

# A Risk Assessment Infrastructure for Powered Wheelchair Motion Commands without Full Sensor Coverage

Pouria TalebiFard, Junaed Sattar and Ian M. Mitchell

**Abstract**—Smart powered wheelchairs offer the possibility of enhanced mobility to a large and growing population—most notably older adults—and a key feature of such a chair is collision avoidance. Sensors are required to detect nearby obstacles; however, complete sensor coverage of the immediate neighbourhood is challenging for reasons including financial, computational, aesthetic, user identity and sensor reliability. It is also desirable to predict the future motion of the wheelchair based on potential input signals; however, direct modeling and control of commercial wheelchairs is not possible because of proprietary internals and interfaces. In this paper we design a dynamic egocentric occupancy map which maintains information about local obstacles even when they are outside the field of view of the sensor system, and we construct a neural network model of the mapping between joystick inputs and wheelchair motion. Using this map and model infrastructure, we can evaluate a variety of risk assessment metrics for collaborative control of a smart wheelchair. One such metric is demonstrated on a wheelchair with a single RGB-D camera in two scenarios: a doorway traversal where the near edge of the doorframe is no longer visible to the camera as the chair makes its turn, and a longer navigation through a typical cluttered office environment.

## I. INTRODUCTION

Mobility impairment is becoming increasingly common as the population ages, and a lack of mobility can lead to a host of further social, mental and physical impairments. Powered wheelchairs (PWCs) can greatly improve the mobility of those who are unable to self-propel in manual wheelchairs; however, existing PWC technology is often inappropriate for users with, for example, low vision, hemispatial neglect, spasticity, tremors, dementia, severe Alzheimers, or other cognitive impairment because they may not be able to safely drive these large, heavy and powerful machines [1]. Smart PWCs seek to overcome this limitation and could benefit a sizeable population [2]. Considerable progress has been made in fully autonomous PWC navigation, but there are populations for which full autonomy is undesirable; for example, older adults with cognitive impairments can become quite agitated when the PWC seems to act on its own initiative [3]. Consequently, recent efforts have focused on smart PWCs which can constantly yet seamlessly collaborate

with the user to accomplish his or her goals. In the context of this paper, the goal of the shared control is minimal intervention in the user's commanded motion while still avoiding collisions.

Because the majority of PWCs are purchased through insurance or public health care plans with significant pressures to reduce expenses, a critical factor in the success of any commercial intelligent PWC will be the cost of sensors. With manufacturer and retailer markups totalling several hundred percent on parts, few payers will be willing to cover the cost of surrounding a power wheelchair with a suite of laser rangefinders (LRFs). For cost purposes, currently available small and cheap RGB-D cameras (such as the Kinect™) are an appealing alternative to LRFs. However, cameras have much narrower fields of view (less than 60° for the Kinect) than LRFs (typically 120°–180°), and camera systems generate much more data than 2D LRFs; for example, a single Kinect nearly saturates a typical laptop's peripheral bus. Even if it were financially and/or computationally practical to completely surround a PWC with LRFs and/or cameras, it is likely that the resulting system's halo of sensors would be unappealing to potential users, both for aesthetic reasons [4] and due to the potential for stigmatization [5].

Consequently, full sensor coverage in a smart PWC system is an essentially unattainable goal, and gaps in coverage can lead to collisions. As an example, a system with a front-facing stereo vision camera was tested with cognitively impaired older adults in [3]. The camera's narrow field of view led to a common failure mode during the trials: The participant would drive the PWC parallel to a wall (for example, down a corridor) until it was close enough to a corner or doorway that the camera could no longer see the adjacent wall; if the participant then made too tight a turn the footrest or side of the PWC would hit the corner or doorframe (see figure 2 for an illustration of this moment).

In this paper we describe a system which can construct a collision risk assessment for short-term trajectories of the PWC which may steer it into regions not currently visible to the sensor system. We adapt a probabilistic mapping technique [6] to construct an egocentric local occupancy map which surrounds the PWC and thereby provides information about (static) obstacles outside the sensor field of view.

In addition to the local obstacle map, a key requirement for risk assessment is accurate modeling of future trajectories. Owing to the proprietary design of most PWCs, it is often infeasible to interact directly with the underlying control system. To work around this constraint we construct a neural net model mapping joystick motion to PWC motion.

