SmartWheeler: A robotic wheelchair test-bed for investigating new models of human-robot interaction

Joelle Pineau and Amin Atrash
for the SmartWheeler team[1]
School of Computer Science
McGill University
Montreal, QC H3A 2A7
jpineau@cs.mcgill.ca

Abstract

The goal of the SmartWheeler project is to increase the autonomy and safety of individuals with severe mobility impairments by developing a robotic wheelchair that is adapted to their needs. The project tackles a range of challenging issues, focusing in particular on tasks pertaining to human-robot interaction, and on robust control of the intelligent wheelchair. The platform we have built also serves as a test-bed for validating novel concepts and algorithms for automated decision-making onboard socially assistive robots. This paper introduces the wheelchair platform, and outlines technique contributions in four ongoing research areas: adaptive planning in large-scale environments, learning and control under model uncertainty, large-scale dialogue management, and communication protocols for the tactile interface.

Introduction

Many people who suffer from chronic mobility impairments, such as spinal cord injuries or multiple sclerosis, use a powered wheelchair to move around their environment. However, factors such as fatigue, degeneration of their condition, and sensory impairments, often limit their ability to use standard electric wheelchairs.

The SmartWheeler project aims at developing—in collaboration with engineers and rehabilitation clinicians—a prototype of a multi-functional intelligent wheelchair to assist individuals with mobility impairments in their daily locomotion, while minimizing physical and cognitive loads.

Many challenging issues arise in this type of application. First, there are a number of technical issues pertaining to the physical design of the wheelchair; these are only briefly mentioned below. Second there are substantial computation issues pertaining to the control of the wheelchair which require close attention. This paper outlines ongoing work targeting a number of these aspects, ranging from new approaches to path planning, to technical innovations for model learning, to the design of the human-robot control interface.

Beyond its technological components, an essential aspect of this project is a strong collaboration with clinicians, to ensure the definition of goals for the mobility functions, for the patient/wheelchair and environment/wheelchair interactions, as well as for the experimental validation of the smart wheelchair.

Our aim is to show that the robotic wheelchair reduces the physical and cognitive load required to operate the vehicle. We are therefore focusing on high-load situations, such as navigating in confined spaces (e.g. entering/exiting an elevator or a public washroom), stressful situations (e.g. exiting a building during a fire alarm), or unknown environments (e.g. transferring flights through a new airport).

Most of these tasks require basic robot navigation capabilities (mapping, localization, point-to-point motion). We rely substantially on previous technology to implement these functionalities. Some challenges remain pertaining to robust navigation in large-scale environments. To address this, we discuss a novel approach for variable-resolution planning under motion and sensory uncertainty.

We are also concerned with the design of the physical interface between the robotic wheelchair and its user. A key aspect of the patient/wheelchair interface involves creating communication protocols that can ensure high-quality information exchanges, minimizing the ambiguity, and progressively improving effectiveness of the interactions over time. We investigate two such protocols: a voice-based dialogue system, and a tactile/visual interface system. Both are discussed below.

Target population

The goal of this project is to increase the autonomy and safety of individuals with severe mobility impairments by developing a robotic wheelchair that is adapted to their needs.

Through discussions with clinical collaborators at the Centre de réadaptation Constance-Lethbridge and Centre de réadaptation Lucie-Bruneau, two rehabilitation clinics in the Montreal area, we have selected a target population for this work, along with a set of challenging tasks. We have chosen to define the target population based on their *abilities*, rather than their *pathologies*. The motivation for doing so is that we can use uniform measures of performance across the target population, thereby allowing us to gauge the usefulness of the deployed robotic system.

[1] The SmartWheeler team at McGill University includes: Amin Atrash, Jeremy Cooperstock, Robin Jaulmes, Robert Kaplow, Nan Lin, Andrew Phan, Chris Prahacs, Doina Precup and Shane Saunderson.

