Auton Robot (2008) 24: 193–211
DOI 10.1007/s10514-007-9064-5

User-adapted plan recognition and user-adapted shared control: A Bayesian approach to semi-autonomous wheelchair driving

Eric Demeester · Alexander Hüntemann · Dirk Vanhooydonck · Gerolf Vanacker · Hendrik Van Brussel · Marnix Nuttin

Received: 5 February 2007 / Accepted: 2 November 2007 / Published online: 15 December 2007
© Springer Science+Business Media, LLC 2007

Abstract Many elderly and physically impaired people experience difficulties when maneuvering a powered wheelchair. In order to ease maneuvering, powered wheelchairs have been equipped with sensors, additional computing power and intelligence by various research groups.

This paper presents a Bayesian approach to maneuvering assistance for wheelchair driving, which can be adapted to a specific user. The proposed framework is able to model and estimate even complex user intents, i.e. wheelchair maneuvers that the driver has in mind. Furthermore, it explicitly takes the uncertainty on the user's intent into account. Besides during intent estimation, user-specific properties and uncertainty on the user's intent are incorporated when taking assistive actions, such that assistance is tailored to the user's driving skills. This decision making is modeled as a greedy Partially Observable Markov Decision Process (POMDP).

Benefits of this approach are shown using experimental results in simulation and on our wheelchair platform Sharioto.

Keywords Plan recognition · Intent estimation · Shared control · User adaptation · User modeling · Partially observable Markov decision process

1 Introduction

Service robots, and assistive robots in particular, differ from traditional industrial robots in their tight collaboration with humans. This imposes specific requirements on these robots' behavior and their physical embodiment. Improvements to current assistive robot designs may concern the robot's physical embodiment, the human-robot interface, or the estimation and decision making algorithms that determine the robot's behavior. Some of our previous work has focused on embodiment and human-robot interfacing (Kröse et al. 2003), with a special focus on emotional feedback, speech interaction, and user intent estimation based on sound localization and ambient intelligence. For this, the robot Lino shown in Fig. 1 was adopted. This paper's focus is more on the control and estimation algorithms behind the robot's interaction skills. Driving with a robotic wheelchair is adopted as a test case for the proposed approach. We assume that a conventional user interface and a conventional wheelchair platform are adopted, which are already optimally adapted to the driver by human factor specialists, and we require our control and estimation algorithms to take these constraints into account. Therefore, the framework should be easily adapted to different user interfaces and robot platforms.

1.1 Motivation

Research on assistive robots is motivated by social, economic and scientific driving forces. The first two driving forces are related to the ageing of society occurring in most countries and to the presence of physical impairment in general. Members of this increasingly important section of our society typically suffer from a reduced mobility. A loss of mobility is often accompanied by a severe decrease of social activity and social contact, an increased dependence on others and a restriction of one's self-determination. This may considerably affect a person's self-esteem, dignity and happiness in general (Cooper and Cooper 2003). Various

E. Demeester (✉) · A. Hüntemann · D. Vanhooydonck · G. Vanacker · H. Van Brussel · M. Nuttin
Department of Mechanical Engineering, Katholieke Universiteit Leuven, Celestijnenlaan 300-B, 3001 Heverlee, Belgium
e-mail: eric.demeester@mech.kuleuven.be

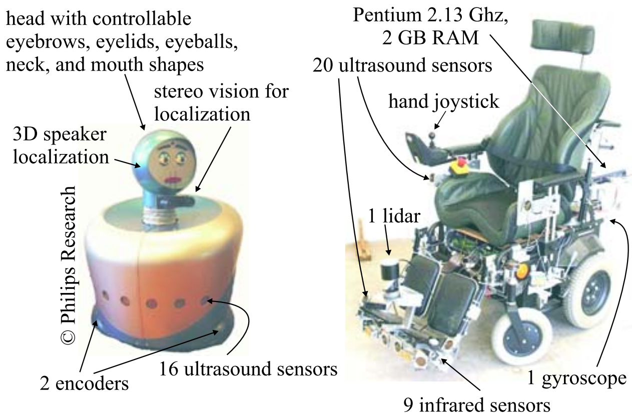

Fig. 1 The autonomous service robot Lino (*left*) and the semi-autonomous assistive wheelchair Sharioto (*right*)

tools are commercially available that increase the mobility of the physically impaired, such as (powered) wheelchairs, walkers, or robotic manipulators. Unfortunately, many of the common every-day-life maneuvers such as docking at a table or driving through a door are experienced as difficult, time-consuming or annoying. Severe accidents such as falling down stairs or ramps, collisions with other chairs or persons, and getting blocked in corridors or elevators regularly occur. In other cases, fatigue or frustration results from controlling these devices. This is partly due to the drivers' inability to give accurate motion commands, which is possibly aggravated by visual impairments (Fehr et al. 2000). Furthermore, the relatively large size of wheelchairs compared to available space in public and private buildings, the presence of castor wheels that may disturb desired wheelchair behavior, the limited field of view when sitting in a wheelchair, and the wheelchair's kinematics and dynamics make these devices difficult to control, sometimes even for able-bodied drivers. A description of these and other user requirements and challenges can be found in (STWW project homepage 2007).

For these reasons, several existing mobility tools have been equipped with sensors and a computerized controller to aid the physically impaired with every-day-life maneuvering. Examples include wheelchairs (Tzafestas 2001; Simpson 2005), walkers (Yu et al. 2003), robotic guide canes for the visually impaired (Aigner and McCarragher 2000), and robotic manipulators (Martens et al. 2001; Tsui and Yanco 2007).

The *social* driving force behind mobile assistive robotics research can be summarized as relieving the lack of mobility with the purpose of increasing the driver's quality-of-life, of reducing the number of accidents, of decreasing the travel time, and of protecting surrounding furniture and coexisting people. The *economic* driving force behind assistive robotics research boils down to a reduction of future social security expenditures. Although the economic impact is hard to predict, the goal of this research is to reduce the number of people that are prematurely sent to nursing homes by increasing their autonomy. This should decrease the costs for individual health care. The *scientific* driving force behind assistive robotics research corresponds in our opinion to the quest for natural human-robot cooperation. Can robots interact reliably and naturally with humans, and if so, how? This quest has lead us to the research problems of plan recognition and shared control.

1.2 Plan recognition and shared control

Providing active navigation aid to a driver who is herself[1] in control of a vehicle requires the robotic system to decide in a proper manner to which degree corrective actions should be taken. This is the key issue in *shared control*, which we define as situations where control over a system is shared among one or more humans and one or more computerized controllers. Related definitions can be found in (Sheridan 1992). The purpose of shared control is to combine the strengths of human and machine and to reduce their weaknesses. For example, humans are typically good at global planning and coarse control, whereas machines are good at fine motion control.

In order to take acceptable corrective actions, socially assistive robots should be aware of the user's intent. For example, it is of no use if a robot aids a user with docking at a table if the user wants to avoid the table in the first place. Furthermore, in order for vehicle control to be easy and intuitive, we believe that drivers should not be required to explicitly communicate their navigation plan prior to executing the maneuver. For a large part of our user group, explicitly stating which task should be executed constitutes a cognitively and physically challenging or even impossible task. Instead, we consider it the task of the assistance system to infer the user's plan from the user's noisy actions and from environmental perception. This is the problem of *plan recognition* or *intent estimation* (Carberry 2001). The problem of plan recognition is more formally defined as "taking (...) *as input a sequence of actions performed by an actor and to infer the goal pursued by the actor and also to organize the actions in terms of a plan structure*" (Schmidt et al. 1978). Due to the noisy user signals or the limited set of user actions that are possible with certain user interfaces, the user's intent is inherently uncertain.

1.3 Paper overview

Figure 2 depicts the role of plan recognition and shared control on our robotic wheelchair Sharioto. It furthermore introduces the symbols that will be used. Wheelchair drivers

[1] In order to avoid tiresome reading due to the use of '*(s)he*' and '*his or her*', this paper adopts only female characters for anonymous individuals.

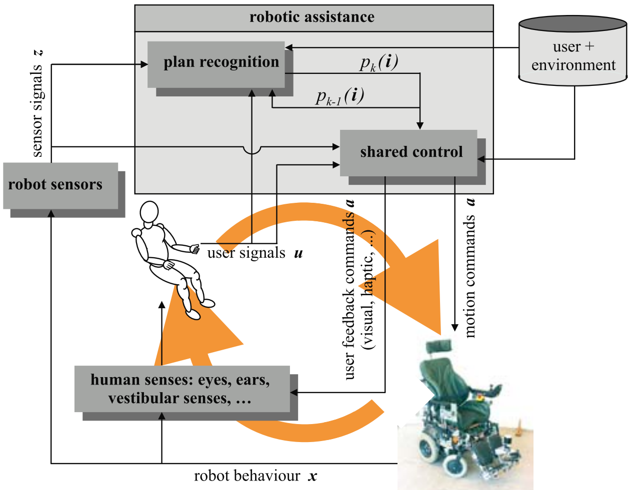

Fig. 2 Wheelchair drivers issue signals $\boldsymbol{u}$ to control their vehicle. These signals together with sensor signals z and past beliefs in user plans $p_{k-1}(\boldsymbol{i})$ are interpreted by the plan recognition module, resulting in new user plan estimates $p_k(\boldsymbol{i})$. These estimates $p_k(\boldsymbol{i})$ are in turn adopted by the shared control module, which produces the actual assistive actions $\boldsymbol{a}$. Through her senses, the user observes the robot behavior $\boldsymbol{x}$ caused by the actions $\boldsymbol{a}$, and reacts to this by issuing new commands $\boldsymbol{u}$. This way, the user in effect closes the loop

issue signals $\boldsymbol{u}$ to control their vehicle. User signals can be given through a variety of interfaces, both continuous interfaces such as hand or chin joysticks and discrete interfaces such as switch-based or sip-and-puff systems. In this paper, experimental results with both a joystick and a button interface will be presented. User signals $\boldsymbol{u}$ together with signals z from the robot's sensors and past beliefs in user plans $p_{k-1}(\boldsymbol{i})$ at time $k-1$ are interpreted by the plan recognition module, resulting in new user plan estimates $p_k(\boldsymbol{i})$ at time k. The Bayesian formalism is adopted to take uncertainty into account and to merge past and present information. Hence, p_k represents the probability distribution over user intents $\boldsymbol{i}$. We model the state $\boldsymbol{i}$ that is continuously estimated by the plan recognition module as a trajectory that the user has in mind from her current location to a goal location. Another important element in this Bayesian framework is the user model, which predicts which signals a specific wheelchair driver gives in order to control the wheelchair. The user plan estimates $\boldsymbol{i}$ are adopted by the shared control module, which produces the actual assistive actions $\boldsymbol{a}$. This paper focuses on wheelchair actions only, but other types of user feedback actions may be adopted such as haptic or visual feedback. Through her senses, the user observes the robot behavior $\boldsymbol{x}$ caused by the actions $\boldsymbol{a}$, and reacts to this by issuing new commands $\boldsymbol{u}$.