This research was supported by CANWHEEL (the Canadian Institutes of Health Research (CIHR) Emerging Team in Wheeled Mobility for Older Adults #AMG-100925), National Science and Engineering Council of Canada (NSERC) Discovery Grant #298211, the Canadian Foundation for Innovation (CFI) Leaders Opportunity Fund / British Columbia Knowledge Development Fund Grant #13113, and Fonds de Recherche du Québec – Nature et Technologie.

The authors are at the Department of Computer Science, University of British Columbia, Vancouver, Canada. Email: {ptfard2, junaed, mitchell}@cs.ubc.ca

Given a map and a method to simulate the trajectory that will be generated by a particular choice of input signal, there are many approaches to assess risk. In a fully autonomous system, this process is relatively straightforward: choose an input signal which has a low risk of collision. In contrast, in a shared control system the user’s (relatively unpredictable) input must be taken into account; therefore, the method for assessing risk must consider both the user’s capabilities and the ways in which the system is willing to intervene if the risk becomes too great. For the purposes of the experiments in this paper we propose a straw man risk assessment to demonstrate the technique. We expect that the choice of risk assessment would depend on operator diagnosis, and therefore we focus our efforts here on mapping and modeling algorithms on top of which a variety of risk assessments can easily be constructed.

With that goal in mind, the contribution of this paper is a system through which a measure of the short-term risk of collision with nearby objects can be evaluated using only a single, low-cost, discreetly mounted RGB-D camera and without detailed knowledge of or access to the PWC control system. This task is accomplished by maintaining an efficient egocentric probabilistic occupancy map (to overcome the camera’s narrow field of view) and constructing a neural network model of PWC motion (to simulate the response of the PWC to the user’s chosen inputs). This map and model infrastructure can easily be modified to handle additional sensors and/or different PWCs.

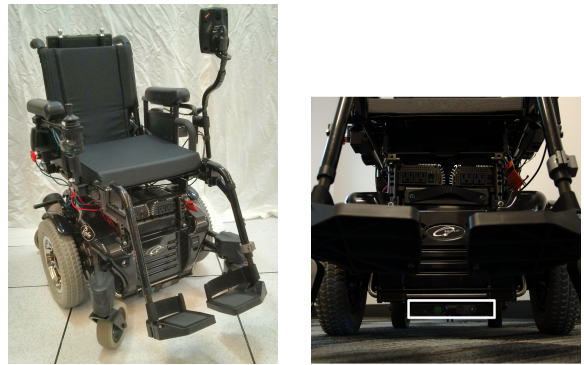
### A. Related Work on Smart Wheelchairs

Research on smart PWCs has a long and broad history, so we focus here on relatively recent work. In particular, a number of prototypes that attempt to provide safe locomotion and navigation to users with cognitive impairments have been described, and they use a range of sensor systems including sonars [7], [8], infrared sensors [8], LRFs [8]–[12], stereo vision cameras [3], and omnidirectional stereo vision cameras [13]. In all of these systems the mounting location for the sensor(s) is chosen to maximize effectiveness, rather than unobtrusive or aesthetically pleasing locations. Broad consumer and payer acceptance of smart PWC products will eventually demand a minimum of sensors placed in less than ideal locations, and our focus here is on mechanisms to evaluate risk of collision despite the resulting limited sensor coverage.

## II. HARDWARE AND SOFTWARE PLATFORM

The system described below is implemented in ROS [14] (Fuerte Turtle release) on top of an Ubuntu version 12.04LTS host OS. The custom code we developed<sup>1</sup> is entirely written in Python. The runtime hardware is a Lenovo W530 laptop with Intel® Core™ i7-3720QM processor at 2.6 GHz, 8GB main memory and a 120GB solid state drive. During execution (without visualization) our system uses approximately 0.1GB memory and 30% of the CPU while grabbing camera data at roughly 20 frames per second.

<sup>1</sup>Available at <http://code.google.com/p/ubc-canwheel/>



(a) PWC with the sensor mounted underneath the base.

(b) Close-up view of the RGB-D sensor highlighted with a white rectangle.

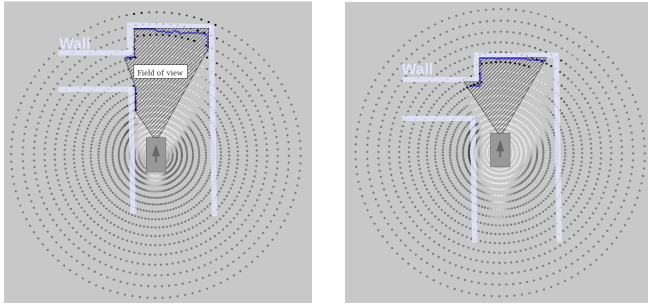
Fig. 1: Photos of our PWC.

