Robotic ADaptation to Humans Adapting to Robots: Overview of the FP7 project RADHAR

Eric Demeester,
Emmanuel Vander Poorten,
Alexander Hüntemann
and Joris De Schutter
Mechanical Engineering
KU Leuven
Heverlee, Belgium
first.last@mech.kuleuven.be

Michael Hofmann,
and Martijn Rooker
Profactor
Steyr-Gleink, Austria
first.last@profactor.at

Gernot Kronreif
ACMIT
Wiener Neustadt, Austria
Gernot.Kronreif@acmit.at

Boris Lau, Markus Kuderer
and Wolfram Burgard
Computer Science
University of Freiburg
Freiburg, Germany
{lau,kudererm,burgard}@
informatik.uni-freiburg.de

Anne Gelin and
Katrien Vanopdenbosch
School Windekind
Leuven, Belgium
ict@windekindleuven.be

Pascal Van der Beeten,
Monia Vereecken
and Stephan Ilsbroukx
National MS Centre
Melsbroek, Belgium
first.last@ms-centrum.be

Andrea Fossati, Gemma Roig,
Xavier Boix, Marko Ristin
and Luc Van Gool
Computer Vision, ETH
Zürich, Switzerland
{fossati,gemmar,boxavier,
ristin,vangool}@
vision.ee.ethz.ch

Hans Fraeyman*,
Lieven Broucke*,
Hendrik Goessaert*
and John Josten‡
Permobil NV* & AB‡
*Aalter, Belgium
‡Kerkrade, The Netherlands
H.Fraeyman@permobil.be
J.Josten@permobil.nl

***Abstract*—This paper presents the research objectives and current state of the FP7 project RADHAR (www.radhar.eu). RADHAR proposes a framework to fuse the inherently uncertain information from both environment perception and a wheelchair driver's steering signals by estimating the trajectory the wheelchair should execute, and to adopt this fused information for providing safe navigation assistance. Furthermore, the wheelchair's level of autonomy is continuously adjusted to the driver's varying capabilities and desires. For each of the key components in the RADHAR framework, experimental results are shown.**

I. Introduction

Autopilots in airplanes greatly reduce the pilot's workload by taking over parts of the navigation. Their success in reducing navigational complexity and improving safety motivates the introduction of navigational assistance in other transportation means as well. However, implementing robotic navigation correction on a large scale also represents a potential safety risk for millions of users.

For this reason, if navigation devices are to be correcting the driver's steering signals, thorough understanding of driver behaviour and pervasive environment perception and interpretation are imperative. Though driver models have been proposed for vehicles where the driver is in full control, such driver models for intelligent vehicles are scarce. Furthermore, the linking between environment perception, driver perception and modelling, and robot decision making has often been weak and ad hoc. The RADHAR project proposes a framework to seamlessly fuse the inherently uncertain information from both environment perception and the driver's steering signals by estimating the trajectory the robot should execute, and to adopt this fused information for safe navigation with a level of autonomy adjusted to the user's capabilities and desires. This requires lifelong, unsupervised but safe learning by the robot. As a consequence, a continuous interaction between two learning systems, the robot and the driver, will emerge, hence RADHAR: Robotic ADaptation to Humans Adapting to Robots.

This framework will be tested on a powered wheelchair as a concrete testbed, given that many wheelchair drivers would benefit from navigation assistance, given that they are driving in very challenging 3D dynamic environments, and given that this is typically a very heterogeneous user group with varying skills and abilities. RADHAR will seek a solution for both semi-autonomous and autonomous wheelchair navigation in everyday environments. Semi-autonomous transport is important even if perfect autonomous transport would be available, as it gives more freedom to drivers who still want to drive themselves, as it stimulates people and develops their dexterity for other activities as well (such as for brushing one's teeth or for eating with a fork) and as it may help in reconstructing a damaged brain motor image.

A. Expected contributions by RADHAR

The RADHAR project targets three main project outcomes:

1) Online 3D perception combining laser scanners and vision with terrain traversability analysis at 10 Hz.
2) A navigation assistance framework for fusing environment and user perception, and for taking safe robot navigation actions based on the estimated robot task from uncertain driver signals, with an ability to take navigation decisions at 5 Hz or higher.
3) A repeatable benchmark test to evaluate navigation assistance systems based on driver models.

B. Demonstration platform

Fig. 1 shows one of the 4 demonstrator platforms that will be built. This contains one haptic joystick, a touch screen with GUI, 4 Kinects (1 pointed at the user), 2 bumblebees for outdoor navigation, 3 laser scanners, 2 magnetic encoders, an Xsens IMU, and a single PC. Besides these physical demonstrators, a wheelchair simulator was built on which navigation assistance algorithms can be tested first in a safe manner, and with which users can be trained and selected.

