft-max operator $\mathcal{M}_\eta \Psi_t(\boldsymbol{s})$:

$$\mathcal{M}_\eta \Psi_t(\boldsymbol{s}) = \sum_{\boldsymbol{a}\in\mathcal{A}} \frac{\exp(\eta \Psi_t(\boldsymbol{s}, \boldsymbol{a})) \Psi_t(\boldsymbol{s}, \boldsymbol{a})}{\sum_{\boldsymbol{a}'\in\mathcal{A}} \exp(\eta \Psi_t(\boldsymbol{s}, \boldsymbol{a}'))}. \quad (2)$$

4. Simulation

Initial testing was conducted within the Stage (`wiki.ros.org/stage`) robot simulator in a typical home environment ($20 \times 10\ m^2$) with furniture added to increase the difficulty of moving through the available space. Training data followed the trajectories depicted in Fig. 4 and was obtained via a healthy demonstrator controlling a simulacrum of the real wheelchair platform.

During the trial an able-bodied volunteer was asked to drive the simulation poorly; despite poor control from the user, the platform was able to manoeuvre safely through the significantly cluttered environment as depicted by the final path shown in Fig. 5, passing through multiple narrow doorways and turning in place under the supervision of the proposed active assistive strategy, softly following the training trajectories' driving style. Fig. 6 demonstrates the ability of the assistive control system in mitigating potentially hazardous user inputs. The end effect is that the user is effectively allowed more control when their actions are deemed to be safe, however excessive inputs along either axis result in a stricter following of expert trajectories.

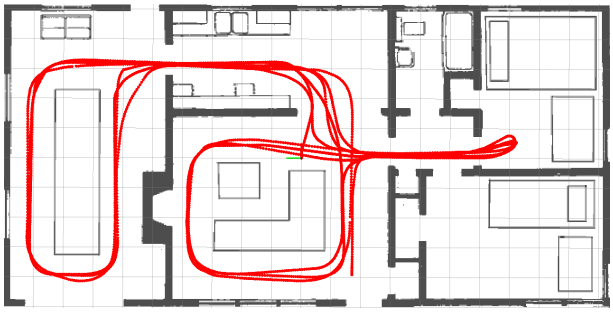

Figure 4: Expert trajectory in simulation.

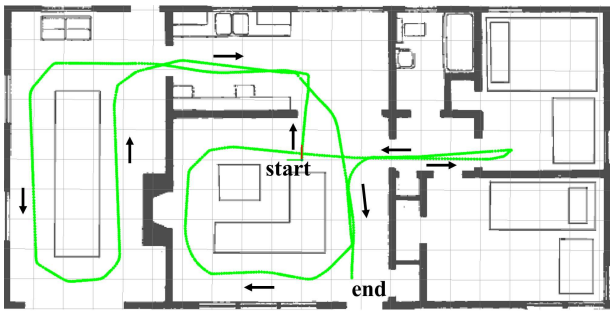

Figure 5: Assistive control trajectory in simulation.

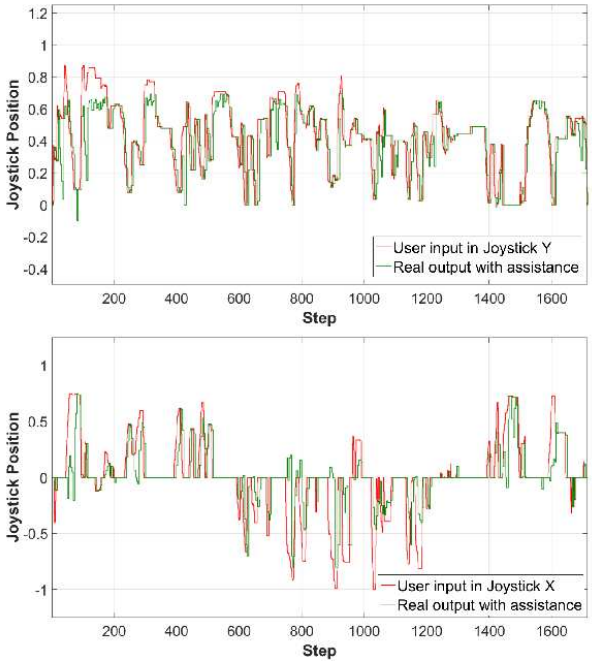

Figure 6: Comparison between user input and real output with active assistance in simulation, corresponding to the path shown in Fig. 5.

Figure 7: The lobby level of University of Technology Sydney (UTS) building 11.

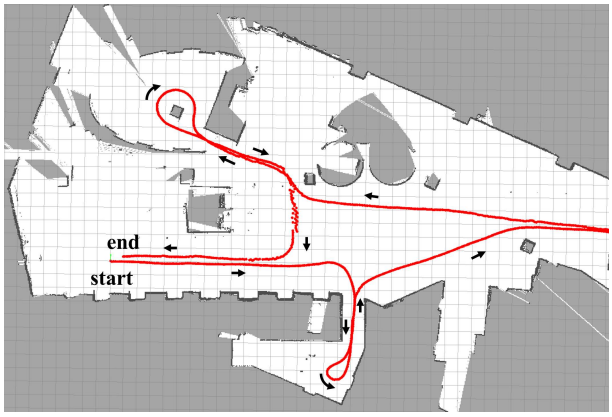

Figure 8: Expert trajectory in real experiment.