Our PWC is a Quickie® Rhythm modified by AT Sciences to include their Drive Safe collision avoidance system [15] and custom rotary encoders on the electric motor driveshafts that we use for odometry measurement. The only component of the Drive Safe system that we use in the research described here is the CANBus to USB interface, which allows us to read the odometers and the joystick and write joystick-like commands to the PWC’s motor controller (we have removed the non-embedded components of the Drive Safe software, as well as the infrared, sonar and bump sensor pods which surrounded the PWC). Our sensor is an Asus Xtion Pro Live RGB-D camera connected to the laptop through a USB 2.0 bus. The camera is mounted almost invisibly below the battery enclosure, looking forward from under the footrests. Figure 1 shows both the PWC and a close-up identifying the location of the camera.

## III. BUILDING AN EGOCENTRIC LOCAL MAP

Robust mapping techniques for obstacle representation and robot navigation in unstructured and unknown environments have taken different forms over the years. From Vector Field Histograms [16] to Dynamic Windows [17] to occupancy grid maps (for example, see [6]), each adopt a distinct obstacle representation that is well suited to their more primary tasks—navigation and obstacle avoidance in most cases. Unlike autonomous navigation, however, our goal is not to find the best trajectory, to find the set of feasible trajectories, or even to reject a subset of trajectories as infeasible; rather we seek a way to measure some risk function for any possible trajectory of the PWC.

Because our goal is to evaluate the risk associated with short-term trajectories, we construct an egocentric occupancy map: the origin of the map coordinate system is always centered on the camera. We furthermore represent the map in a polar coordinate system with angle  $\phi \in [-\pi, +\pi]$ , where  $\phi = 0$  is directly in front of the PWC, and radial distance  $\rho \geq 0$ . The map is stored on a curvilinear grid generated by a tensor product of samples in  $\phi$  and  $\rho$  (the white, grey and black dots in figure 2). The  $\phi$  samples are equally spaced over the interval  $[-\pi, +\pi]$ . The  $\rho$  samples are drawn from an interval  $[\rho_{\min}, \rho_{\max}]$  where  $\rho_{\min}$  is chosen



(a) Polar map representation before reaching the corner and while the adjacent wall is still in the perceptual field. (b) Polar map representation where the adjacent wall is not in the field of view.

Fig. 2: Screenshot of the polar map with actual boundaries (walls) and field of view superimposed on top. White dots correspond to empty spaces, grey dots correspond to unknown cells and darker dots correspond to occupied cells. Dark blue dots are the raw range readings.

such that all points  $\rho < \rho_{\min}$  would be within the PWC and  $\rho_{\max}$  is chosen based on the maximum range of the RGB-D camera. Furthermore, the spacing of samples in the  $\rho$  direction grows quadratically, to match both the degradation of accuracy of the camera’s depth readings as the distance to obstacles grows as well as the decreased importance of distant obstacles in the short-term risk assessment process.

The egocentric map requires that the entire map is updated whenever the chair moves, but there is no need to maintain a localization estimate within the map. While updating the map is as much as or slightly more expensive than updating a typical localization estimate, a significant computational benefit accrues during the simulation of future trajectories for risk assessment: There is no need to deal with uncertainty in the initial position from which these trajectories start. We choose the polar coordinate system because updates from the camera (or any range sensor at the origin) are cheap to apply to the map—a distance reading in a particular direction updates all  $\rho$  values for the corresponding  $\phi$ —and movement updates are no more expensive than they would be in a rectangular coordinate system.

The algorithm for maintaining the map follows a fairly standard approach [6] with minor modifications to account for limited sensor coverage and the fact that the egocentric map can change due to either obstacle or PWC movement. At each node in the map  $(\rho, \phi)$  we maintain a log odds ratio

$$l_t(\rho, \phi) = \frac{p(m_t(\rho, \phi) | m_{t-1}, z_t, u_t)}{1 - p(m_t(\rho, \phi) | m_{t-1}, z_t, u_t)}$$

where  $p(m_t(\rho, \phi) | m_{t-1}, z_t, u_t)$  is the probability that the location  $(\rho, \phi)$  is occupied at time  $t$  given the previous map  $m_{t-1}$ , the most recent sensor reading  $z_t$  and the most recent movement  $u_t$ . A binary Bayes filter is used to update the log odds ratio every time a new sensor reading arrives or movement is performed. Full details can be found in [18].