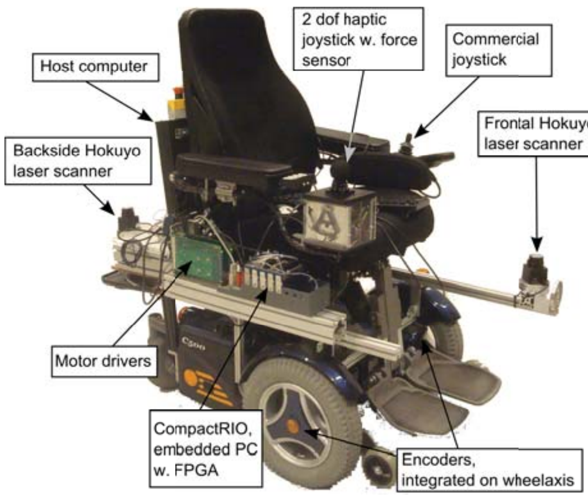


Fig. 1: One of the 4 RADHAR demonstrator platforms.

C. Evaluation by user groups

The RADHAR system will be evaluated by wheelchair drivers from the school Windekind and from the Belgian national centre for Multiple Sclerosis. In particular, 3 groups of wheelchair drivers were selected: a group of starters who did not drive any powered wheelchair before, a group who easily collides when driving due to physical limitations, and a group with mainly cognitive problems, who disregard important aspects of their environment for driving such as other people that are on their path.

D. Overview and system integration

This section provides an overview of the paper and describes at the same time how the different RADHAR components fit together.

Section II describes RADHAR's Bayesian navigation assistance framework in general. This framework estimates the driver's navigation intention from a set of executable robot trajectories and the driver's steering signals. For this, a statistical model of the driver's steering behaviour is learned (Sections II-A to II-C). To verify driver model assumptions such as focus-of-attention, the driver's posture and eye lid state (open or closed) will be continuously estimated with a Kinect (Section II-D).

Besides driver perception and modelling, thorough environment perception and interpretation is required. Section III-A describes the detection and labelling of dynamic obstacles, which will be used as input to a socially compliant motion planner described in Section IV-A. Section III-B describes the building of local maps, which is a necessary input to the motion planner as well.

Based on the estimated map of the local surroundings, based on a traversability analysis and based on the labelled dynamic objects, a socially compliant trajectory is computed as discussed in Section IV-A. In order to compute safe navigation actions the driver's body posture is continuously observed and it is verified whether the driver is not outside a safety volume (Section IV-B). This information will allow us to prevent the wheelchair from driving through a door when the driver's arms are pending outside of the wheelchair for example.

The set of safely executable trajectories in the robot's local environment as well as the probability distribution over these trajectories will be adopted to provide navigation assistance to the driver. As described in Section V, RADHAR aims to perform this using a haptic interface, as this establishes a fast bilateral communication channel between driver and computer.

II. RADHAR's navigation assistance framework

Users with different abilities and needs require a general methodology that can adapt navigation assistance to each individual, and to different types of interfaces. In order to avoid frustrating its user, the robot should recognise navigation plans in a way that considers the user's driving abilities.

We propose a *probabilistic* framework to recognise the user's navigation plans out of a set of *local trajectories*. In order to increase robustness, this framework considers the *uncertainty* when recognising user plans. It fuses past driving information with the user's specific driving style in order to estimate a posterior probability over user plans. Further, these estimated plans are the basis to share the control over the wheelchair according to the user's abilities and needs. The robot ensures the user's safety because it reasons about collision-free trajectories that are also physically executable, i.e. kinematically and dynamically feasible.

The robot first generates all possible user plans for a given situation (*plan generation*). These plans represent all navigation intents in an environment, and take the user's body posture (Section IV-B) and dynamic obstacles (Section IV-A) into account. Next, the robot reasons about the user's abilities to determine the probability of each plan. Calculating the probability of docking at a table (estimating the posterior) directly is more difficult than obtaining the probability of the input signals that are necessary to reach the table, assuming the user has a plan to reach the table in mind. We call the latter probability *user model*. However, the information available to the robot from the user model is quite limited. Often the robot is unable to disambiguate several intentions. Hence, it becomes essential to consider past driving behaviour.