5. Real Experiment

Real experiment was carried out on the lobby of a building part of UTS (50×30 m^2, Fig. 7). Demonstration data used for training was the driving data collected during a single a-priori map building exercise (Fig. 8). The built map is only for visualization of results and was not utilized in the experiment itself. Apart from running actual hardware drivers in place of Stage, the computing setup was the same configuration as in the simulated experiment. In this experiment the test driver drove in a more aggressive manner than in the simulated trial. The driving assistance is able to generalize a portion of the travel across the center area of the map which was unexplored by the training data, as well as safely permit movement in narrow spaces and tight areas as per the simulation experiment. Due to the worsened inputs from user, the deviations from the safer assisted control output are more significant (Fig. 10); however the driving assistance accepted the user's inputs when they were considered safe enough for the immediate environment surrounding the platform.

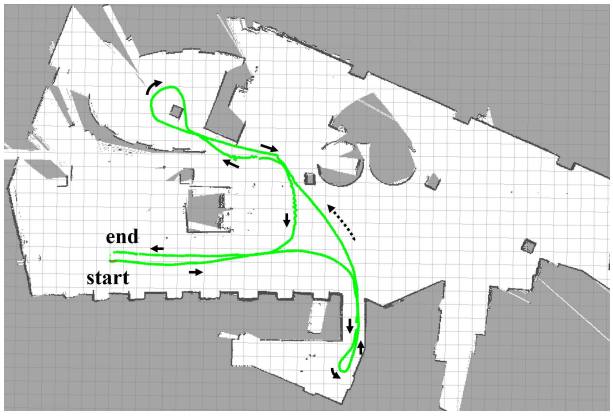

Figure 9: Assistive control trajectory in real experiment.

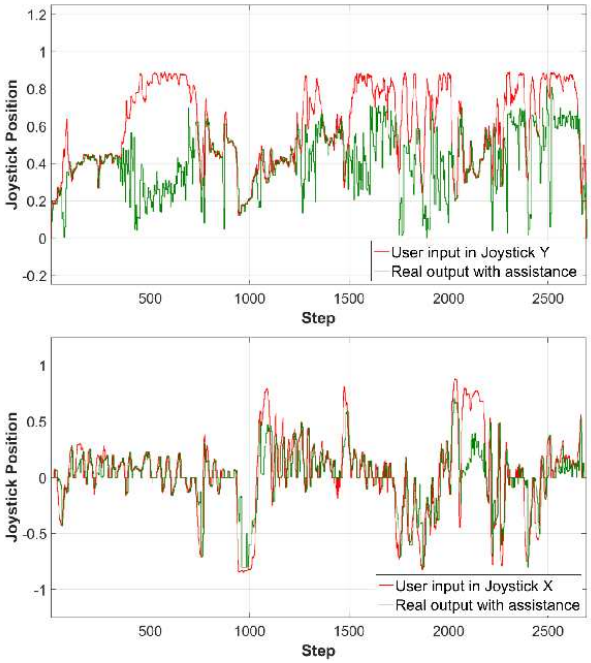

Figure 10: Comparison between user input and real output with assistance in real experiment.

6. Conclusion and Future Work

This work has proposed a system capable of learning both local user intention estimation and planning of short-term trajectories from expert demonstration, without the conventional reliance upon static longer-term destinations in a given map. The user intention is predicted by a map-free local estimator using several Radial Basis Function Network which reduces the computational complexity due to the curse of dimensionality. An optimal control algorithm with smooth policy update is applied to generated intuitive and safer trajectories by smoothly combining the predicted user intention, current sensor data and expert demonstrations. Experimental results in both simulation and in a real building show that the system is able to accurately infer user intentions in order to both follow training data in relatively confined spaces, and generalize trajectories through regions without training data coverage.

The primary task for future development is moving towards the removal of dependence on position information for path lookup, requiring a mechanism that can rapidly match incoming sensor data against a database of training data instances in order to obtain likely local paths to consider for DPP planning.

7. Acknowledgment

This work is supported by NAIST ' s Global Initiative Program 2015-2016, and the Japan Society for the Promotion of Science's open partnership joint project 2015-2017 grant for collaboration between NAIST and UTS.

Reference

[1] A. Huntemann, E. Demeester, E. Poorten, H. Van Brussel, and J. De Schutter: "Probabilistic approach to recognize local navigation plans by fusing past driving information with a personalized user model", in Robotics and Automation (ICRA), 2013 IEEE International Conference on, pp. 43764383, May 2013.

[2] M. Patel, J. V. Miro, and G. Dissanayake: "A probabilistic approach to learn activities of daily living of a mobility aid device user", in Proceedings of the IEEE International Conference on Robotics and Automation, ICRA Hong Kong, China, pp. 969974, 2014.

[3] T. Matsubara, J. Miro, D. Tanaka, J. Poon, and K. Sugimoto: "Sequential intention estimation of a mobility aid user for intelligent navigational assistance", in Robot and Human Interactive Communication (RO-MAN), 2015 24th IEEE International Symposium on, pp. 444449, Aug 2015.

[4] M. G. Azar, V. G omez, and H. J. Kappen: "Dynamic policy programming", The Journal of Machine Learning Research, vol. 13, no. 1,pp. 3207-3245, 2012.

[5] M. Orr: "Introduction to radial basis function networks", tech. rep., University of Edinburgh, Scotland, 1996.

[6] C. J. Watkins and P. Dayan: "Q-learning", Machine learning, vol. 8, no. 3-4, pp. 279-292, 1992.

[7] R. S. Sutton and A. G. Barto: "Reinforcement learning: An introduction", MIT press Cambridge, 1998.