Figure 2 shows a visualization of the mapping results for a scenario with limited sensor coverage while navigating a

corner. The dots in concentric circles are the map nodes; white corresponds to nodes with low log-odds of being obstacles, black to high log-odds, and grey to intermediate log-odds. The blue dots are raw sensor readings from a single camera frame. The wheelchair (grey rectangle), camera field of view (crosshatched cone) and corridor walls (blue grey lines) are overlaid for the convenience of the reader; only the map nodes and sensor readings are known to the algorithm. Not surprisingly the map shows appropriate colors for nodes lying within the sensor’s field of view; however, its key feature is that in the right subfigure it likewise correctly shows nodes which are black under the left corridor wall and white just inside the doorway, even though these nodes are not within the camera’s current field of view (nodes further through the doorway remain grey because they were never viewed by the camera).

#### IV. RISK ASSESSMENT

Because the appropriate measure of and response to risk in a collaboratively controlled PWC may be diagnosis and even user specific, our goal in this paper is not to propose a particular approach to risk assessment, but rather to construct components from which a variety of assessments can be built.

The fundamental tool for constructing a risk assessment in this framework is the simulation of the PWC’s future pose (*i.e.*  $x_{t+1}$ ) through the local map  $m_t$ . To that effect, a neural network is trained using empirical data collected from approximately 60 minutes of driving, which uses (i) current joystick position  $v_t$ , (ii) change in joystick position since last sample  $\delta v_t = v_t - v_{t-1}$  (to account for control system delay), (iii) wheel rotation since last sample  $u_t$  (a proxy for PWC velocity), and (iv) change in wheel rotation since last sample  $\delta u_t = u_t - u_{t-1}$  (a proxy for PWC acceleration), to estimate (left and right) wheel rotations at the next sample time  $u_{t+1} = f(v_t, \delta v_t, u_t, \delta u_t)$ . For convenience, we will assume that both functions are discrete time with a constant sampling period (e.g.: 200ms – details can be found in [18]). The PWC has nontrivial size, so at each sample of the simulation it will overlap with some subset of the map’s grid cells, each of which has some probability of containing an obstacle. The design of a risk assessment therefore has a number of parameters to determine, including: (i) simulation horizon, (ii) number of times to evaluate risk over that horizon (could be as often as every 200ms, although the PWC will not typically move very much over such a short period), (iii) how to account for the shape of the PWC, (iv) number of future joystick signals to consider, and (v) what future joystick signals to consider.

Depending on available interventions and how the risk assessment will be used in these interventions, the form of the risk assessment might vary from a separate assessment for each possible input, a time dependent assessment, or perhaps just a single overall value. It is fairly clear that the reported risk should not decrease if the PWC overlaps with a grid cell which is more likely to be occupied, if more of the PWC overlaps with obstacles, if collision is reported for more input signals and/or if a collision will occur

sooner; however, these monotonicity constraints still leave considerable flexibility in choosing aggregation functions when more compact risk assessments are desired.

As an example of a risk assessment which results in a single number at a given point in time, consider the greatest log-odds of collision:

$$\text{risk} = \max_{v_0:t_{\max}-1 \in V} \max_{0 \leq t \leq t_{\max}} \max_{\hat{x} \in \mathfrak{W}(x_t)} l_t(\hat{x}) \quad (1)$$

where  $V$  is a set of input signals,  $t_{\max}$  is the time horizon,  $\mathfrak{W}(x_t)$  is the set of all points inside the PWC when it is at point  $x_t$ , and  $l_t(\hat{x})$  is the log-odds of point  $\hat{x}$  (or its nearest neighbour in the grid) being occupied by an obstacle according to the map at time  $t$  (as constructed in section III). Typically  $\mathfrak{W}(x_t)$  would be approximated by a sampling of the PWC’s shape. In the next section we demonstrate our system using a version of this risk assessment, but there are many more possibilities. A Bayes’ risk model [19], for example, has been used for robotic task execution under uncertainty. Domain-specific risk evaluation measures have also been applied to ensure task safety in human-robot dialog [20]. Another probabilistic model for safety assessment in dynamic environments under an imperfect sensing model was proposed in [21].