Section 2 presents the proposed Bayesian plan recognition framework. Section 3 demonstrates how user plan estimates are adopted to make user-adapted, assistive actions under uncertainty. Section 4 presents experimental results regarding plan recognition, user modeling and shared control in simulation and on the wheelchair Sharioto. Finally, Sect. 5 compares current approaches to driving assistance with the presented framework in this paper and discusses related work regarding plan recognition and user modeling.

2 User-adapted plan recognition

The purpose of the presented Bayesian plan recognition framework is threefold. First, we would like to recognize even complex user plans such as parking maneuvers in narrow spaces in an accurate way. Second, the uncertainty on these plan estimates should be determined. Third, the estimation algorithm should take general human characteristics into account such as the maximum possible human control bandwidth, and should also be adaptive to the specific user that is interacting with the robot. In order to realize these requirements, a representation of user plans is chosen that is different from previous approaches, which nearly all represent intents in terms of tasks such as *follow-corridor* or *avoid-obstacle*, which are usually specified in terms of inaccurate language concepts.

2.1 Representation of user plan hypotheses

Generally speaking, wheelchair drivers want to reach a certain goal pose $\boldsymbol{p}_{goal} = [x_{goal}\ y_{goal}\ \theta_{goal}]^T$ with a certain goal twist $\boldsymbol{t}_{goal} = [v_{goal}\ \omega_{goal}]^T$, where θ_{goal} represents the robot orientation at the goal position $[x_{goal}\ y_{goal}]^T$, and where v_{goal} denotes the desired linear velocity and ω_{goal} the desired rotational velocity at $\boldsymbol{p}_{goal}$. A twist $\boldsymbol{t}$ and pose $\boldsymbol{p}$ will be represented jointly as the robot state $\boldsymbol{x}$. A user plan $\boldsymbol{i}_k$ at time k can then be generically described as trajectory

$$\boldsymbol{i}_k = \{\boldsymbol{x}_{current}, \dots, \boldsymbol{x}_{goal}\}, \tag{1}$$

which the user has in mind to achieve the goal state $\boldsymbol{x}_{goal}$ from the current robot state $\boldsymbol{x}_{current}$. *Any* user plan can be modeled with this representation in a precise way, though we do not assume that users wish to follow an exact geometric path. Furthermore, it is not required for the trajectory to be physically executable, in order to be able to model e.g. plans of unexperienced drivers. Inclusion of the robot twist in the plan representation may be useful to estimate if drivers want to accelerate or decelerate. Initially, only goal states were incorporated in our user plan representation (Demeester et al. 2003). However, users may have a specific trajectory in mind to reach the desired goal state from the robot's current state. If the assistive robot correctly estimates the user's desired goal state, but assists the user by following another trajectory to the same goal state than expected by the user, the user may be confused due to a discrepancy between her expectations and the robot behavior, and she may feel that the robot is not assisting her in the correct way.

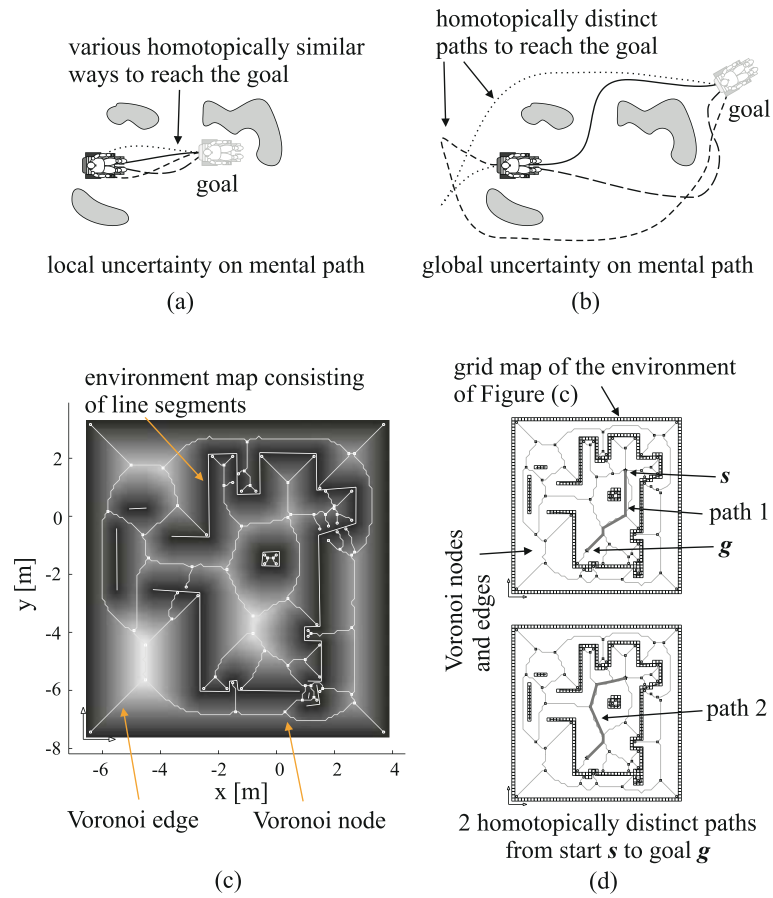

Fig. 3 This figure illustrates the presence of local (**a**) and global (**b**) uncertainty on mental paths to goal states. In order to determine the global uncertainty, a Voronoi graph is built (**c**) for an estimated map and this graph is searched for distinct paths (**d**). Loops in the paths are not considered

Therefore, we include the trajectory to the goal location in the user intent representation.

2.2 Generation of user plan hypotheses

Hypotheses regarding user plans can be generated in a variety of ways. In this paper, a two-step approach will be adopted. First, all plausible goal state candidates are generated, and in a second phase all trajectories to these goal states.

In the first step, goal state candidates can be learned by remembering at which poses the user stands still for a certain amount of time. Goal state candidates can also be generated by recognizing geometric patterns in the environment such as doors or tables, and by generating one or more goal states in the neighborhood of each of these patterns. For example, recognizing that the robot is in a corridor allows to generate goal states all along the corridor. Furthermore, a user or physiotherapist can generate goal state hypotheses by indicating on an estimated map all possibly interesting places. This allows to take goal states into account that are otherwise difficult to recognize using sensor information only, such as light switches.

In a second step, trajectories to the candidate goal states should be generated. Goal states may be reached in a variety of ways. There may be both *global* and *local* uncertainty on trajectory candidates to a goal state. Global uncertainty stems from the various fundamentally different ways to reach a goal state, which are called homotopically distinct trajectories in motion planning (Latombe 1991). This global uncertainty is illustrated in Fig. 3(b) and is determined in this work by first building the Voronoi graph for the estimated map of the environment. A Voronoi graph consists of the set of points that are equidistant to at least two obstacles. Then, all distinct paths from a start to a goal position are found in the Voronoi graph by exhaustively searching the graph of Voronoi nodes and edges. Loops in the paths are not considered since these are assumed to be irrational for most users. Figures 3(c) and (d) illustrate this approach. On each of the paths found in this way, some local uncertainty is present as depicted in Fig. 3(a). At this moment, this local uncertainty on paths is not explicitly modeled and incorporated in our approach.

The trajectory $\{\boldsymbol{x}_{current}, \ldots, \boldsymbol{x}_{goal}\}$ corresponding to a certain intent $\boldsymbol{i}_k$ is calculated by a geometric fine motion planner. In order to be able to recognize also complex user plans such as docking and parking maneuvers, the fine motion planner takes the robot's kinematics, orientation and geometry explicitly into account. It is imperative to recognize these user plans, as elderly and disabled tend to experience more difficulties with these maneuvers. Most current approaches approximate one or more of these constraints. Consequently, it seems unlikely that they are able to recognize these complex maneuvers.

The fine motion planner first constructs the free configuration space, which consists of all discretized 3D robot poses $[x\ y\ \theta]^T$ that do not collide with a modeled object. Then, a search algorithm is adopted that searches for an optimal path from the current robot pose $\boldsymbol{p}_{current}$ to each intent's goal pose $\boldsymbol{p}_{goal}$. The search algorithm constructs edges between free 3D configurations that point in the direction of the minimum cost path, where the edges respect the robot's kinematic constraints. The algorithm makes a compromise between intrinsic costs (such as the distance to obstacles) and transition costs (such as costs for moving backwards or for traveled distance). The search algorithm constructs a cost function over collision-free cells by assigning to all reachable free cells a value that indicates the cost to go to the cell from the current pose. The algorithm finishes when all reachable free cells are visited and have an optimal value. Finding the optimal path from a goal cell to the current robot pose then simply boils down to following the gradient of the cost function. These paths are computed almost instantaneously once the cost function is constructed. In a final step, the discretized path is smoothed.

In order to speed up the generation of motion plans, a multi-resolution approach is adopted, where high-resolution 3D planning is only performed in the neighborhood of the robot, and low-resolution 3D or 2D planning is performed farther away from the current pose. Figure 4 shows a typical set of local 3D paths to various local goal states. This search algorithm is re-run each time the probability distribution over user plans is recalculated, which allows to deal with changing environments as path generation is based on the latest sensor data. More extended information about the planning algorithm can be found in (Demeester et al. 2005).

Several essential differences exist between assistive robot types. These differences have strong repercussions on plan recognition and the generation of plan hypotheses, but they do not seem to have been truly acknowledged before. For robotic *guide dogs* and *robotic walking assistants*, it is mainly the user who provides the dynamics of the human-machine system, because in most cases she is responsible for pushing the robot ahead. This furthermore results in a system that is unlikely to move backwards. Also the shape of the human-machine system is to a large extent determined by the human, and this may be rather variable. Furthermore, the inputs of users to *walking assistants* are usually forces and torques, and thus rather different from wheelchairs, where often joysticks are used. Users adopting *robotic guide dogs* are visually impaired by definition. This prohibits them to visually plan local maneuvers, especially if unforeseen obstacles are present. Therefore, their plans may

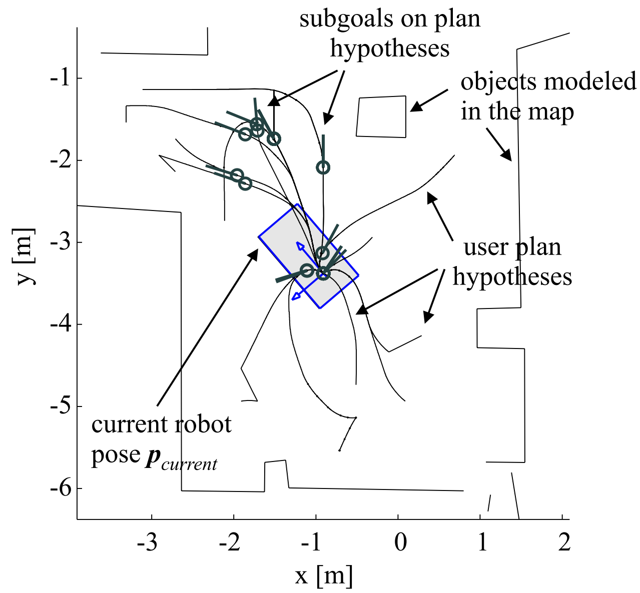

Fig. 4 House-like environment where experiments with the wheelchair Sharioto have been conducted. The figure depicts the wheelchair along with calculated paths to possible goal poses at a certain moment in time. The paths are computed with a planner in the (x, y, θ) configuration space. The planner takes the robot geometry and kinematic constraints into account. Also shown are subgoals on the paths, which are adopted to predict the user signals

be much coarser than plans of users without visual impairments. Nevertheless, the user may still have a global plan in mind if she is aware of where she is in a certain environment. *Wheelchair drivers* are basically sitting on top of the robot. The dynamics and the shape of the human-machine system are thus to a large extent determined by the wheelchair. Wheelchair drivers are regularly forced to drive backwards, due to the narrow places in which they have to move. During these maneuvers, their field of view may be strongly limited, and hence their knowledge of surrounding obstacles to mentally plan paths.