Our probabilistic formulation employs Bayes' theorem to calculate the posterior probability of user plan $\boldsymbol{i}_k$, given the user's current input, $\boldsymbol{u}_k$, and past driving behaviour as follows:

$$\overbrace{p_{post}\left(\boldsymbol{i}_k|\boldsymbol{u}_k,\boldsymbol{\mathcal{H}}_{0:k}\right)}^{\text{posterior over user plans}} = \overbrace{p_{user}\left(\boldsymbol{u}_k|\boldsymbol{i}_k,\boldsymbol{\mathcal{H}}_{0:k}\right)}^{\text{user model}} \cdot \overbrace{p_{prior}\left(\boldsymbol{i}_k|\boldsymbol{\mathcal{H}}_{0:k}\right)}^{\text{prior over user plans}} \cdot \eta \tag{1}$$

where η is a scale factor necessary to normalize the probability distribution. The history $\boldsymbol{\mathcal{H}}_{0:k}$ includes all previous user inputs up to time $k-1$, the sequence of robot actions $\boldsymbol{a}_{0:k}$, the sequence of robot poses, $\boldsymbol{x}_{0:k}$, and any external sensor readings, $\boldsymbol{z}_{0:k}$.

A. Learning the models of plan recognition

In order to recognise the user's navigation plans it is necessary to devise a model of the user's driving, as well as a way to relate past behaviour to the present. The first component is the user model of equation (1), which represents how a specific user transforms mental plans into inputs to the robot. The time evolution, linking the past to the present, concerns how to initialise the prior probability ($p_{prior}(\cdot)$) with the posterior of the previous time step.

We propose to learn the user model from direct observation of the user's driving. The user moves in the environment following a predefined sequence of destinations. Afterwards, during calibration, the complete trajectory of the user is known. Hence, it becomes possible to compare the actual trajectory with the set of local plans at each time step. The local path that resembles the actual trajectory the best is employed to predict the user's input with a probabilistic function estimation technique called Gaussian Process Regression [14]. Please refer to [10] for a complete description of the procedure to learn the user model.

The dataset of human-robot navigation also allows to learn the parameters of the prior in equation (1). Local plans, which resemble the actual trajectory of the user at subsequent times, should inherit probability. We formalise this insight as a series of soft probabilistic constraints in a Dynamic Bayesian Network [13] in order to learn the prior. We explain the complete procedure in [8].

B. Evolution of the framework

The framework to recognise the user's navigation plans was first proposed by Demeester et al. [4]. In this early work, navigation intentions are global destinations in the environment. Later work [5] expands the intention representation with the path to reach the goal, as the same destination might be reached in different ways.

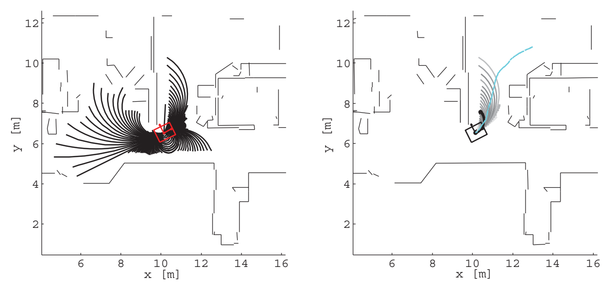
Another important aspect is the impact of steering disabilities on the user model. Some users might be unable to execute all trajectories perfectly. Hünteman et al. study in [9] this effect for a user who is unable to turn left.

A further improvement is to learn the user model directly from driving data. In [10] Hünteman et al. propose a learning framework based on Gaussian Process Regression [14] for user modelling and validate it on the driving pattern of a spastic user. The navigation plans in [10] are local trajectories instead of global plans. These local plans allow the robot to reason *locally* with rich collision information about a user who can drive anywhere. However, linking local plans across time without fixed global references is hard. The dissertation of Hünteman [8] describes a learning framework to learn all parameters of local plan recognition.

C. Analysing the plans of a user with spastic quadriplegia

In order to verify the validity of the approach, we have analysed the navigation plans of a user with spastic quadriplegia driving an electric wheelchair in an indoor environment. No navigation assistance was provided to understand how the user drives without external interference.

Fig. 2 shows a snapshot of the experiment, at which the user is locally turning on the spot after having driven forward. Fig. 2a, shows the set of considered local trajectories for plan recognition. Fig. 2b displays the same plans according to their posterior probability. Darker colours indicate higher probability. A cyan trajectory indicates the future path the user will follow, which allows to compare the estimated navigation plans with the actually executed trajectory.



(a) all local plans adapted to obstacles in the environment

(b) probability of each user plan, where darker colours indicate higher posterior probability

Fig. 2: Recognising the navigation plans of a spastic user.