## V. EXPERIMENT

We test the effectiveness of the system described above in two scenarios. In the first, the system attempts to detect higher risk maneuvers when the user turns into a doorway from a corridor, a scenario meant to replicate a common source of collision observed in the trials described in [3] (as discussed in section I). In the second, we demonstrate the effectiveness of the system as the user drives an extended trajectory through a cluttered environment while maneuvering the PWC in all directions. Note that the system is passively evaluating the risk metric in all of these experiments; the user (the first author) is always in full control of the motion.

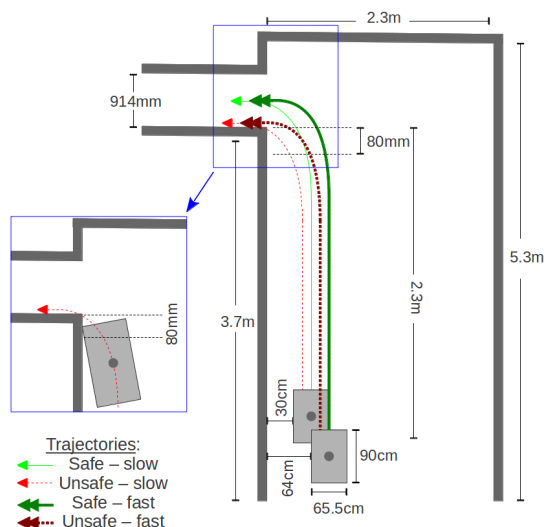


Fig. 3: Experimental setup and the four tested trajectories.

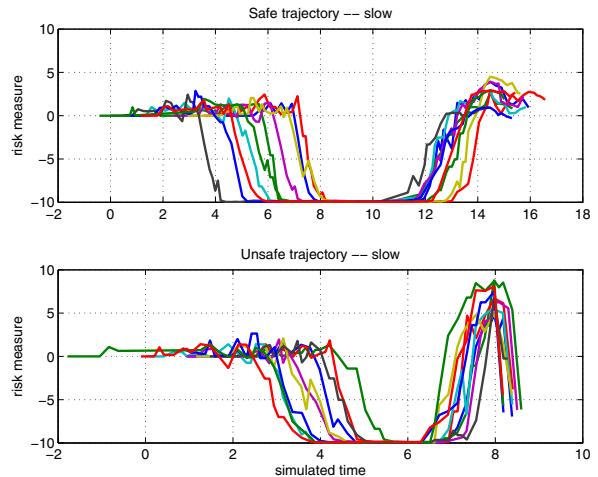


Fig. 4: Safe (top) and unsafe (bottom) trajectories in slow-speed mode.

For the first scenario, the experimental setup uses a 914mm (36in) wide simulated doorway (a common width in North American public buildings). It should be noted that our PWC is 655mm wide and that health care facilities typically mandate wider doors—for example, 1168mm (46in) in the province of Ontario [22]—so our doorway poses a moderate degree of collision risk simply because of the relatively tight fit.

As shown in Figure 3, we perform four experiments: (i) a safe slow turn, (ii) an unsafe (early) slow turn, (iii) a safe fast turn, and (iv) an unsafe (early) fast turn. In every experiment the PWC starts from rest approximately 2.3m down the corridor, and the user attempts to drive it parallel to the wall until the turn is initiated. For the “slow” trajectories, the speed of the PWC is capped at 50% of the maximum speed ( $\sim 0.4\text{m/s}$ ) and the initial portion of the trajectory is  $\sim 30\text{cm}$  from the corridor wall. For the “fast” trajectories the speed is 70–90% of maximum ( $\sim 0.7\text{m/s}$ ) and the distance from the corridor wall is  $\sim 64\text{cm}$ . A turning point was identified on the floor for each of the “unsafe” trajectories such that the PWC would impact about 7.5cm before the corner, although the PWC was stopped just before impact. For the “safe” trajectories the turn was initiated approximately 15cm further along the corridor, so the PWC passed no closer than about 7.5cm from the corner, and was stopped after it had passed through the doorway. The right side of figure 2 shows a typical map status just before turning begins; the corner is not within the field of view of the camera and remains outside of it throughout the turn.