2.3 Bayesian framework for plan recognition

Figure 5 depicts the Bayesian network model behind our plan recognition framework. The human and robot part have been indicated separately. The robot can be approximated fairly accurately as a first-order Markov model. The robot state $\boldsymbol{x}_k$ can be predicted from its previous state $\boldsymbol{x}_{k-1}$ and the shared control action $\boldsymbol{a}_k$ applied to the robot. The sensor measurements z_k at time k only depend on the robot state $\boldsymbol{x}_k$ and the map $\boldsymbol{m}$, which is assumed to be static in this model. Sensor measurements can often be approximated to be conditionally independent, i.e. conditioned on the robot pose.

The human driver on the other hand is assumed to have some goal location $\boldsymbol{x}_k^g$ in mind and to observe the current robot state $\boldsymbol{x}_k$ at each time instant k. It is assumed that

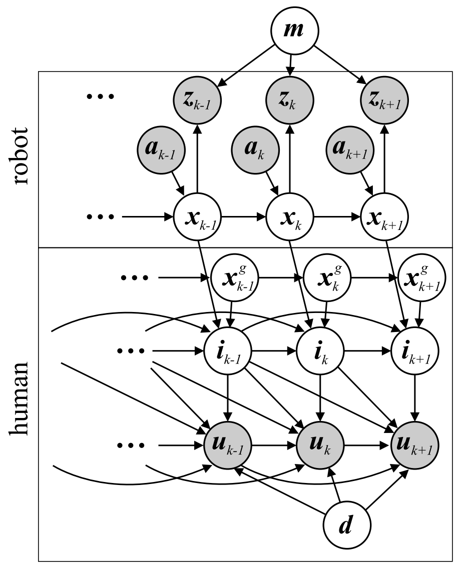

Fig. 5 Graphical model of plan recognition with a time window of 3 time steps

the mental goal location $\boldsymbol{x}_k^g$ does not change continuously but only at a slow rate, for example when the user actually reaches that location or when the user changes her mind. The user is furthermore assumed to make a mental plan $\boldsymbol{i}_k$ to reach the goal location, based on an observation of the environment[2] $\boldsymbol{m}$ and through some mental processes. For users with cognitive problems such as amnesia, who do not know which goal location they desire to reach or how to reach a goal location, some form of autonomous driving to the desired goal state should be adopted. This paper focuses on users who have a navigation plan in mind to reach a desired goal location. The human's mental plan $\boldsymbol{i}_k$ may depend on more than just the previous plan $\boldsymbol{i}_{k-1}$. In order to execute her plan, the user issues signals $\boldsymbol{u}_k$. These signals may depend on earlier interface signals $\boldsymbol{u}_{k-m+1:k-1} = \{\boldsymbol{u}_{k-m+1}, \ldots, \boldsymbol{u}_{k-1}\}$, on earlier user plans $\boldsymbol{i}_{k-m+1:k-1} = \{\boldsymbol{i}_{k-m+1}, \ldots, \boldsymbol{i}_{k-1}\}$ and on user model parameters $\boldsymbol{d}$, which may e.g. incorporate the user's disorder or mental models maintained by the user about the robot. User signals are not necessarily conditionally independent. The main reason for this is that the user closes the loop and acts (at least partly) as a feedback controller, as shown in Fig. 2.

It is assumed here that information from at most m past time steps influences the current user plan and user signal. Figure 5 shows a model for which $m = 3$. Therefore, in order to capture the user's intent, at time $k-1$ a probability distribution is maintained over the set of possible user intents $\boldsymbol{i}_{k-m:k-1}$. At time k, new user plan hypotheses $\boldsymbol{i}_k$ are generated using the approach of Sect. 2.2. These hypotheses are added to the state, which becomes $\boldsymbol{i}_{k-m:k}$. Based on the actual interface signals $\boldsymbol{u}_k$ and the robot behavior encoded as $\boldsymbol{x}_k$, the probability function over $\boldsymbol{i}_{k-m:k}$ is updated as follows:[3]

$$\begin{aligned}
&p_k(\boldsymbol{i}_{k-m:k}|\boldsymbol{u}_{k-m:k}) \\
&\quad = p_{user}(\boldsymbol{u}_k|\boldsymbol{i}_{k-m:k}, \boldsymbol{u}_{k-m:k-1}) \\
&\qquad \cdot p_{process}(\boldsymbol{i}_k|\boldsymbol{i}_{k-m:k-1}, \boldsymbol{u}_{k-m:k-1}) \\
&\qquad \cdot p_{k-1}(\boldsymbol{i}_{k-m:k-1}|\boldsymbol{u}_{k-m:k-1}) \cdot \eta \\
&\quad \overset{\text{assumptions}}{=} p_{user}(\boldsymbol{u}_k|\boldsymbol{i}_{k-m+1:k}, \boldsymbol{u}_{k-m+1:k-1}) \\
&\qquad \cdot p_{process}(\boldsymbol{i}_k|\boldsymbol{i}_{k-m:k-1}) \\
&\qquad \cdot p_{k-1}(\boldsymbol{i}_{k-m:k-1}|\boldsymbol{u}_{k-m:k-1}) \cdot \eta
\end{aligned} \tag{2}$$

where:

1. p_{k-1} is the a priori distribution over user intents, given all user signals $\boldsymbol{u}_{k-m:k-1}$. It reflects the belief in the different possible user intents prior to having moved and prior to having taken new user signals into account.
2. p_{user} is the *user model*, which expresses the likelihood that the user gives the observed interface signal $\boldsymbol{u}_k$, given that the user has had intent evolution $\boldsymbol{i}_{k-m+1:k}$, and given all user signals $\boldsymbol{u}_{k-m+1:k-1}$.
3. $p_{process}$ is the *plan process model*, which determines both the shape and the probability of a user plan $\boldsymbol{i}_k$ at time k, given that the user has had intent evolution $\boldsymbol{i}_{k-m:k-1}$.
4. p_k is the a posteriori distribution over user intents, i.e. the probability of the different user plans after user signals and wheelchair motion have been taken into account.
5. η is a scale factor to normalize the probability distribution.
6. the assumptions made in the second equation correspond to (1) the assumption that only a time window of size m is adopted, and (2) the assumption that the mental plan construction process is independent of previously given user signals.

A priori determined parameters $\boldsymbol{\theta}_{k-m:k-1}$ such as maps $\boldsymbol{m}$ or user model parameters $\boldsymbol{d}$ have been left out for notational simplicity. Because of the assumption that only information in the time window m is required, marginalizing p_k over $\boldsymbol{i}_{k-m}$ allows to keep the state size fixed:

$$\begin{aligned}
&p_k(\boldsymbol{i}_{k-m+1:k}|\boldsymbol{u}_{k-m+1:k}) \\
&\quad = \sum_{\boldsymbol{i}_{k-m}} p_k(\boldsymbol{i}_{k-m:k}|\boldsymbol{u}_{k-m:k})
\end{aligned}$$

[2] For reasons of clarity, the link between $\boldsymbol{m}$ and the user plan $\boldsymbol{i}_k$ is not shown in the graphical model of Fig. 5.

[3] Since the robot state $\boldsymbol{x}_k$ is encoded in our user plan representation $\boldsymbol{i}_k$, $\boldsymbol{x}_k$ does not explicitly appear in (2).

$$= \eta \cdot p_{user}(\boldsymbol{u}_k|\boldsymbol{i}_{k-m+1:k}, \boldsymbol{u}_{k-m+1:k-1}) \cdot \sum_{\boldsymbol{i}_{k-m}} \big(p_{process}(\boldsymbol{i}_k|\boldsymbol{i}_{k-m:k-1}) \cdot p_{k-1}(\boldsymbol{i}_{k-m:k-1}|\boldsymbol{u}_{k-m:k-1})\big). \quad (3)$$

This is also formulated and executed as a prediction and correction step. The *prediction* or *time update* step corresponds to the calculation of the sum in (3). This step predicts the user plans and their probability after a robot action has been executed. The *correction* step takes the user signals into account based on the user model. The new state becomes $\boldsymbol{i}_{k-m+1:k}$.

The power of Bayes rule to estimate stochastic variables stems from the fact that it tackles the estimation problem 'the other way around'. For example, it may be hard to directly estimate how probable it is that a user wants to dock at a table from an arbitrary starting pose (i.e. to directly estimate p_k), but estimating which signals a user will give (i.e. p_{user}) assuming that she wants to follow a given trajectory to the table seems much easier.

The probability distribution over user plans can then be used to take decisions under uncertainty to aid the user with the execution of a desired maneuver. In the presented approach, these shared control decisions only require knowledge of $p(\boldsymbol{i}_k|\boldsymbol{u}_{k-m+1:k})$, which is obtained by marginalizing over $\boldsymbol{i}_{k-m+1:k-1}$:

$$p(\boldsymbol{i}_k|\boldsymbol{u}_{k-m+1:k}) = \sum_{\boldsymbol{i}_{k-m+1:k-1}} p(\boldsymbol{i}_{k-m+1:k}|\boldsymbol{u}_{k-m+1:k}). \quad (4)$$

The following sections will discuss the user model p_{user} and the plan process function $p_{process}$. Also, further assumptions will be discussed. The user model p_{user} is the cornerstone of this plan recognition framework as it allows to make plan recognition adaptive to the user. The accuracy and usefulness of the framework depend to a large degree on the predictive power of this user model.

2.4 User model p_{user} of wheelchair drivers

In order to update the probability distribution using Bayes rule, it should be determined how likely the user signals are given that the user has a certain intent. In this paper, the wheelchair driver is modeled as a path tracker. Determining the likelihood of user signals then corresponds to determining how likely the user signals are for controlling the wheelchair such that it follows a certain trajectory. We adopt different user models for different user interface types.