Individuals of interest will be those who meet the reduced mobility criteria necessary to qualify for a powered wheelchair under the Régie de l'assurance maladie du Québec (the provincial public health board). There are well established guidelines for applying this criteria, and our clinical collaborators have long expertise in evaluating these.

Once an individual is approved for use of a powered wheelchair, s/he may require substantial configuration of the vehicle to achieve maximum usability. In particular, many patients require custom interfaces that go beyond the standard joystick, for example sip-and-puff devices, or pressure sensors that can be activated with minimum head, chin or hand control. Yet despite clinicians' significant customization efforts, control of a powered wheelchair (even with a joystick) remains a significant challenge. According to a recent survey, 40% of patients found daily steering and maneuvering tasks to be difficult or impossible, and clinicians believe that nearly half of patients unable to control a powered wheelchair by conventional methods would benefit from an automated navigation system (Fehr et al., 2000).

Robot platform

SmartWheeler, shown in Figure 1, is built on top of a commercially available Sunrise Quickie Freestyle, to which we have added front and back laser range-finders, wheel odometers, a touch-sensitive graphical display, a voice interface, and an onboard computer. The laser range-finders and odometers are used for navigation and obstacle avoidance. The display, voice interface, and joystick are the main modes of communication with the user. The onboard computer interfaces with the wheelchair's motor control board to provide autonomous navigational commands. Additional devices will be integrated in the future, including stereo vision, IR sensors, and a modified joystick. All hardware and electronic design were performed in-house by staff members at McGill's Center for Intelligent Machines.

Figure 1: SmartWheeler robot platform.

The robot's basic mapping and navigation functions are provided by the Carmen robot navigation toolkit (Montemerlo et al., 2003). This toolkit can be adapted to a variety of robot platforms and has been used in the robotics community for the control of indoor mobile robots in a variety of challenging environments. The toolkit can be used to build a high-resolution 2-D grid-based representation of an indoor environment. When used for online navigation, it provides robust laser-based robot localization, obstacle detection, and path planning.

Carmen is particularly useful for validation of new algorithms because its simulator is known to be highly reliable and policies with good simulation performance can typically be ported without modification to the corresponding robot platform.

However there are some limitations to the current software, and we are actively developing a number of new capabilities, for example:

- Detection and avoidance of negative obstacles (e.g. downward staircase).
- Robust point-to-point planning and navigation.
- Shared control between autonomous controller and human user.
- Adaptive (user-specific) control strategy.
- Adapted interface for low-bandwidth communication.

The remainder of the paper discusses four ongoing areas of research pertaining to this project.

Adaptive planning in large-scale environments

As part of its task domain, the robot will be called upon to navigate robustly in very large environments. There exists a number of well known approaches for robot path planning, however they tend to roughly fall into two classes.

The first group assumes deterministic effects on the part of both the robot and the environment; these methods can therefore scale to high-dimensional domains but are not robust to uncertainty in the motion or sensor model. An example of such algorithm is the variable resolution cell decomposition technique.

The second group considers probabilistic motion effects and sensor readings; these methods are therefore robust to uncertainty, but generally scales poorly and can only handle small environments. An example of such an algorithm is the Partially Observable Markov Decision Process (POMDP) framework.

We aim to combine these two types of techniques, in an attempt to devise a planning approach that affords the computational flexibility of variable resolution techniques with the robustness of POMDs. Before describing our approach, we review briefly the POMDP framework.