In this situation, plans with an intermediate curvature have the highest probability. Plans of high and low curvature still remain probable, whereas plans leading backwards are improbable. The most probable plan coincides locally with the future path of the user. For an in-depth analysis of the user's driving style, please refer to the dissertation of Hüntemann [8].

D. Head orientation estimation

The driver model in Equation 1 assumes the driver is focused on the navigation task. However, this may not always be the case. We intend to estimate the driver's focus of attention based on the head orientation, amongst others. For this, a Kinect

Fig. 3: Output of the user head pose estimation system. The green cylinder represents the head orientation.

sensor is mounted on the wheelchair and points towards the driver.

Our approach for 3D head pose estimation [6] does not rely on specific graphics hardware and can be tuned to achieve a trade-off between accuracy and computation cost. We formulate the problem as a regression, estimating the head pose parameters directly from the depth data provided by a Kinect. The regression is implemented within a random forest framework [3], learning a mapping from simple depth features to a probabilistic estimation of real-valued parameters such as 3D nose coordinates and head rotation angles. Since random forests (as any regressor) need to be trained on labeled data and the accuracy depends on the amount of training, we generate an arbitrary number of synthetic training examples without the need of laborious and error-prone annotations. Our system works in real-time on a frame-by-frame basis, without manual initialization. It works for unseen faces and can handle large pose changes, variations such as facial hair, and partial occlusions, e.g. due to glasses, hands, or missing parts in the 3D reconstruction. Moreover, as it does not rely on specific features, e.g. for the nose tip detection, our method can be adapted to the localization of other parts of the face.

Fig. 3 shows an example frame, with our method successfully estimating the head pose even when the nose is badly occluded and thus most of the other approaches based on 3D data would fail. Facial expressions also do not seem to cause problems to the regression in most of the cases, even though the synthetic training dataset contains only neutral faces.

Once the head pose of the user is estimated, it can be used to evaluate his/her attention level by a simple comparison to the motion direction. The assumption here is that a user focused on the navigation task will more likely watch towards the direction in which the wheelchair is moving.

III. Environment modelling

In order to provide safe trajectories for both autonomous and semi-autonomous navigation assistance, a reliable model of the wheelchair's local surroundings should be built. In order to do so, typical dynamic objects in everyday wheelchair environments are detected and labelled (Section III-A), and a 3D model of the environment is built (Section III-B).

(a) (b)

Fig. 4: Sample output when detecting other wheelchairs a) and a person b).

A. Dynamic object detection and classification

To detect dynamic obstacles we rely on the input provided by a Kinect placed in front of the RADHAR wheelchair and pointed towards the environment. Our algorithm is based on Hough forests, which are sets of decision trees learned on the training data. Each tree in the Hough forest maps local appearance of image elements to its leaves, where each leaf is attributed a probabilistic vote in the Hough space. In line with the general random forest paradigm [1, 3], the training process for each tree is governed by a combination of randomization and optimization strategies. The set of leaves in the Hough forest can thus be regarded as an implicit appearance codebook that has been directly optimized for Hough-based detection. Similar to general random forests, Hough forests are efficient to learn and to apply. The combination of the tree structure and simple binary tests makes training and matching against the codebook very fast, whereas clustering-based learning of explicit codebooks as in [12] is considerably more expensive in memory and time. In our case, the random forests have been trained on labeled data captured in the scenario in which the RADHAR wheelchair will most likely navigate. Sample obstacle classes are people and wheelchairs, for which some example detection results are provided in Fig. 4.

B. Robust 3D SLAM

An important building block in autonomous and assisted driving is the construction of a suitable map as representation of the environment around the wheelchair. To properly address tasks like docking at a table or driving on ramps, the RADHAR project puts an emphasis on perceiving and mapping the environment in 3D, and providing methods for path planning and collision avoidance based thereon.

When moving in previously unknown environments, a map can be constructed using simultaneous localization and mapping (SLAM). While laser-based SLAM for robots moving on a 2D plane can be considered a solved problem, the extension to 3D

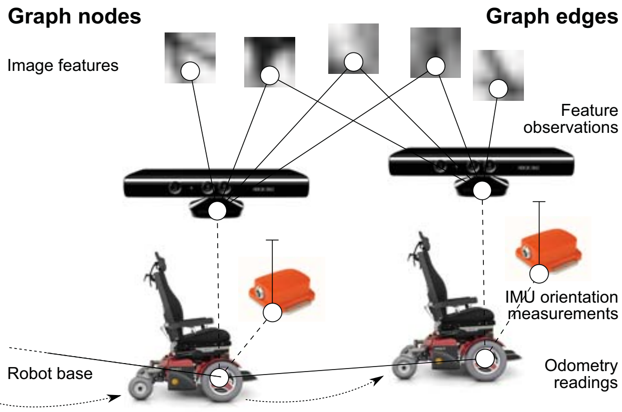


Fig. 5: Observations like tracked features or sensor readings produce constraint edges between consecutive robot poses modeled by nodes in a graph.