Continuous user interfaces. The following user model is adopted for continuous interfaces similar to hand joysticks. We assume a time window of $m = 1$. Furthermore, we hypothesize that the perceptual cues for wheelchair drivers

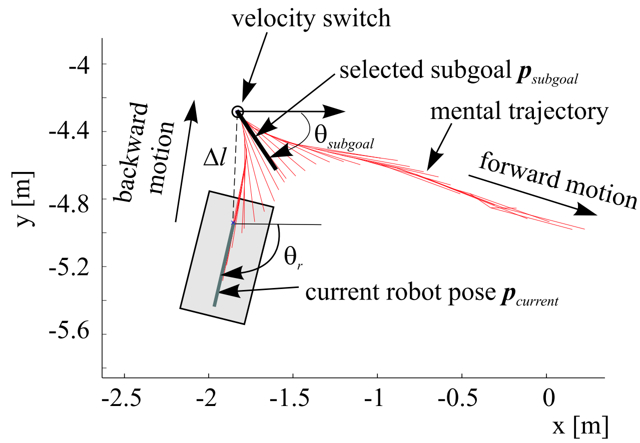

Fig. 6 Position of the subgoal in a trajectory in which a velocity switch occurs. A velocity switch is referred to here as a pose at which the robot's linear velocity changes sign. Δl is the distance from the current robot pose to the subgoal pose

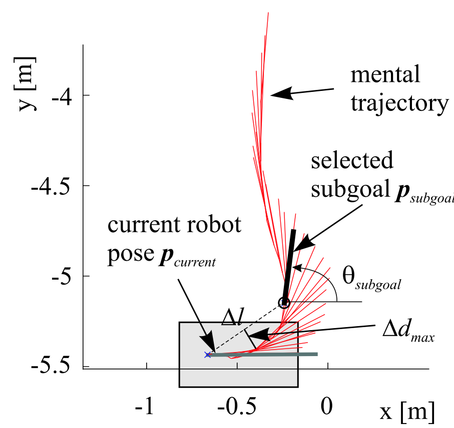

Fig. 7 Position of the subgoal in a trajectory in which no velocity switch occurs. Δl is the distance from the current robot pose to the subgoal pose

stem from subgoals that are lying on the mental path that the user is tracking. Path tracking errors are specified in terms of differences between the current robot pose and the subgoal pose. The parameters of the tracking controller are then estimated through least-squares regression.

We propose to put the subgoal at the pose where a velocity switch occurs, as shown in Fig. 6. A velocity switch is referred to as a pose on the path where the robot's linear velocity changes sign, i.e. where the robot changes from backward motion to forward motion or vice versa. If no velocity switch occurs in the trajectory, the farthest pose $\boldsymbol{p}_{subgoal}$ on the trajectory is considered to be the perceived subgoal, for which the distance of any point on the trajectory in between the current robot pose and $\boldsymbol{p}_{subgoal}$ to the straight line between the current robot pose and $\boldsymbol{p}_{subgoal}$ is smaller than a threshold Δd_{max} (see Fig. 7). Figure 4 illustrates the position of the subgoals for various plan hypotheses in a house-like environment.

From these subgoals, the regressors $\Delta\theta$ and Δl_{cor} are determined:

$$\Delta\theta = \theta_{subgoal} - \theta_r + \omega_r \cdot \widehat{T}_{sys}, \tag{5}$$

$$\Delta l_{cor} = \Delta l \cdot \left(\frac{1+\cos\Delta\theta}{2}\right)^{\alpha} + v_r \cdot \widehat{T}_{sys}, \tag{6}$$

where $\boldsymbol{t_r}(v_r, \omega_r)$ is the robot's twist, $\widehat{T}_{sys}$ is an estimate of the system delay computed as the peak of the correlation between user input and robot speeds, and Δl is the distance to the subgoal. This distance Δl is weighted with a bell-shaped function in (6), which models the observed behavior that wheelchair drivers tend to turn first until they are more or less aligned with the subgoal before they drive forward.

These regressors are then used to predict the user's linear and rotational joystick signals $\boldsymbol{u}_k(v_u, \omega_u)$ using the following regression model:

$$\omega_u^p = a_1 \cdot \Delta\theta + \varepsilon_\omega, \tag{7}$$

$$v_u^p = b_1 \cdot \Delta l_{cor} + \varepsilon_v, \tag{8}$$

$$\varepsilon_\omega \sim \mathcal{N}(0, \sigma_\omega), \quad \varepsilon_v \sim \mathcal{N}(0, \sigma_v). \tag{9}$$

The likelihood of a given user signal $\boldsymbol{u}_k(v_u, \omega_u)$ is then calculated from the predicted user signal $\boldsymbol{u}_k^p(v_u^p, \omega_u^p)$ as follows:

$$p_{user}(\boldsymbol{u}_k|\boldsymbol{i}_k) \sim \exp\left(-\frac{(v_u - v_u^p)^2}{2\sigma_v^2}\right) \cdot \exp\left(-\frac{(\omega_u - \omega_u^p)^2}{2\sigma_\omega^2}\right). \tag{10}$$

Discrete user interfaces. For the shared control experiments in Sect. 4.4 a discrete interface was adopted consisting of nine buttons as shown in Fig. 15(b). The corresponding wheelchair actions consist of a translation in the nine directions. The procedure to determine the likelihood of discrete user signals is similar to the procedure for continuous interfaces. Since the user is now controlling an omnidirectional robot without orientation, the user's mental path is represented as a set of two-dimensional (2D) $[x\ y]^T$ coordinates. The user model first extracts a subgoal from this 2D path, i.e. a position on the path that is still visible from the current robot position. This subgoal is assumed to be the perceptual cue for the user to issue certain interface signals. The subgoal is found by stepping from the robot position through the 2D path till a position where the straight line between the position on the 2D path and the robot position intersects with some object. Figure 8 shows the subgoal positions as stars, for a set of modeled mental paths.

The likelihood function is modeled based on what an optimal robot controller would give as output to track a path. For example, user signal *west* is assigned the likelihood $p_A = 1$ for all subgoal positions in area A shown in Fig. 9(b). In order to model human mistakes, a lower but still high likelihood $p_B = 0.1$ is assigned to area B. Subgoals in this area are not reached in an optimal way if the user issues signal west, but users were observed to make these mistakes. For subgoals lying in area C, user signal west is assigned a likelihood $p_C = 0.01$, and for subgoals in area D a likelihood $p_D = 0.001$. After having assigned a probability to each of the nine possible user signals for each intent $\boldsymbol{i}_k$, the likelihood function $p_{user}(\boldsymbol{u}_k|\boldsymbol{i}_k)$ is normalized.

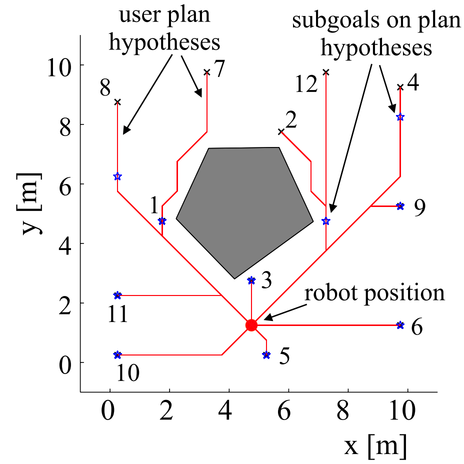

Fig. 8 This figure shows the subgoals for several mental paths to possible 2D global goal positions in a simulation environment

Since we would specifically like to illustrate our framework's ability to adapt to user-specific needs, we impose a handicap on the interface as follows. Whenever the user pushes one of the *east* buttons, the user signal is modified to the signal corresponding to the button left of it. Hence, signal *a* is transformed to signal *b*, signal *c* to signal *d*, and signal *e* to signal *f* in Fig. 15(b). This results in a user who cannot steer to the right by herself. The calibration of this user model is challenging, since the user is not capable to reach goal positions that lie east of the robot by herself. Therefore, shared control should be activated while gathering data for user model calibration. However, in order to take (optimal) decisions, the shared control algorithm needs the user model. This chicken-and-egg problem may be complicated even further if the user's plan is unknown. If the user's driving characteristics vary considerably over short periods of time, the characteristics may have to be estimated continuously on-line, under unknown user plans. Online calibration of user models was not the main objective of this work. Therefore, the adopted user model for the handicap in this work was constructed based on a priori knowledge of the actual handicap, and starting from the able-bodied user model in Fig. 9. More specifically, the likelihoods of the *east* buttons are calculated and then added to the buttons left of

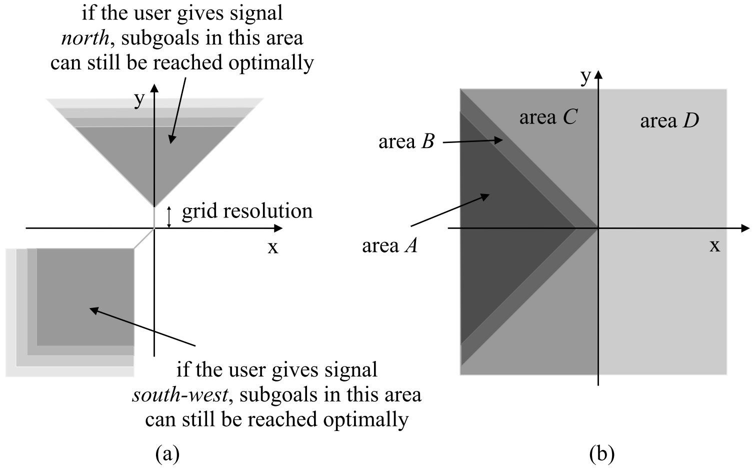

Fig. 9 Figure (**a**) indicates *two grey areas* in which subgoals lie that can be reached optimally (i.e. following a shortest path), given that the next user signal equals *south-west* for one area and *north* for the other area. The *fading shades* indicate that the *grey areas* actually extend till infinity in the direction suggested. Figure (**b**) discerns different areas in which subgoal positions may lie, where subgoals in different areas receive a different likelihood. This concrete division of areas determines the likelihood of user signal *west*. *Dark areas* correspond to subgoal positions that are likely if a user signal *west* is given. For example, subgoal positions in area D are the least likely if signal *west* is given. All subgoals in this area receive the same low likelihood p_D

it:

$$
\begin{aligned}
p_{user}(\boldsymbol{u}_{hand,k}|\boldsymbol{i}_k) &= \sum_{\boldsymbol{u}_k} p(\boldsymbol{u}_{hand,k}, \boldsymbol{u}_k|\boldsymbol{i}_k) \\
&= \sum_{\boldsymbol{u}_k} p(\boldsymbol{u}_{hand,k}|\boldsymbol{u}_k, \boldsymbol{i}_k) \cdot p(\boldsymbol{u}_k|\boldsymbol{i}_k) \\
&= \sum_{\boldsymbol{u}_k} p(\boldsymbol{u}_{hand,k}|\boldsymbol{u}_k) \cdot p(\boldsymbol{u}_k|\boldsymbol{i}_k) \qquad (11)
\end{aligned}
$$

where $\boldsymbol{u}_{hand,k}$ denotes the handicapped user signal, and $\boldsymbol{u}_k$ the intended user signal similar to what an able-bodied person would give. Equation (11) also provides the rationale behind a two-step approach of modeling handicaps, where the first step consists of constructing the user model of an able-bodied user $p(\boldsymbol{u}_k|\boldsymbol{i}_k)$, and the second step consists in modeling the physical handicap $p(\boldsymbol{u}_{hand,k}|\boldsymbol{u}_k)$. We have applied this approach to continuous user interfaces as well, see (Hüntemann et al. 2007).