The POMDP is defined by the n-tuple: $\{S, A, Z, T, O, R\}$ where S defines the state space, A defines the action space, Z defines the set of observations, $T = Pr(s'|s, a)$ defines the state-to-state transition probabilities (e.g. motion model), $O = Pr(z|s, a)$ defines the probability of seeing each observation (e.g. sensor model), and $R(s, a)$ defines a real-valued reward function. Unlike many traditional planning paradigms, in POMDPs the state of the system is not necessarily known, but can be inferred probabilistically from the observation sequence. To do this, we track a belief state, $b(s) = Pr(s_t = s | z_t, a_{t-1}, z_{t-1}, ..., a_0, z_0, b_0)$, which defines the probability of each state at a given time step t. The goal of the POMDP is to select a sequence of actions such as to maximize the sum of rewards over time. This is defined formally as: $V(b) = R(b, a) + \sum_{b'} T(b, a, b')V(b')$. Further details on POMDP solution techniques can be found in the literature (Kaelbling et al., 1998; Hauskrecht, 2000; Pineau et al., 2006). For the purposes of this paper, it is sufficient to say that planning complexity increases quadratically with the size of the state space. Thus it is crucial to find a good compact state representation to tackle planning in large-scale environments.

With this goal in mind, we have developed a new approach to planning in metric environments with unifies the variable resolution cell decomposition and POMDP approaches. The variable resolution technique allows us to select the appropriate state representation for the environment. This automatically discretizes the robot's navigation space using fine grid resolution near obstacles or goal areas, and coarser discretization in large open spaces. Once we have computed the discretization, we must infer a probabilistic motion model to capture the robot's motion uncertainty. This is done using sampling techniques; the robot runs through a number of trajectories and the motion model is estimated from statistics computed over these trajectories. A probabilistic sensor model is also needed; this is generally crafted by hand based on expert knowledge. In the next section we discuss how to incorporate automated learning in this phase of the approach. Given a variable resolution state representation and corresponding parameterized models, we are able to compute an optimal path planning strategy over the entire state space. This is done using recent POMDP solution techniques (Pineau and Gordon, 2005).

Figure 3 shows a sample map taken from (Carmen, 2006), along with the the state representation that was extracted by our approach. As expected, large rooms receive coarse resolution, whereas hallways and narrow areas show finer resolution. For this map, the use of the variable cell decomposition yielded a 400-fold reduction in computation, compared to using the full map resolution. Further reduction could be achieved at the expense of some loss in performance. Thus the variable resolution approach allows us to trade-off planning time and plan quality in a flexible manner.

Learning and control under model uncertainty

The POMDP framework requires a known parametric model defining the dynamics of the problem domain. The parametric model contains two parts: the *motion* model (or transition probabilities) and the *sensor* model (or observation probabilities). In the section above, we assumed that the motion model was learned by acquiring motion samples and building a parametric representation. We also assumed that the observation model was given by an expert. Both of these assumptions can be problematic in some realistic applications. Learning a model from samples is feasible in cases where data is plentiful and inexpensive, for example when a simulator is available, however the quality of the produced model is only as good as the simulator's fidelity. Alternately, asking a domain expert to specify a parametric model is feasible in some domains, for example in the physical world where there are natural constraints. However in other cases, for example human-robot interaction, acquiring an accurate model can be challenging because of our poor understanding of the dynamics that occur in that domain.

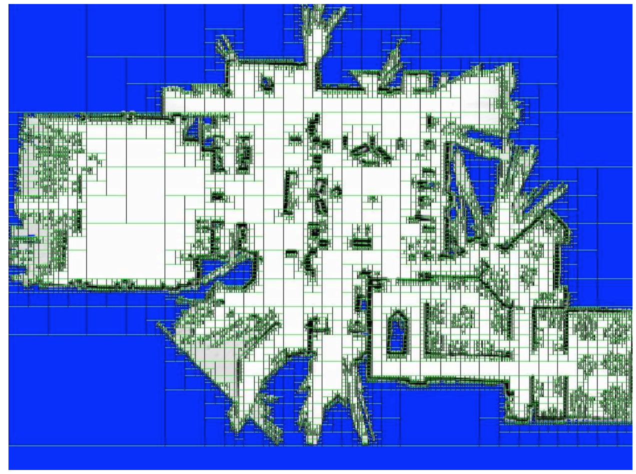


Figure 2: Variable resolution map

Ideally, we would like to combine both expert knowledge and data-drive learning to produce a more flexible approach to model acquisition. To achieve this, the key is to develop new ways of representing the POMDP paradigm, such that model uncertainty is taken into account. The approach we propose, called MEDUSA, relies on a Bayesian formulation of uncertainty, which is particularly appropriate to offer a flexible trade-off between a priori knowledge engineering and data-drive parameter inference.