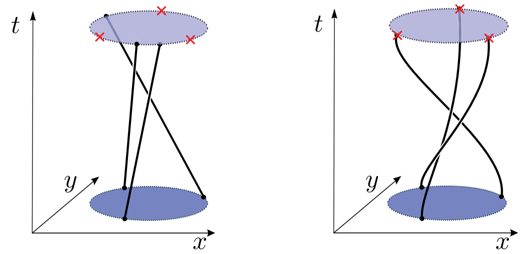


Fig. 6: Joint trajectories comprising three agents. The target positions of the agents are depicted by red crosses. Left: randomly initialized. Right: joint trajectories converged.

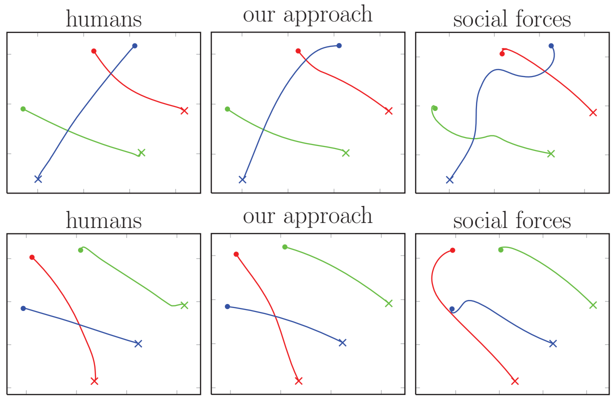


Fig. 7: Predicting real-world human navigation behaviour: fhe figures show a comparison of the human trajectories recorded in a motion capture studio and the trajectories predicted by our approach and by the social forces method.

with cameras instead of laser scanners still poses problems, e.g., when lighting changes occur, or when camera images are blurred due to fast movements.

To achieve robust 3D SLAM, we propose a framework that allows for easy integration of several sensor modalities that complement each other. A modular frontend collects observations from feature tracking on color and depth images, laser scanners, and readings from inertial measurement units and odometry. These observations are treated as constraint edges between consecutive robot poses modeled as nodes in a graph as shown in Fig. 5. Using our general graph optimization framework "g2o" as backend [11], the position of the nodes can be jointly optimized to determine a consistent map.

IV. Trajectory Generation for Autonomous and Semi-Autonomous Navigation

Based on the estimated map of the local surroundings, a traversability analysis and the labelled dynamic objects, a set of trajectories is computed that is compliant with the wheelchair's dynamic environment (Section IV-A). Moreover, the driver's body posture is monitored to detect potentially dangerous situations (Section IV-B).

A. *Socially compliant motion planning*

To allow for socially compliant motion planning, our goal is to learn and imitate human behaviour. In interaction with pedestrians, the autonomous wheelchair should react similarly to a wheelchair that is steered manually. However, humans navigate cooperatively in populated environments and the behaviour of all interacting agents are mutually dependent. The actions of any agent influences the behaviour of its environment, but at the same time its actions are influenced itself by the behaviour of the surrounding agents.

Bennewitz et al. [2] describe how sequential path planning can generate plans for several agents. However, using this approach the plan for each agent only depends on a subset of the other agents. To solve the problem of mutual interaction between all agents, Helbing and Molnar [7] presented the social forces model that aims to describe the local interaction behaviour of humans. Similarly, van den Berg et al. [16] introduced reciprocal velocity obstacles, which also describe rules for cooperative collision avoidance. However, these methods are mainly reactive which results in a limited capability to predict future interaction.

In contrast, we employ joint path planning for actual trajectories of all involved agents including the autonomous wheelchair and humans. For these joint trajectories, we define parameterized features like acceleration and inter-agent distances. Using a maximum entropy-based learning method we compute the probability distribution that closely reproduces the feature values observed in recorded human interactions.

Given a navigation task (cf. Fig. 6 and Fig. 7), the autonomous wheelchair computes the probability distribution over joint trajectories. It then selects the most likely joint trajectory which includes a predicted trajectory for the wheelchair itself. By executing this plan the wheelchair navigates cooperatively through populated environments.