2.5 Plan process function $p_{process}$

The plan process function $p_{process}$ predicts the shape and the probability of a user plan at time k, given previous user plan hypotheses in time window m. We hypothesize that users almost instantaneously adapt their mental trajectory as the wheelchair moves, and that the mental trajectory at time k only depends on the mental trajectory at time $k-1$, i.e. time window $m = 1$. In order to obtain computational efficiency, every mental trajectory at time $k-1$ is transformed into at most one mental trajectory at time k, and the probability of the trajectory at time $k-1$ is transferred entirely to its corresponding trajectory at time k. This is illustrated in Fig. 10. In practice, paths $\boldsymbol{i}_k$ to desired goal poses are recalculated at time k if the wheelchair's pose changed since time $k-1$, and the (prior) probabilities assigned to these new paths $\boldsymbol{i}_k$ correspond to the (posterior) probabilities assigned to paths $\boldsymbol{i}_{k-1}$ at time $k-1$ that were planned to the same goal poses. In order to allow for fast reaction to possible changes in user plan, noise is injected after prediction by imposing a minimum probability for each intent hypothesis, thereby increasing the entropy of the probability distribution.

2.6 Complete plan recognition procedure

The complete plan recognition algorithm proceeds as follows. The update frequency of Bayes rule in (2) is determined by the user model. For continuously sampled interfaces such as hand joysticks, this frequency is chosen to be twice the frequency of the user's control bandwidth (estimated to be around 2 Hz), in accordance with Nyquist's theorem. Each time the probability distribution should be updated, a new set of mental trajectories is predicted first using $p_{process}$, i.e. the prediction step in (3) is computed at the same frequency as the correction step. Then, the observed user signals are adopted to update the probability distribution with the user model and Bayes rule.

3 User-adapted shared control

As depicted in Fig. 2, the probability distribution over user intents is adopted to make assistive decisions under uncer-

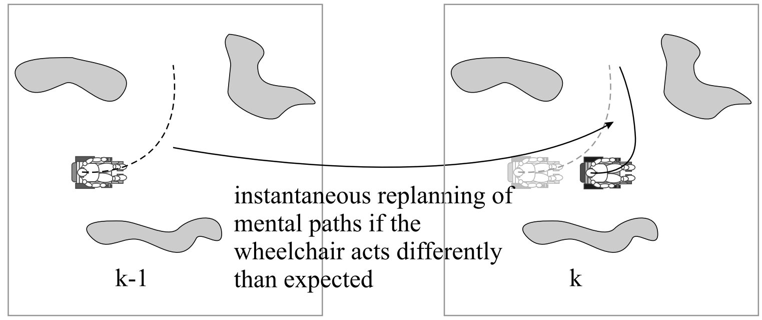

Fig. 10 This figure illustrates the hypothesis that while moving, the driver almost instantaneously replans the path to the desired goal pose. In our framework, each mental trajectory at time $k-1$ is replanned to at most one mental trajectory at time k, and the probability is completely transferred

tainty by the shared control algorithm. The design of this shared control system corresponds to the design of the controlled element in a closed-loop system, since the human driver can be considered to be the controller that has a reference trajectory in mind. In contrast, traditional control theory usually designs the controller instead of the controlled element. The design of a shared control system is furthermore complicated compared to that of classical controllers, for the following reasons:

– Much inherent, possibly multi-modal uncertainty is present regarding the state of the system, where the state corresponds in this case to the user's concrete maneuvering plan.
– Due to the human's strongly non-linear and adaptive behavior, it may not be sufficient to fit a model through the human's behavior while she controls a system fully by herself, and use that model to design the shared control system. Instead, models of human behavior depend heavily on the behavior of the controlled element. This chicken-and-egg design problem advocates assigning adaptive properties to the controlled element. However, one of the often stated requirements to avoid mode confusion[4] is that the system to be controlled should be deterministic. These requirements are contradicting and it is unclear at this moment how robots should learn and evolve such that stable and intuitive human-machine interaction is obtained at all times.
– It is not clear when users actually feel assisted. Furthermore, if we would like to design an 'optimal' shared controller, which criterion should be optimized? Can this criterion be learned?

Previous approaches to intelligent wheelchair control usually choose the most likely driving assistance mode based on the latest user signal and sensor information only, resulting in Maximum Likelihood (ML) decisions. One step further would be to use the assistance mode that is most probable after having taken previously inferred information into account, resulting in so-called Maximum A Posteriori (MAP) estimates. This corresponds to using the maximum of p_k instead of the maximum of p_{user} in Bayes rule, cf. (2). An even further step would be to use Partially Observable Markov Decision Processes (POMDPs) and to consider besides the a posteriori probabilities additionally the effects of possible actions. This approach chooses those actions that maximize the expected reward. This is the approach that we propose to adopt, since just executing the intent path that is most likely or most probable may result in actions that thwart the user's actual plans considerably, because various equally probable intents may have been estimated. Therefore, besides considering the posterior probabilities, the shared control framework should additionally account for effects of possible actions. Figure 11 illustrates the difference between various shared control approaches.

POMDPs provide a powerful decision-theoretic framework for planning under uncertainty. In this paper, the POMDP formulation of Kaelbling et al. (1998) is adopted. At each time step k, the human and robot are in an unknown discrete state $\boldsymbol{x}_k^{\text{POMDP}} \in \mathcal{X}$. The robot can take a discrete action $\boldsymbol{a}_k \in \mathcal{A}$, after which the human and robot arrive in a new state $\boldsymbol{x}_{k+1}^{\text{POMDP}} \in \mathcal{X}$ with probability $p(\boldsymbol{x}_{k+1}^{\text{POMDP}} \mid \boldsymbol{a}_k, \boldsymbol{x}_k^{\text{POMDP}})$. In this new state, the robot perceives a discrete observation $\boldsymbol{u}_{k+1} \in \mathcal{U}$ with probability $p(\boldsymbol{u}_{k+1} \mid \boldsymbol{x}_{k+1}^{\text{POMDP}}, \boldsymbol{a}_k)$.

A basic assumption in POMDPs is that the state of the agent is not known, but can only be partially perceived through observations. The uncertainty over the state is represented using a probability distribution b. The challenge is to pick a good sequence of actions, despite this state uncertainty. In order to find an optimal sequence of actions, a reward function $r_k\ (\boldsymbol{x}_k^{\text{POMDP}}, \boldsymbol{a}_k)$ is adopted that quantifies the cost or usefulness of its arguments.

The solution of a POMDP is a policy π, a function that maps each possible belief state to an action: $\pi(b) \to \boldsymbol{a}$. Exact and approximate techniques for solving POMDPs have been proposed in literature, see e.g. (Kaelbling et al. 1998; Cassandra 1998; Hauskrecht 2000). In general, available POMDP techniques discretize either the action or the state

[4]Humans are believed to maintain mental models of the systems they control. If the system's true state or *mode* differs from the system mode as estimated by the user, and if the system consequently acts differently than predicted by the user, mode confusion arises.

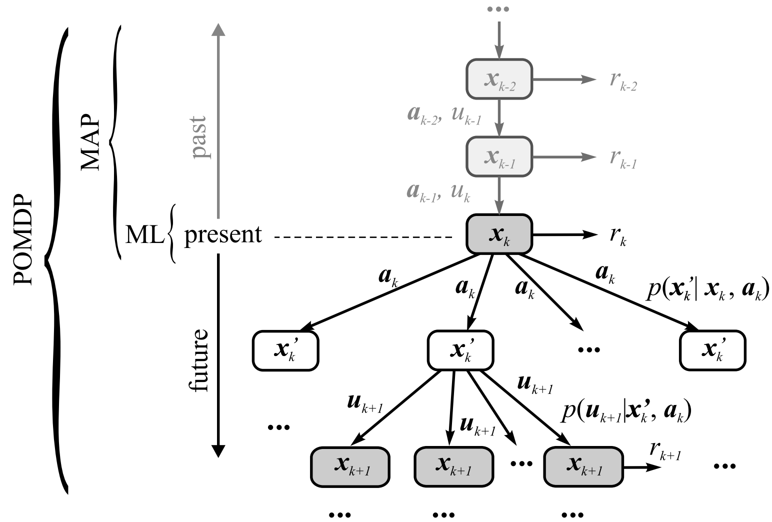

Fig. 11 The displayed decision tree illustrates the difference between Maximum Likelihood (ML), Maximum A Posteriori (MAP) and Partially Observable Markov Decision Processes (POMDP) approaches to shared control. ML takes shared control actions based on the latest sensor and user signals only, MAP additionally takes previously estimated user plans into account to take actions, and POMDP shared control additionally looks into the future and evaluates the effects of actions prior to choosing an action

space since finding exact solutions for continuous actions and states is intractable. Recently though, some approximate techniques have tried to cope with either continuous actions or states (Spaan and Vlassis 2005; Brooks et al. 2006; Thrun 2000).

In order to limit the computational complexity, we adopt a *greedy* POMDP approach, i.e. we look just one time step into the future and take that action that maximizes the immediate next payoff:

$$\pi_1(b) = \arg\max_{\boldsymbol{a}} \sum_{\boldsymbol{x}_k^{\text{POMDP}}} r_k(\boldsymbol{x}_k^{\text{POMDP}}, \boldsymbol{a}_k) b(\boldsymbol{x}_k^{\text{POMDP}}). \tag{12}$$

Observations in our POMDP model correspond to user signals $\boldsymbol{u}_k$. In our current implementation, the POMDP state $\boldsymbol{x}_k^{\text{POMDP}}$ is chosen to equal the user plan hypothesis, i.e. a trajectory to a possible goal state, such that $\boldsymbol{x}_k^{\text{POMDP}} = \boldsymbol{i}_k$. We furthermore assume that the robot's pose is completely observed after having taken an action. This proved to be a justified assumption during the performed experiments. The observation function $p(\boldsymbol{u}_{k+1}|\boldsymbol{x}'_k, \boldsymbol{a}_k)$ corresponds to the user model p_{user} in Bayes rule. Consequently, decisions are tailored to the user's specific needs, resulting in user-adapted shared control. Actions $\boldsymbol{a}_k$ the robot can take and the shape of the reward function $r_k(\boldsymbol{i}_k, \boldsymbol{a}_k)$ depend on the application and will be discussed in Sect. 4.4. Table 1 shows the mapping between the POMDP formulation and the shared control application.

4 Experimental results

4.1 Wheelchair platform

The plan recognition framework discussed above has been implemented on the wheelchair Sharioto, which is depicted in Fig. 1. This is a standard powered wheelchair that is differentially driven. Joystick and motors communicate via a Controller Area Network (CAN) bus. During a multi-disciplinary research project, novel sensors were developed (MLR homepage 2007), comprising one lidar (LIght Detection And Ranging) sensor, and 4 ultrasound sensors without a dead zone. The platform is furthermore equipped with 16 commercially available Polaroid ultrasound sensors, 9 Sharp infrared sensors, and one Gyrostar gyroscope for rotational velocity estimation and rotational velocity feedback. All programs for sensor and platform control are multi-threaded and object-oriented, written in C++. A laptop with Pentium 2.13 GHz processor and 2 GB RAM reads the sensors using two National Instruments DAQ-700 cards and connects to both the joystick and the motors via a CAN-bus driver. Velocity commands from the user interface are redirected via the laptop, corrected, and sent to the motors.