We report the greatest log-odds of collision risk metric (1). The horizon was chosen as  $t_{\max} = 1.4\text{s}$ . The set of future input signals  $V$  considered included only one: that the joystick was held constant at its current position. The PWC shape  $\mathfrak{W}(x(t))$  is represented by 17 samples (5 along each of the long edges, 3 along the short edges and 1 in the center of a rectangular bounding box). The risk is recomputed every time a new joystick input arrives.

Each of the four experiments was repeated 10 times. Figures 4 and 5 show the time evolution of the risk metric (1)



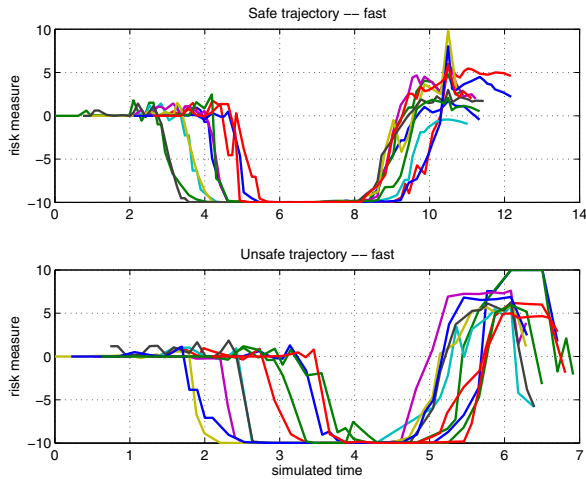


Fig. 5: Safe (top) and unsafe (bottom) trajectories in fast-speed mode.

during each run of each experiment (time shifted so that the peak of every run for a single experiment is aligned). The scaling of the vertical axis is essentially arbitrary and the values of the metric have been capped to  $\pm 10$ , so we are primarily interested in comparing the value of the metric between experiments rather than drawing conclusions from its specific value.

All of the experimental runs follow the same basic pattern. When the system is first turned on at the start of each run, the risk metric hovers around zero until the sensors have time to determine that there is free space in front of the chair; we will consider this the “baseline” risk of driving through a region whose obstacles are unknown. At that point the metric drops significantly as the chair moves safely along the corridor parallel to the wall. When the chair begins its turn the metric rises quickly toward a peak, and then falls again when the chair is stopped at the end of the run.

The distinction between the four experiments lies in the height of the peak as the chair makes its turn. Consider first the “slow” experiments shown in figure 4. In the “slow unsafe” case on the bottom, all runs show the risk metric rising significantly above the baseline risk detected at the beginning of the run, indicating that a collision is a significant threat. In contrast, for the “slow safe” case on the top all runs show a distinctly lower peak. The fact that the chair is making a blind turn into a narrow doorway explains why these peaks are still at or slightly above the baseline risk at the beginning of the run: toward the end of the run, the chair is again predicting motion through unknown portions of the map. Now consider the “fast” experiments shown in figure 5. In the “fast unsafe” case, all runs show high peaks similar to those in the “slow unsafe” case, again indicating that the threat of collision is significant. In the “fast safe” case the range of peaks displayed by different runs is much wider—everything from baseline to the top of the scale—however, we are not terribly concerned that the risk may be overestimated in this case because most PWC users would consider a  $90^\circ$  turn at nearly full speed around a blind corner

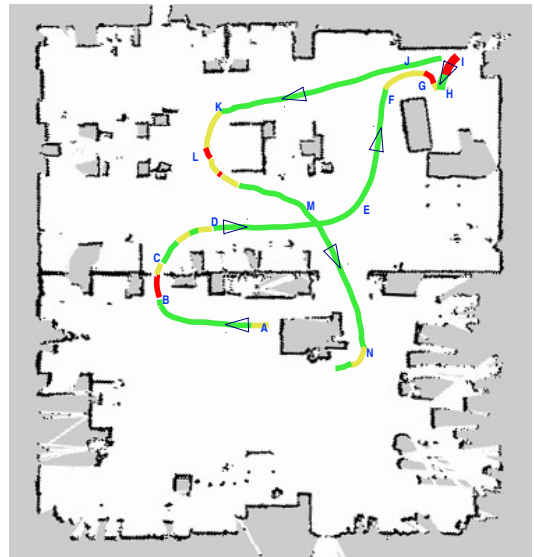


Fig. 6: Occupancy-grid map of the environment navigated during the long-run trial.

to be a somewhat dubious maneuver even if they were certain of avoiding the corner itself.