The approach is relatively simple. First, we assume an expert specifies a prior on the model parameters $P(M)$. We then observe data Y from standard trajectories. Assuming we can specify a simple generative process $P(Y|M)$, then it is straight-forward to apply Bayes rule and obtain a posterior model $P(M|Y)$ which combines both the expert knowledge and the data acquired. Since POMDP model parameters are generally represented using multinomial distributions, it is convenient to represent the model prior (and posterior) using Dirichlet distributions, which are the conjugate prior for the multinomial. There is one more obstacle: to apply this method we need to know the *state* corresponding to each data point acquired, since this will tell us which parameter to update. But the state is not usually given in POMDPs. To overcome this, we assume access to an oracle which can identify the state of the system when queried. This is a relatively strong assumption, in the POMDP context. However it is standard in most other planning frameworks. We comment below on ways to relax this assumption.

Given these preliminaries, we formulate an algorithm which uses Dirichlet distributions to capture model uncertainty. The algorithm relies on a sampling of POMDP models from these distributions to plan and select actions. As learning progresses, the set of sampled models will gradually converge to the correct model. Here is the full algorithm:

1. Define Dirichlet priors over the model.
2. Sample a set of POMDPs from the distribution.
3. Solve each model using standard POMDP technique.
4. Use policy to select good actions.
5. Query the oracle and observe answer (s,a,z,s)
6. Increment Dirichlet parameters $Dir(\alpha_{s,a,s})$, $Dir(\alpha_{s,a,z})$
7. Continue until convergence.

Throughout, we maintain a weight indicating the probability of a sampled model. Models with low weights are dropped periodically, and replaced by re-sampling the Dirichlet distributions.

In practice, we also try to minimize the number of queries to the oracle. For example, if the robot happens to be in a part of the state space that has already been well-explored, then it is not useful to query the oracle since no new information will be provided. In such cases, the robot should simple behave optimally until it moves towards less-explored regions. To accomplish this as part of MEDUSA, we considered a number of heuristics designed to decide when the oracle should be queried, and when the robot should instead follow its policy. The method we settled on combines information about the *variance* over the value computed by each model, the *expected information gain* that a query could yield, the *entropy* in the belief, and the *number* of recent queries. This aspect of MEDUSA is somewhat ad-hoc. We have considered methods to formalize it, such as including the decision of whether to query within the POMDP decision-making (Jaulmes et al., 2005a). However solving this optimally proves intractable for all but the smallest problems (e.g. 2 states, 2 actions), therefore we continue to use the heuristics mentioned above.

Experimental validation of the MEDUSA technique has focused on a scenario where the SmartWheeler must navigate in an environment, with the goal of autonomously finding a caregiver that is also mobile. Similar versions of this problem have been studied before in the POMDP literature under various names (Hide, Tag, Find-the-patient). Previous work always assumed a fully modeled version of this problem, where the person's location is unknown, but the person's motion model is precisely modeled, as are the robot's sensor and motion models. We now consider the case where in addition to not knowing the person's position, we are also uncertain about the person's motion model and the robot's sensor model. In total, MEDUSA is trying to learn 52 distinct parameters. We consider the environment shown in Figure 3. Planning and learning are done over a discretized version; the associated POMDP has 362 states, 24 observations and 5 actions. We assume a fixed-resolution grid (future plans include integration of the method describe in the previous section). Execution assumes the continuous state representation and in that case belief tracking is done onboard Carmen using the full particle filter.