Table 1 Mapping from the POMDP formulation to the shared control problem. The right column depicts the number of states, observations and actions that were present in the shared control experiment of Sect. 4.4

POMDP	Shared control	Number
States $\boldsymbol{x}^{\text{POMDP}}$	User plans $\boldsymbol{i}$	12
Observations $\boldsymbol{z}^{\text{POMDP}}$	User signals $\boldsymbol{u}$	9
Actions $\boldsymbol{a}^{\text{POMDP}}$	Robot commands $\boldsymbol{a}$	9

4.2 User modeling results

In order to find the regression parameters a_1, b_1, σ_ω, and σ_v in (7) and (8) for a specific user, the user was asked to drive around in a test environment, and to stop at goal poses that she desired to reach, such as at a table. In order to record the followed trajectory, a map of the environment was first built using a feature-based simultaneous localization and

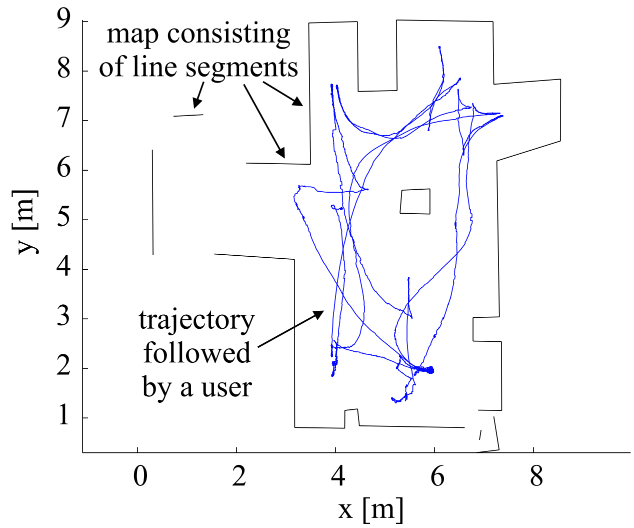

Fig. 12 Representation of the map that was built using a feature-based SLAM algorithm. The adopted features are line segments. Also shown is a typical trajectory executed by a wheelchair driver in this environment

mapping (SLAM) algorithm based on single-level relaxation (Frese et al. 2005). Then, for real-time experiments, local localization in the estimated map is performed using an improved version of the algorithm presented by Cox (1989). This algorithm matches laser scans in a least-squares way to segment features in the map. Our modifications are similar to those of Gutmann (2000). Figure 12 shows the estimated map and a typical trajectory executed by a user. Test runs last for about 10 minutes. This section reports about experiments with able-bodied users.

Goal poses in the recorded trajectories are automatically extracted based on the time periods during which users stand still at certain places. For each robot pose along the segmented trajectories, the subgoals and the corresponding regressors $\Delta\theta$ and Δl_{cor} are calculated as proposed in Sect. 2.4. Figures 13(a) and (b) show the data points and the regression. Though the variance on the data seems considerable, the linear model appears to be a reasonable choice. The negative slope for the $(\omega_u, \Delta\theta)$ data stems from the fact that the rotational joystick units as adopted on this wheelchair type (i.e. NV, from *Network Variable*) are negative where the SI units for rotational velocity (i.e. rad/s) are positive. Both least-squares and robust regression were adopted to estimate a_1, b_1, σ_ω, and σ_v.

4.3 Plan recognition results

The plan recognition framework presented in Sect. 2.3 was evaluated both in simulation and on the wheelchair Sharioto. In order to show the applicability of the framework to different user interface types, simulation experiments were conducted with a discrete, switch-based interface, whereas experiments on the wheelchair platform were conducted with a continuous joystick interface. The simulation results are described in Sect. 4.4 together with experiments on shared control.

The performance of the plan recognition framework has been evaluated in a house-like environment for which global goal poses were determined by learning the poses at which users stand still for a minimum amount of time. Furthermore, some additional goal poses were determined by manual indication on the estimated map.

It was checked whether user plan estimates converged to the true user plans. Knowledge about true user plans was obtained after execution of a maneuver. Figure 14 shows a typical evolution of the probability distribution during a parking maneuver performed with Sharioto. The user starts from a pose that is more or less perpendicular to a wall, and ends parallel to the wall. In order to do so, the user first turns over about 190° to the left (time steps 1100 to 1112), and then drives backward while turning over about 90° to the right (time steps 1112 to 1128). The convergence of the probability function towards the true user plans shows that even complex user plans such as this one can be estimated correctly, as long as the user's mental path is predicted accurately. User models that only look at the direction the user points to, are unlikely to be able to infer from these signals the user's true plan, and may consequently take incorrect assistive actions for those maneuvers. Furthermore, for this particular maneuver, the probability function converged quickly to a uni-modal distribution.

Twenty one different user plan hypotheses were taken into account simultaneously. As shown in Fig. 14, the probability function may at times be multi-modal. If these different hypotheses would require different correcting actions, a proper decision should be taken. Being aware of various hypotheses that are simultaneously probable allows to prevent the system from taking premature actions. Furthermore, this allows *active sensing* or *active acting*, i.e. executing specific actions that are expected to disambiguate between equally probable plan hypotheses.

4.4 Shared control results

The POMDP shared control framework is evaluated in a simulated environment, in which an omni-directional robot is controlled using a discrete interface. Consider the situation depicted in Fig. 15(a). Suppose we would like to move a robot to position 3, stay there for a while, and then move to position 9. Figure 15(a) shows some paths a user may have in mind to perform these maneuvers. Remark that there are various equally efficient ways to reach position 9 from position 3, e.g. path B or C. This corresponds to the local uncertainty depicted in Fig. 3(a). In this specific experiment, the user has path A-B in mind. As explained in Sect. 2.4, a virtual handicap is imposed such that the user

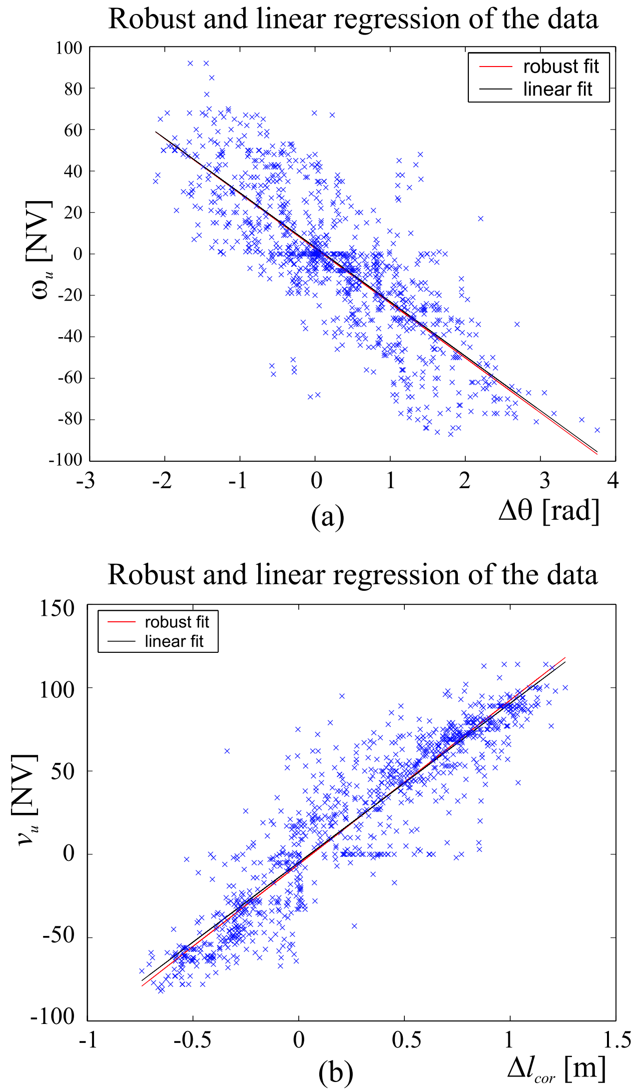

Fig. 13 Representation of the rotational user signal ω_u (*up*) and linear user signal v_u (*down*) as a function of the angle difference $\Delta\theta$ and weighted distance Δl_{cor} respectively. Joystick units in this figure are expressed in hardware-specific units (NV), which range from -127 to 127

is not able to steer to the right by herself. It is up to the shared control framework to use the inferred intents to compensate for the capabilities the user lacks. Nine actions $\boldsymbol{a}_k$ are available to the shared controller, corresponding to all actions in Fig. 15(b). Intents are again represented as paths to goal positions. Handicaps like these occur for example in brain-computer interfaces, where users may experience difficulties to generate a specific EEG signal corresponding to one of the discrete steering classes. Furthermore, we actually observed wheelchair drivers in our user group who exhibit similar handicaps with continuous interfaces. The experiment is chosen for its simplicity, which allows easy analysis.

In order to obtain safe and intuitive behavior, the reward function r_k for these experiments consists of three terms. The first assigns a large cost to an action if the resulting robot pose is obstructed by an obstacle. The second term assigns a large cost to a combination of a machine action and an intent hypothesis if the action corresponds poorly to the intent hypothesis. The third cost term assigns a large cost to a combination of a machine action and the user signal if the machine action corresponds poorly to the user signal. This cost term makes sure that if there is some freedom in executing a path, the path will be executed as closely as possible to the user's signals. These cost terms are weighted in the POMDP model with the user model p_{user}, which predicts the likelihood that the user will actually give that signal.