In the second scenario the PWC is driven in a typical office environment, as depicted by the occupancy-grid map in Figure 6 (generated offline), along the path shown in colors and with arrows indicating wheelchair heading. The green lines indicate areas evaluated as not risky, yellow indicates areas of moderate risk, and red indicates areas of high risk along the path. The time-series plot of risk assessment along the path is shown in Figure 7. We have added corresponding annotations (alphabetically ordered in time along the trajectory) to both figures to denote specific areas of interest. As can be seen, while travelling close to obstacles (such as at G and L) or through narrow doorways (such as between B and C), risk assessment is high. Of particular interest is the section of the map where the wheelchair performs a “backing up” maneuver between H and I with the wall directly behind it. This wall was detected by the camera and incorporated into the egocentric map while the chair was facing toward it between F and G; however, it is behind the chair and hence invisible to the camera while the chair is backing up between H and I. In spite of the obstacle being in the sensor’s blindspot, the risk assessment goes from low to high as the chair approaches the wall while backing up after H, and then drops again when the chair stops backing at I in preparation for forward motion toward J. This reaction of the risk metric demonstrates the utility of our egocentric map in detecting potential collisions with previously observed obstacles now lying outside the camera’s narrow field of view.

## VI. CONCLUSION

We have described a method of constructing a dynamic egocentric occupancy grid to efficiently represent local obstacles both inside and outside the current sensor coverage, and a neural network model of PWC motion in response

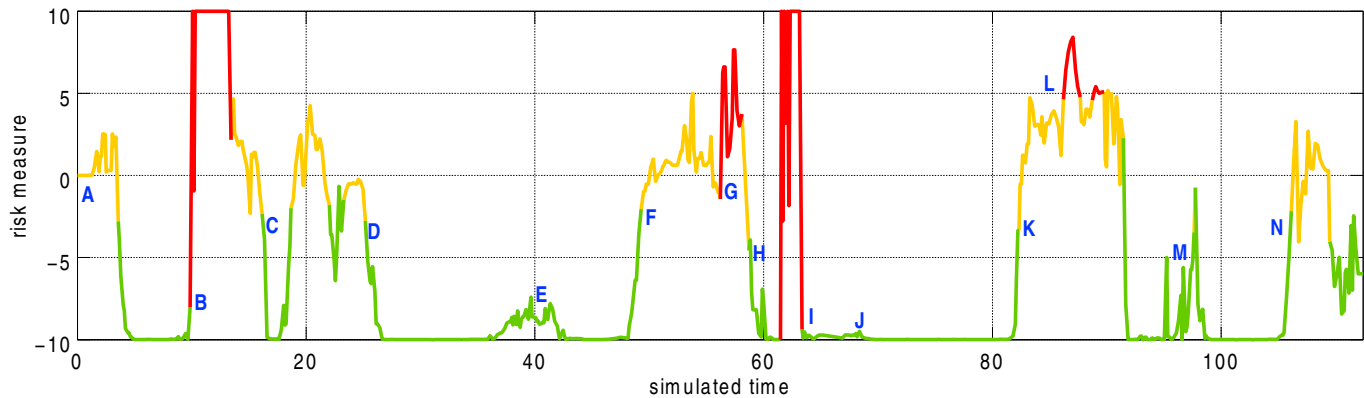


Fig. 7: Time-series plot of risk assessment during the long-run trials shown in Figure 6.

to joystick inputs. This infrastructure can be combined to produce a variety of risk assessment metrics even when it is not possible to sense all the regions into which the PWC might be steered. We recognize that deployable smart PWCs will need more than the single camera and discontinued PWC product used in this prototype, but modification of the algorithms to multiple sensors and/or other PWCs is straightforward. On the flip side, these algorithms allow for accurate risk assessment despite inevitable sensor blind spots and black-box PWC controllers.

The specific risk assessment metric used in the experiments reported here is merely a straw man, and we hope that current Wizard-of-Oz trials with our target population will yield insight into user-appropriate choices of risk assessment metric(s), which we can then implement using the infrastructure developed in this paper.

#### ACKNOWLEDGMENT

The authors would like to thank Pooja Viswanathan for extensive discussion about the trials described in [3], Nando de Freitas for help with the NN model, and the whole CAN-WHEEL team for their clinical and technological feedback.