During learning, MEDUSA makes several queries about the state. Since there is no model uncertainty about the

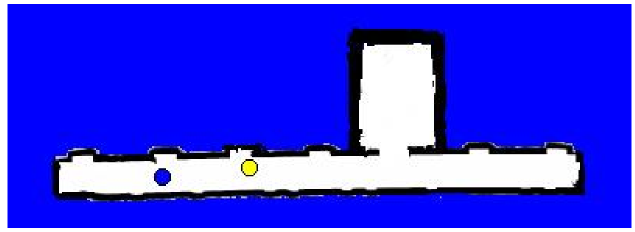
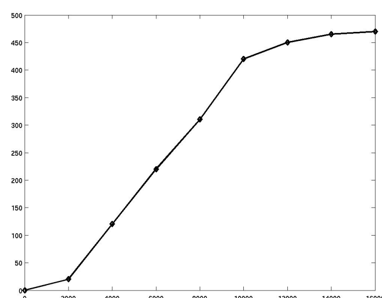


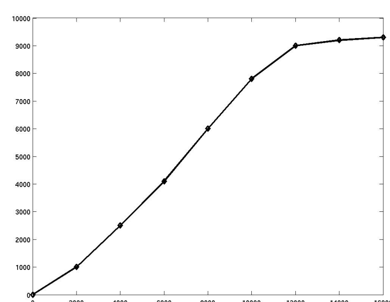
Figure 3: Map of the environment used for the robot simulation experiment.

robot's motion, this is equivalent to asking the caregiver to reveal his/her position so that MEDUSA can infer his/her motion model. The answer to the queries can be provided by a human operator, though for convenience of carrying out multiple evaluations, in our experiments they are produced using a generative model of the caregiver.

As we can see from Figure 4, MEDUSA converges within roughly 12,000 time steps, after having received answers to approximately 9,000 queries. While this may seem large, it is worthwhile pointing out that MEDUSA's oracle can in fact take the form of a high-precision sensor. It is realistic to assume for example that the caregiver will carry around a GPS sensor that can answer queries automatically during the learning phase, and that this will play the role of the oracle. In such a setup, 9,000 queries seems a small price to pay to obtain a full probabilistic model of the person's motion model.



(a) Discounted reward as a function of the number of time steps.



(b) Number of queries as a function of the number of time steps.

Figure 4: Results for the robotic task domain.

The other reason why MEDUSA requires so many queries for this problem is that the experiment assumed completely uninformed initial priors on the robot's sensor model and the caregiver's motion model. Using a more informed prior would lead to faster learning, but would require more knowledge engineering. Finally, to further reduce the number of queries, we could also build a simpler model with fewer Dirichlet parameters, in effect assuming stronger correlations between model parameters.

Further information on this component is available in (Jaulmes et al., 2005a; Jaulmes et al., 2005b; Jaulmes et al., 2007).

Large-scale dialogue management

A natural medium for communication between a user and an intelligent system is through voice commands. While many commercial software solutions are available for speech recognition and synthesis, there is no commercial equivalent for handling the actual dialogue (i.e. production of responses by the robot). In fact, in a spoken dialogue system, determining which action the robot (or computer) should take in a given situation is a difficult problem due to the uncertainty that characterizes human communication.

Earlier work pioneered the idea of POMDP-based dialogue managers, but were limited to small domains (e.g. 2-3 topics in a question-answer format). This prompted an investigation of new techniques for tracking the dialogue state and efficiently selecting dialogue actions in domains with large observation spaces. In particular, we study the applicability of two well-known classes of data summarization techniques to this problem.

We first investigated the use of the K-means algorithm to find a small set of summary observations. The idea is to cluster the natural observations Z into the clusters Z', such that observations with similar emission probabilities over all states are clustered together. It is crucial to use the *normalized* observation probabilities $Pr(z|s,a)/Pr(z)$ (rather than the unnormalized $Pr(z|s,a)$) to ensure that observations that are clustered together provide (near-)equivalent inference information over the set of states.

We also investigated the use of a dimensionality reduction algorithm along the lines of Principal Component Analysis (with a few added constraints) which finds a good low-dimensional representation of the observation probability model. The idea is to use the lower-dimensional projection during planning, which should reduce computation time.