The experiments show that with this user plan representation, user model, and reward function, a user who cannot go to the right is able to drive to the right. The main prerequisite for achieving this is that the user model p_{user} is aware of the user's physical limitations. The probability function over intents in Fig. 15(c) shows that at various places the probability function is indeed multi-modal. Furthermore, as can be seen in the snapshots of Fig. 15(d), the robot executes

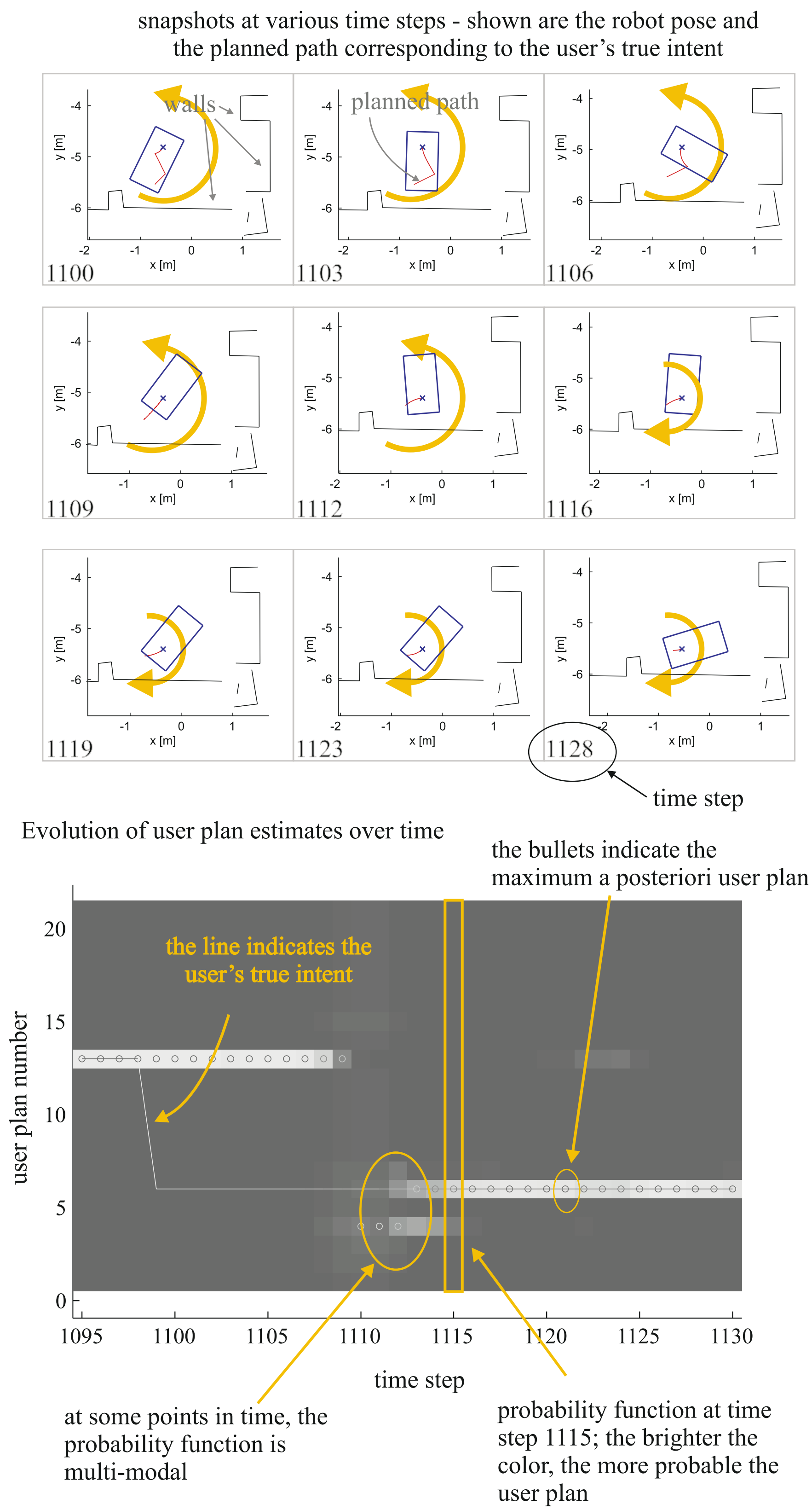

Fig. 14 The snapshots in the *top figure* show a complex parking maneuver, with the intent to position the wheelchair parallel to the wall. Also shown is the path planned by the plan recognition module, corresponding to the true intent. The corresponding evolution of the probability function over user plans is shown at the bottom, together with the true evolution of user plans

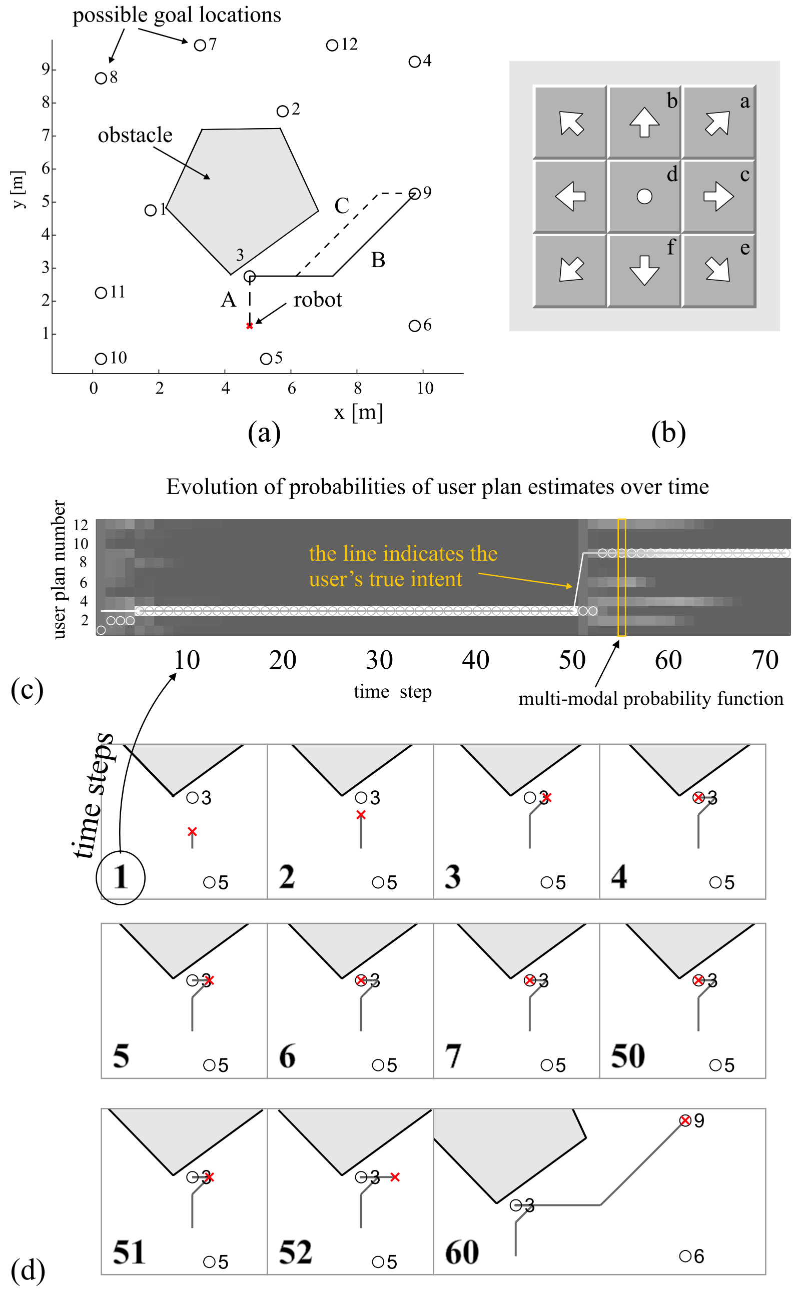

Fig. 15 Figure (**a**) shows an environment with a number of possible goal positions indicated by circles. The *cross* in the figure represents the position of a robot that the user tries to control. This figure also shows the path the user has in mind, which consists in first going to position 3 (path A), and then to position 9 from position 3 (path B). Figure (**b**) shows the discrete interface that is used to control the robot. Figure (**c**) shows the evolution of the probability function p_k over time. The *line* denotes the user's true intent, and the *white circles* show the maxima of the probability function at each time step. Figure (**d**) shows the behavior of the human-machine system as snap-shots at different time steps. Position 3 is reached at time step 4. The user stays at this position from time step 6 till time step 50, and then moves on to position 9

the path the user has in mind (path A-B in Fig. 15(a)). Consequently, the maneuver is executed the way the user would like it to be executed.

5 Related work

5.1 Assistive robotics

The software architecture of nearly all existing robotic wheelchairs, robotic guide dogs for the visually impaired, and robotic walking assistants is comprised of several *modes*, sometimes called *behaviors* or *agents*. Among these modes, the *user-control mode* is the most common one. In principle, plan recognition is not required in user-control mode, but it may be useful for suggesting assistive actions. For example, traditional scanning interfaces for wheelchairs successively highlight one of a set of elementary directions the driver may want to take, such as *turn left* or *drive forward*. With a binary yes/no signal, the user can indicate if she actually wants the wheelchair to follow the direction that is highlighted at that time instant. If these commands were augmented with a more sophisticated task that the user is estimated to desire to execute, the chair's functionality would be enhanced without requiring additional physical capabilities of the user. This may considerably reduce the travel time and increase safety. Besides a user-control mode, an *autonomous* mode is often defined, in which user signals are completely disregarded and plan recognition amounts to an explicit indication by the user of the goal location that she would like to reach. Many assistive robots also offer several *semi-autonomous* modes that require continuous collaboration between robot and user. Each of these semi-autonomous modes encodes a separate user intent. Various projects adopt modes that are directly derived from the specification of tasks that a wheelchair driver typically wants to perform, such as *avoid obstacle*, *drive through door*, or *dock at table* (Simpson et al. 1998; Röfer and Lankenau 2000). Others specify modes based on environmental characteristics, such as *outdoor* versus *indoor* (Yanco 2000), *cluttered* versus *open*, or *dynamic* versus *static* (Prassler et al. 1999).

The presence of multiple modes requires a selection or combination procedure for these modes. This corresponds to the plan recognition problem. Some systems require the user to select the correct mode *manually* (Prassler et al. 1999; Parikh et al. 2004). However, this work focuses on robotic assistance that activates modes automatically, as these are hoped to be less tiring and more intuitive than manual mode selection procedures. Various approaches exist for this *automatic* mode selection. One option is to merge the motion commands given by several assistance modes, though this is a less frequent approach (Aigner and McCarragher 2000). Another option is to select a single mode to be active, resulting in a *winner-takes-all* strategy, which is the prevailing strategy. Most approaches can be considered to determine how *likely* or *suitable* a certain assistance mode is at a certain time instant, after which the shared control algorithm selects the assistance mode that is most suitable. This suitability measure, though it is not always explicitly calculated, may be based on several possible information sources:

1. **Sensor signals**. Some suitability values of assistance modes are purely based on sensor information. These approaches disregard the user signals, and are therefore better classified as fully autonomous systems. Simpson et al. (1998) adopt a Bayesian network that uses evidence from the latest sensor readings and from an estimate of the robot location to calculate the probabilities of assistance modes *follow corridor*, *drive through door*, and *avoid obstacle*. Yanco (2000) adopts several sensors and a trained classifier to detect whether the wheelchair is outside or inside, and to trigger the corresponding outside/inside assistance mode. However, disregarding user signals may activate incorrect assistance modes. Consider for example a wheelchair user standing near a door in a corridor. Intuitively, a decision to drive through the doorway should stem from the user pointing more or less in the direction of the door, and if not, the user may be assumed to desire to follow the corridor. If only sensor signals are used however, the system would e.g. always opt to drive through the doorway independently of the user signals, strongly reducing the user's freedom in this way.
2. **Sensor and user signals**. Other automatic mode selection techniques base their decisions both upon environmental perception and user signals. Though these approaches adhere more to the user's expectations as they consider user signals to activate certain assistance modes, their user model is usually fixed and hard-coded, i.e. the same decision rules are adopted for all drivers. Examples of these approaches can be found in (Tzafestas 2001) and (Mittal et al. 1998).
3. **Sensor signals, user signals, and assistance actions**. Very few suitability values are calculated by additionally taking the outputs of assistance modes into account (Aigner and McCarragher 2000; Parikh et al. 2004). Aigner and McCarragher (2000) compare the user signals with the outcome of an avoid-obstacle assistance mode in order to decide which commands should be sent to the motors. In case of a strong conflict between these signals, the user obtains full control, but at reduced velocity.