B. *User posture analysis*

Using a Kinect mounted on the RADHAR wheelchair and pointed towards the user, we need to detect the posture of the subject sitting in the wheelchair. The solution we adopted to achieve this goal is to detect the person in the depth image,

Fig. 8: Output of the user posture analysis system. The body parts which are outside of the safety volume are marked in red.

define a safety bounding volume around the chair, and check whether the whole body of the person is inside this volume or not. The solution we propose does not require any initialization and is based on the Flood Fill algorithm applied to depth data. We assume that the central pixel of the image belongs to the person sitting on the wheelchair (this is a reasonable assumption, since the Kinect will be placed in front of the chair, at a distance of about 70cm, thus the person will cover the central part of the image), and we extend the region from that pixel. More precisely, we include a pixel only if its depth value is close enough to the depth of one of its neighbors which already belong to the region. Then the safety volume can be arbitrarily set, depending on the size of the wheelchair. As a final step, we check whether all the 3D locations of the pixels belonging to the person are inside the bounding volume or not. Fig. 8 shows the output of our system: the parts of the body inside the safety volume are depicted in green, while those outside it are shown in red.

V. Haptic feedback and shared control

Given an estimation of the safe trajectories the driver most probably wants to execute, the RADHAR system is now able to provide actual navigation assistance to the driver. This will be performed by adopting a novel haptic joystick.

A. Haptic navigation assistance

Touch and kinesthesis (in short *haptics*) are subtle, effortless senses that are critically important for fast and accurate interaction with our environment. While picking up a pencil a complex interaction and exchange of forces takes place that allows us to grasp it in a stable manner without slipping nor thumbling. An interaction of that complexity takes place almost effortlessly, requiring hardly any mental effort. A *haptic display system* is a robotic joystick that can be programmed to exert well-controlled forces upon its user. Connected to a virtual reality environment with dynamics engine such joystick can replicate the forces that arise when e.g. picking up the pencil. In theory, and when well designed a user might find it troublesome to distinguish a virtual from a real pencil. Such system would thus feel natural and be intuitive to interact with.

Forces displayed by the haptic device should not necessarily replicate a purely physical phenomenon in order to be perceived as natural or intuitive. RADHAR looks at ways to *encode* navigation assistance schemes and present them via a haptic joystick to the wheelchair driver. If designed well, such *haptic environmental feedback* could feel intuitive and allow effortless navigation through complex environments. However, since there is often no physical equivalent for a navigation assistance scheme, finding an intuitive encoding is not straightforward at all. Furthermore, since wheelchair users often have limited power and get easily tired, haptic feedback should not increase the overall effort.

Compared to traditional navigation assistance or so-called shared control schemes [15], where users only perceive the decisions by the navigation system *after* wheelchair displacement, haptic guidance allows a more profound sharing of control. The user can negotiate directly with the navigation assistance system over this fast *bilateral* communication channel and should thus encounter less surprises on how the wheelchair will move (Fig. 9). Furthermore, by applying sufficient force upon the joystick, the user can always *overrule* the suggestion by the assistance system. So, in the end, the user can keep the control over the system.

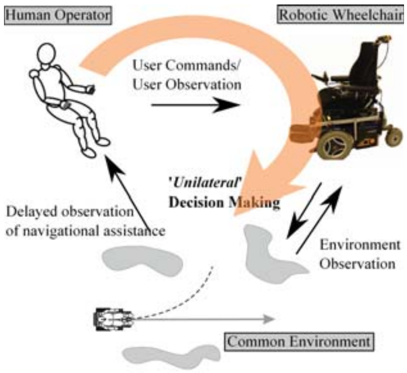
B. Haptic obstacle avoidance along circular paths

A novel haptic guidance scheme was developed to help steer the powered wheelchair users through narrow and complex environments [17]. The scheme encodes the local environment of the wheelchair as a set of collision-free circular paths. An adaptive impedance controller is then tuned based on this encoding so that it slows the wheelchair down in the direction of short circular paths (thus with imminent collisions) and bends the user towards longer circular paths (helping to avoid the collisions). Fig. 10 displays the resistance a user would feel at the joystick, for an environment and circular encoding displayed in 10a.