Both methods have been used in the past to summarize high-dimensional data. In general, common wisdom might suggest that Principal Component Analysis yields a higher quality solution, since it projects the entire observation space. K-means has the advantage that it is usually much faster to compute for very large observation sets.

Before implementing either method onboard the robot, we tested both with the SACTI dialogue corpus (Willians and Young, 2004). The results we obtained suggest that the K-means clustering works well, even in this complex dialogue domain. This is encouraging because the clustering algorithm is simple to implement, fast to compute, and generates intuitive compressed representations. We also found that a constrained-based PCA performed on par with K-means on this 450 word domain, however computation was significantly slower. Further technical details and results are provided in (Atrash and Pineau, 2005). We are now extending the results to topics relevant to the robot domain.

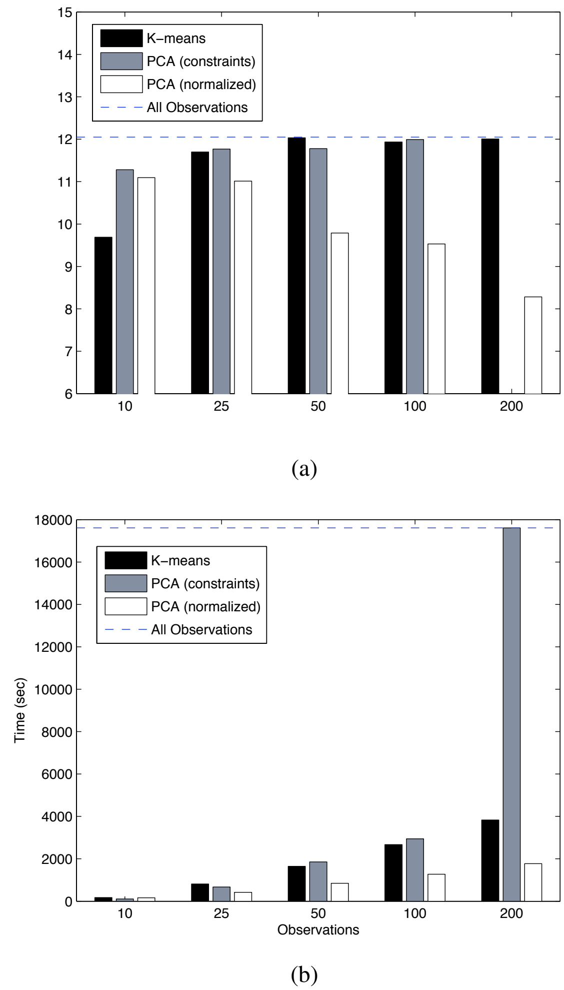


Figure 5: Results for Dialogue POMDP: (a) Expected Reward and (b) Planning time

High-level goal specification

The last component aims at designing and validating a new communication protocol for allowing users to provide high-level navigation commands to the wheelchair. Conventional control of a motorized wheelchair is typically done through a joystick device. For those unable to operate a standard joystick, alternatives include sip-and-puff devices, pressure sensors, etc. Regardless of the device used, the user input set is restricted to displacement and velocity commands. Operation of a wheelchair in this manner can result in fatigue over time, as well it is often difficult to manoeuver the wheelchair in constrained spaces (e.g. elevators, crowded rooms, etc).

The prototype robotic wheelchair we are developing seeks to alleviate these challenges by allowing the users to specify high-level navigation goals (e.g. *Go to room 103.*) This requires a new communication protocol which will allow the user to input such commands.

The input protocol we proposed initially for specifying high-level navigation goals was based on EdgeWrite (Wobbrock and Myers, 2006), a unistroke text entry method for handheld devices, designed to provide high accuracy text entry for people with motor impairments. We have adapted this method for the control of a motorized wheelchair by customizing the set of strokes, input constraints, and feedback display, to the task of wheelchair control.