#### REFERENCES

- [1] L. Fehr, W. E. Langbein, and S. B. Skaar, "Adequacy of power wheelchair control interfaces for persons with severe disabilities: A clinical survey," *Development*, vol. 37, no. 3, pp. 353–360, 2000.
- [2] R. C. Simpson, E. F. LoPresti, and R. A. Cooper, "How many people would benefit from a smart wheelchair?" *Journal of Rehabilitation Research and Development*, vol. 45, no. 1, pp. 53–71, 2008.
- [3] P. Viswanathan, "Navigation and obstacle avoidance help (NOAH) for elderly wheelchair users with cognitive impairment in long-term care," Ph.D. dissertation, University of British Columbia, 2012.
- [4] G. Pullin, *Design Meets Disability*. MIT Press, 2009.
- [5] J. Miller Polgar, "The myth of neutral technology," in *Design and Use of Assistive Technology*, M. M. K. Oishi, I. M. Mitchell, and H. F. M. Van der Loos, Eds. Springer, 2010, pp. 17–23.
- [6] S. Thrun, W. Burgard, and D. Fox, *Probabilistic robotics*. MIT Press, 2005.
- [7] G. Bourhis, O. Horn, O. Habert, and A. Pruski, "An autonomous vehicle for people with motor disabilities," *IEEE Robotics & Automation Magazine*, vol. 8, no. 1, pp. 20–28, 2001.
- [8] E. Prassler, J. Scholz, and P. Fiorini, "A robotics wheelchair for crowded public environment," *IEEE Robotics & Automation Magazine*, vol. 8, no. 1, pp. 38–45, 2001.

- [9] E. B. Vander Poorten, E. Demeester, E. Reekmans, J. Philips, A. Huntemann, and J. De Schutter, "Powered wheelchair navigation assistance through kinematically correct environmental haptic feedback," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2012, pp. 3706–3712.
- [10] Y. Wang and W. Chen, "Hybrid map-based navigation for intelligent wheelchair," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2011, pp. 637–642.
- [11] G. Peinado, C. Urdiales, J. Peula, M. Fdez-Carmona, R. Annicchiarico, F. Sandoval, and C. Caltagirone, "Navigation skills based profiling for collaborative wheelchair control," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2011, pp. 2229–2234.
- [12] T. Carlson and Y. Demiris, "Increasing robotic wheelchair safety with collaborative control: Evidence from secondary task experiments," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2010, pp. 5582–5587.
- [13] Y. Satoh and K. Sakaue, "An omnidirectional stereo vision-based smart wheelchair," *EURASIP Journal on Image and Video Processing*, no. 087646, 2007.
- [14] M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, and A. Ng, "ROS: an open-source robot operating system," in *IEEE Intl. Conf. on Robotics and Automation (ICRA) Workshop on Open Source Robotics*, Kobe, Japan, May 2009.
- [15] R. Simpson, E. LoPresti, S. Hayashi, I. Nourbakhsh, D. Miller, *et al.*, "The smart wheelchair component system," *Journal of Rehabilitation Research and Development*, vol. 41, no. 3B, pp. 429–442, 2004.
- [16] J. Borenstein and Y. Koren, "The vector field histogram-fast obstacle avoidance for mobile robots," *Robotics and Automation, IEEE Transactions on*, vol. 7, no. 3, pp. 278–288, 1991.
- [17] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *Robotics & Automation Magazine, IEEE*, vol. 4, no. 1, pp. 23–33, 1997.
- [18] P. TalebiFard, "A risk assessment infrastructure for powered wheelchair motion commands without full sensor coverage," Master's thesis, University of British Columbia, 2014.
- [19] F. Doshi, J. Pineau, and N. Roy, "Reinforcement learning with limited reinforcement: Using bayes risk for active learning in POMDPs," in *Proceedings of the 25th International Conference on Machine Learning*. ACM New York, NY, USA, 2008, pp. 256–263.
- [20] J. Sattar and J. J. Little, "Ensuring safety in human-robot dialog – a cost-directed approach," in *Proceedings of the IEEE International Conference on Robotics and Automation, ICRA*, Hong Kong, China, May/June 2014.
- [21] D. Althoff, J. Kuffner, D. Wollherr, and M. Buss, "Safety assessment of robot trajectories for navigation in uncertain and dynamic environments," *Autonomous Robots*, vol. 32, no. 3, pp. 285–302, 2012.
- [22] Ontario Ministry of Health and Long-Term Care, "Long-term care facility design manual," 1999.