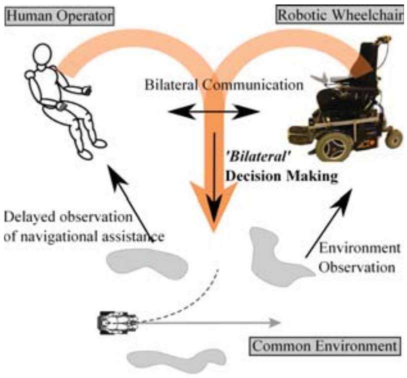
C. Experiments with backwards driving into an elevator

Preliminary experiments were conducted with this kind of haptic navigation assistance. With cardboard boxes an artificial environment was built up to represent an elevator. The user is asked to maneuver the wheelchair backwards inside this elevator beginning from a fixed starting position. The maneuvering capability with or without navigation assistance is measured during the execution of this task. Parameters that were recorded are time until completion and the number of collisions. At this stage of the research all experiments are conducted by one single able-bodied user (male, 33 years) with limited expertise in driving powered wheelchairs. Both experiments were conducted 10 times. Three types of experiments were conducted and executed in random order:

Type 1: navigation without guidance. The user was allowed



(a) Traditional shared control schemes display asymmetry in the decision making process. Users only understand decisions after actual displacement took place.



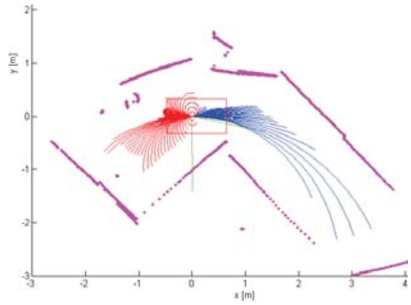
(b) Bilateral shared control schemes feature a deeper level of control sharing. The user can negotiate directly with the wheelchair over a haptic communication channel.

Fig. 9: Navigation assistance through unilateral or bilateral shared control.

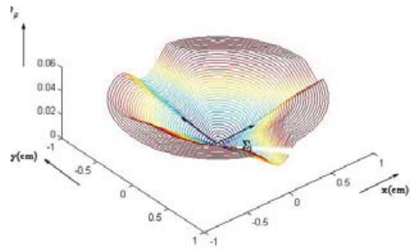
to look backwards over the shoulder during these experiments. Note that this way of operation is not possible or very tiring for many typical wheelchair users.

Type 2: navigation with visual guidance. The user was asked to maneuver the wheelchair while observing a GUI that displayed the environment encoded as a set of collision-free path lengths. **Type 3: navigation under haptic feedback**. The user was asked to drive 'blindly' inside the elevator solely relying on his sense of touch and the haptic guidance.

Table I and Fig. 11 summarize the results from the different experiments. At this moment, navigating the wheelchair while looking backwards over the shoulder is still superior, but, also here, collisions could not be avoided. Indeed, the task is quite challenging as the elevator is quite narrow, leaving only about 10 centimeters of space at both sides between wheelchair and door post. With only visual or haptic guidance, the amount of successful executions dropped to $6/10$. This score might seem quite low, but it must be stressed that the user did not have to look backwards over the shoulder. So such navigation strategy could come in handy to help especially those users that experience problems in looking over their shoulder while steering a wheelchair.



(a) Collision-free paths.



(b) Associated force field.

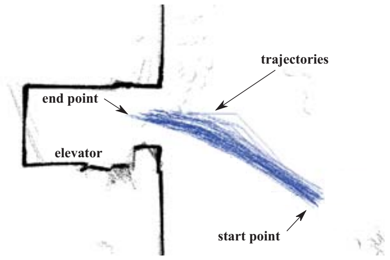
Fig. 10: Collision-free paths and associated force field in haptic joystick.

TABLE I: Summary of results (time in s, average and standard deviation calculated for successful runs only).

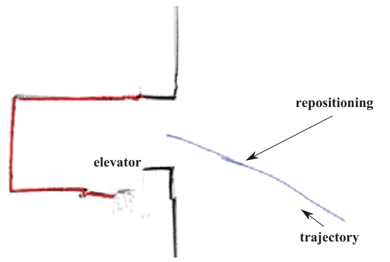
run	1	2	3	4	5	6	7
type 1	13.23	13.15	10.11	14.31	11.94	9.45	9.69
type 2	11.00	12.09	11.47	14.01	9.6	10.34	17.51
type 3	25.4	10.43	21.86	15.23	19.48	13.37	26.65
run	8	9	10	**av.**	**stdev**	**coll.**	
type 1	10.46	12.07	10.14	11.14	1.47	1	
type 2	11.84	9.82	15.73	11.60	1.48	4	
type 3	13.02	40.83	17.94	18.23	5.89	4	

At this moment GUI-based navigation is still faster than navigating solely based upon haptic guidance (Table I). Under haptic guidance the user is somehow 'palpating' the environment to feel where the passage is, whereas the GUI immediately shows the user where the passage is. On the other hand the haptic guidance warns the user when a collision is near. Fig. 11b shows an exemplary trajectory where the user turns the wheelchair after such warning and successfully completes the task.