User experiments currently under way are comparing entry of robot navigation goals using: direct map selection, menu selection, and EdgeWrite gesture entry. For each input modality, the user is shown a floor map of a building on the screen, and guided through a list of locations that must be selected quickly and accurately using the different input selection methods. We measure error rate, input time, and motion time needed to reach the target location. Early results with a control population indicate that menu selection (from a static vertical list) was twice as fast as selecting the targets directly on the map and three times as fast as entering the corresponding EdgeWrite symbol, which is as expected for this population. We must now replicate the experiment with disabled users. Since this population has significant motor constraints, we may obtain significantly different results regarding the preferred mode of input.

Discussion

This paper highlights some of the key components of the SmartWheeler project. We are currently working on their integration, and planning out a sequence of experiments with the target population. Through close collaborations with engineers and rehabilitation researchers, we hope to one day have a positive impact on the quality of life for individuals with severe mobility impairments.

It is worth noting that many of the techniques developed in this project are not specific to the mobility-impaired population, but are relevant to building service robots for a large number of applications. The SmartWheeler platform is proving to be an exciting new test-bed for exploring novel concepts and approaches in automated decision-making, human-robot interaction, and assistive robotics.

Acknowledgments

The SmartWheeler team acknowledges the financial support of the Natural Sciences and Engineering Research Council of Canada and the Canada Foundation for Innovation. We also thank Sunrise Medical Canada Inc. for their generous equipment donation.

References

Atrash, A. and Pineau, J. "Efficient Planning and Tracking in POMDPs with Large Observation Spaces". AAAI-06 Workshop on Empirical and Statistical Approaches for Spoken Dialogue Systems. 2006.

Carmen-Team. Robot Navigation Toolkit. http://carmen.sourceforge.net. Retrieved 2006.

Fehr, L., Langbein, E. and Skaar, S. B. "Adequacy of power wheelchair control interfaces for persons with severe disabilities: A clinical survey". Journal of Rehabilitation Research and Development, 37(3). pp. 353-360. 2000.

Hauskrecht, M. "Value-Function Approximations for Partially Observable Markov Decision Processes". Journal of Artificial Intelligence Research. 13. pp.33-94. 2000.

Jaulmes, R., Pineau, J. and Precup, D. "Active Learning in Partially Observable Markov Decision Processes". European Conference on Machine Learning (ECML). LNAI 3729. Springer-Verlag. 601-608. 2005.

Jaulmes, R., Pineau, J. and Precup, D.. "Probabilistic Robot Planning Under Model Uncertainty: An Active Learning Approach". NIPS Workshop on Machine Learning Based Robotics in Unstructured Environments. 2005.

Jaulmes, R., Pineau, J. and Precup, D. "A formal framework for robot learning and control under model uncertainty". IEEE International Conference on Robotics and Automation (ICRA). 2007.

Kaelbling, L. P., Littman, M. L. and Cassandra, A. R. "Planning and Acting in Partially Observable Stochastic Domains". Artificial Intelligence. 101. pp.99-134. 1998.

Montemerlo, M., Roy N. and Thrun, S. "Perspectives on Standardization in Mobile Robot Programming: The Carnegie Mellon Navigation (CARMEN) Toolkit". IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). 2003.

Pineau, J., Gordon G. and Thrun, S.. "Anytime Point-Based Approximations for Large POMDPs". Journal of Artificial Intelligence Research (JAIR). 27. pp. 335-380.

Pineau, J. and Gordon, G. "POMDP Planning for Robust Robot Control". International Symposium on Robotics Research (ISRR). 2005.

Williams, J.D. and Young, S. "Characterizing task-oriented dialog using a simulated ASR channel. International Conference on Speech and Language Processing (ICSLP). 2004.

Wobbrock, J.O. and Myers, B.A. "Trackball text entry for people with motor impairments". ACM Conference on Human Factors in Computing Systems (CHI). 2006.