There is definitely still some room for improvement. But note that it is quite remarkable that already now in $6/10$ cases the user manages to drive into the elevator solely relying on haptic guidance. Without such guidance, without GUI or looking backwards over the shoulder such maneuver would be close to impossible.



(a) All trajectories of the experiments plotted out. The clock is stopped when the wheelchair's frontal wheels are inside the elevator.



(b) Example trajectory under haptic guidance. Through haptic feedback the user turns wheelchair closeby elevator and makes a successful entry

Fig. 11: Navigation trajectories when driving backwards into an elevator.

VI. Conclusion

This paper has described the main technological components of the RADHAR navigation assistance approach. Experimental results for each component separately have been shown. These components are now being integrated into one system, so that it can be tested by the user groups.

Acknowledgment

The research leading to these results has received funding from the European Community's Seventh Framework Programme FP7/2007-2013 - Challenge 2 - Cognitive Systems, Interaction, Robotics - under grant agreement No. 248873-RADHAR.

References

[1] Y. Amit and Donald Geman Y. Shape quantization and recognition with randomized trees. *Neural Computation*, 9:1545–1588, 1997.

[2] M. Bennewitz, W. Burgard, and S. Thrun. Optimizing schedules for prioritized path planning of multi-robot systems. In *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, volume 1, pages 271 – 276 vol.1, 2001.

[3] L. Breiman. Random forests. *Mach. Learn.*, 45(1):5–32, October 2001. ISSN 0885-6125.

[4] E. Demeester, M. Nuttin, D. Vanhooydonck, and H. Van Brussel. A model-based, probabilistic framework for plan recognition in shared wheelchair control: Experiments and evaluation. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1456–1461, 2003.

[5] E. Demeester, A. Huntemann, D. Vanhooydonck, G. Vanacker, H. Van Brussel, and M. Nuttin. User-adapted plan recognition and user-adapted shared control: A bayesian approach to semi-autonomous wheelchair driving. *Journal of Autonomous Robots*, 24(2):193–211, February 2008.

[6] G. Fanelli, T. Weise, J. Gall, and L. Van Gool. Real time head pose estimation from consumer depth cameras. In *33rd Annual Symposium of the German Association for Pattern Recognition (DAGM'11)*, 2011.

[7] D. Helbing and P. Molnar. Social force model for pedestrian dynamics. *Physical Review E*, 51:4282, 1995.

[8] A. Hüntemann. *Probabilistic Human-Robot Navigation – Plan recognition, user modelling and shared control for robotic wheelchairs*. PhD thesis, Katholieke Universiteit Leuven, March 2011.

[9] A. Huntemann, E. Demeester, G. Vanacker, D. Vanhooydonck, J. Philips, H. Van Brussel, and M. Nuttin. Bayesian plan recognition and shared control under uncertainty: Assisting wheelchair drivers by tracking fine motion paths. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3360–3366, 2007.

[10] A. Huntemann, E. Demeester, M. Nuttin, and H. Van Brussel. Online user modelling with gaussian processes for bayesian plan recognition during power-wheelchair steering. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 285–292, 2008.

[11] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard. g2o: A general framework for graph optimization. In *IEEE Intl. Conference on Robotics and Automation (ICRA)*, China, May 2011.

[12] B. Leibe, A. Leonardis, and B. Schiele. Robust object detection with interleaved categorization and segmentation. *Int. J. Comput. Vision*, 77(1-3):259–289, May 2008. ISSN 0920-5691.

[13] K.P. Murphy. *Dynamic bayesian networks: representation, inference and learning*. PhD thesis, University of California, Berkeley, 2002.

[14] C.E. Rasmussen and C.K. I. Williams. *Gaussian Processes for Machine Learning*. Adaptive Computation and Machine Learning. The MIT Press, second edition, 2006. ISBN 0-262-18253-X.

[15] R. C. Simpson. Smart wheelchairs: A literature review. *J Rehabil Res Dev*, 42(4):423–436, 2005. ISSN 0748-7711.

[16] J. van den Berg, S.J. Guy, Lin M., and D. Manocha. Reciprocal n-body collision avoidance. In *International Symposium on Robotics Research (IJRR)*, 2009.

[17] E.B. Vander Poorten, E. Demeester, E. Reekmans, J. Philips, A. Huntemann, and J. De Schutter. Powered wheelchair navigation assistance through kinemally correct environmental haptic feedback. In *IEEE Conference on Robotics and Automation*, Saint Paul, Minnesota, May 2012.