Hence, an important difference with the framework proposed in this paper is that most existing approaches determine the most likely assistance mode using the latest sensor and user interface signals only, and uncertainty upon these signals is often disregarded. Moreover, due to the rather broad specification of assistance modes, a more accurate estimation of the user's plan has to be performed by these assistance modes once they are activated. For example, activating the obstacle-avoidance mode does not yet specify whether to avoid an obstacle to the left or to the right, and to which extent the obstacle should be avoided to the left or to the right. It is the responsibility of the assistance modes to infer these more concrete user plans. In contrast, this paper proposed a framework for plan recognition that can be tailored to the particular driving characteristics of the user. This framework explicitly deals with uncertainty present in user signals and user plans, and merges past beliefs regarding user plans with new evidence from user signals.

5.2 Plan recognition

In literature, approaches that are comparable to the probabilistic plan recognition framework of this paper can be found in the analysis of motion behaviors. Bennewitz et al. (2002) propose a framework to cluster and predict human motion based on past motion. Walking trajectories are classified into motion behaviors by employing Expectation Maximization (EM). The goal of the framework is to find the Maximum a Posteriori motion pattern given a trajectory in order to predict the motion of people. Glover et al. (2004)

estimate the activity a person is engaged in when using a walking aid. They adopt a Hidden Markov Model (HMM) that integrates metric, topological and temporal information into a probabilistic estimate over activities. In contrast to the framework proposed in this work, neither of these approaches directly includes user signals into predictions of motion behavior or motion plans, and they implicitly assume that users can go everywhere autonomously.

5.3 User modeling

Wheelchair driving is a typical example of manual control. *Manual control* is the situation in which a person receives through her senses (visual, tactile, vestibular, etc.) information regarding the desired state of some variables, and in which she manipulates a mechanical device (handles, knobs, joysticks, etc.) in order to minimize the perceived error (Sheridan and Ferrell 1981). Important milestones in the field of manual control were achieved in the 70s and early 80s, e.g. by McRuer (1980).

A manual control example that is closely related to wheelchair driving is car driving, for which several manual control models have been proposed in literature. Weir and McRuer (1970) use conventional control theory to model car drivers tracking other cars or lanes on a road with disturbances originating from e.g. wind gusts. Kleinman et al. (1970) use optimal control theory to characterize humans performing manual control. Other prototypes of intelligent vehicles using controllers reproducing human driving performance can be found in literature (e.g. Wang et al. 2002; Zheng et al. 2004).

Several differences exist between most manual control experiments and assistive wheelchair applications. First of all, many manual control experiments consist of *compensatory* tasks, for which the human operator only perceives the momentary error to be minimized. In practice however, wheelchair drivers perform *precognitive*, *preview* tasks, i.e. they usually have some a priori knowledge regarding the system and they observe not only the instantaneous path tracking errors, but also the reference path they want to track. Secondly, in contrast to car driving, wheelchair drivers usually do not follow a wheelchair in front of them, nor do they have to drive inside lanes when driving indoors. Due to the fact that the user plan representation proposed in this paper explicitly models the mental trajectory the user tries to follow, the 'lane' that should be tracked is automatically provided in our framework. Consequently, models of car drivers may be adapted to robotic wheelchair driving.

6 Conclusion

This paper presented a novel framework for user-adapted plan recognition and shared control. One of the main innovations with respect to previous approaches is the estimation of the user's intent in order to provide adapted driving assistance. Adaptation means here that the assistance is tailored to the user's driving skills. In order to achieve this, intents are modeled as a goal pose and goal twist together with a trajectory to achieve the goal pose and twist. The user is modeled as a path-tracking controller that issues uncertain control signals. Intent paths are calculated in the framework with a fine-motion planner that takes the geometry and kinematic constraints of the robotic platform into account such that the framework can be adapted to different wheelchair types. Additionally, the proposed framework considers the uncertainty on the user's intent in order to cope with the human's inherently stochastic driving performance. This modeling of uncertainty allows in its turn to take informed assistance decisions. We have applied a greedy POMDP approach to take such informed actions for assisting the user. These actions can be tailored to the user's driving skills since the same user model as adopted for plan recognition is employed for making decisions under uncertainty.

Experiments have been conducted to demonstrate the feasibility of the novel framework both in simulation and on a real robotic wheelchair. Plan recognition has been validated with a continuous interface on the real platform, whereas results on shared control have been obtained with a discrete interface in simulation. Future work will extend the simulation results of the POMDP model to our robotic wheelchair.

Although we have selected wheelchair driving assistance as a case study of human-machine interaction, we believe that the results are potentially applicable to other domains such as collaborative human-robot manipulation. In those applications, a similar trajectory planning technique can be employed to determine trajectories to be followed by the robotic manipulator, which can be considered as possible intents of the user.

Future work will focus on on-line learning of the user model and the reward function, since the driver's behavior may change over time. Inspiration may be gained from other fields such as dialog management (Doshi and Roy 2007) and wheelchair navigation (Jaulmes et al. 2007). Furthermore, recently developed POMDP techniques will be applied in order to plan more than one time step ahead.

Acknowledgements This research has been carried out with the financial help of the EU STREP IST Movement project, the EU STREP IST Maia project and the FWO project G.0317.05. This research is also funded by a Ph.D. grant of the Institute for the Promotion of Innovation through Science and Technology in Flanders (IWT-Vlaanderen). A. Hüntemann is research assistant of the research foundation Flanders FWO.

References

Aigner, P., & McCarragher, B. J. (2000). Modeling and constraining human interactions in shared control utilizing a discrete event framework. *IEEE Transactions on Systems, Man, and Cybernetics—Part A: Systems and Humans*, *30*(3), 369–379.

Bennewitz, M., Burgard, W., & Thrun, S. (2002). Using EM to learn motion behaviors of persons with mobile robots. In *Proceedings of the IEEE/RSJ international conference on intelligent robots and systems* (pp. 502–507). EPFL, Lausanne, Switzerland.

Brooks, A., Makarenko, A., Williams, S., & Durrant-Whyte, H. (2006). Parametric POMDPs for planning in continuous state spaces. *Robotics and Autonomous Systems*, *54*(11), 887–897.

Carberry, S. (2001). Techniques for plan recognition. *User Modeling and User-Adapted Interaction*, *11*(1–2), 31–48.

Cassandra, A. R. (1998). *Exact and approximate algorithms for partially observable Markov decision processes*. Ph.D. thesis, Brown University.

Cooper, R. A., & Cooper, R. (2003). Trends and issues in wheeled mobility technologies. In E. Steinfeld, V. Paquet (Eds.), *International workshop on space requirements for wheeled mobility* (pp. 1–26). Buffalo, NY, USA.

Cox, I. J. (1989). Blanche: Position estimation for an autonomous robot vehicle. In *Proceedings of the IEEE/RSJ international workshop on intelligent robots and systems* (pp. 432–439). Tsukuba, Japan.

Demeester, E., Nuttin, M., Vanhooydonck, D., & Van Brussel, H. (2003). A model-based, probabilistic framework for plan recognition in shared wheelchair control: Experiments and evaluation. In *Proceedings of the 2003 IEEE/RSJ international conference on intelligent robots and systems* (pp. 1456–1461). Las Vegas, Nevada, USA.

Demeester, E., Nuttin, M., Vanhooydonck, D., Vanacker, G., & Van Brussel, H. (2005). Global dynamic window approach for holonomic and non-holonomic mobile robots with arbitrary cross-section. In *Proceedings of the 2005 IEEE/RSJ international conference on intelligent robots and systems* (pp. 2694–2699). Edmonton, Alberta, Canada.

Doshi, F., & Roy, N. (2007). Efficient model learning for dialog management. In *Proceedings of the ACM/IEEE international conference on human-robot interaction* (pp. 65–72). Arlington, Virginia, USA.

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

Frese, U., Larsson, P., & Duckett, T. (2005). A multilevel relaxation algorithm for simultaneous localization and mapping. *IEEE Transactions on Robotics*, *21*(2), 196–207.

Glover, J., Thrun, S., & Matthews, J. T. (2004). Learning user models of mobility-related activities through instrumented walking aids. In *Proceedings of the 2004 IEEE international conference on robotics & automation* (Vol. 4, pp. 3306–3312). New Orleans, LA.

Gutmann, J. S. (2000). *Robuste navigation autonomer mobiler systeme*. Ph.D. thesis, University of Freiburg.

Hauskrecht, M. (2000). Value-function approximations for partially observable Markov decision processes. *Journal of Artificial Intelligence Research*, *13*, 33–94.

Hüntemann, A., Demeester, E., Vanacker, G., Vanhooydonck, D., Philips, J., Van Brussel, H., & Nuttin, M. (2007). Bayesian plan recognition and shared control under uncertainty: Assisting wheelchair drivers by tracking fine motion paths. In *Proceedings of the 2007 IEEE/RSJ international conference on intelligent robots and systems* (pp. 3360–3366). San Diego, California, USA.

Jaulmes, R., Pineau, J., & Precup, D. (2007). A formal framework for robot learning and control under model uncertainty. In *Proceedings of the 2007 IEEE international conference on robotics and automation* (pp. 2104–2110). Roma, Italy.

Kaelbling, L. P., Littman, M. L., & Cassandra, A. R. (1998). Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, *101*(1–2), 99–134.

Kleinman, D. L., Baron, S., & Levison, W. H. (1970). An optimal control model of human response—part I: Theory and validation. *Automatica*, *6*, 357–369.

Kröse, B. J. A., Porta, J. M., van Breemen, A. J., Crucq, K., Nuttin, M., & Demeester, E. (2003). Lino, the user-interface robot. In *Lecture notes in computer science: Vol. 2875*. *Ambient intelligence* (pp. 264–274). New York: Springer. ISBN 3-540-20418-0.

Latombe, J. C. (1991). *Robot motion planning*. Dordrecht: Kluwer Academic. ISBN 0-7923-9129-2.

Martens, C., Ruchel, N., Lang, O., Ivlev, O., & Gräser, A. (2001). A FRIEND for assisting handicapped people. *IEEE Robotics and Automation Magazine*, *8*(1), 57–65.

McRuer, D. (1980). Human dynamics in man-machine systems. *Automatica*, *16*, 237–253.

Mittal, V. O., Yanco, H. A., Aronis, J., & Simpson, R. (Eds.). (1998). Assistive technology and artificial intelligence—applications in robotics, user interfaces and natural language processing. In *Lecture notes in computer science: Vol. 4158*. *Lecture notes in artificial intelligence*. New York: Springer.

MLR homepage. (2007). *Mobile learning robots homepage*. www.mech.kuleuven.be/mlr/.

Parikh, S. P., Grassi, V., Jr., Kumar, V. (2004). Incorporating user inputs in motion planning for a smart wheelchair. In *Proceedings of the 2004 IEEE international conference on robotics & automation* (Vol. 2, pp. 2043–2048). New Orleans, LA, USA.

Prassler, E., Scholz, J., & Fiorini, P. (1999). Navigating a robotic wheelchair in a railway station during rush hour. *The International Journal of Robotics Research*, *18*(7), 711–727.

Röfer, T., & Lankenau, A. (2000). Architecture and applications of the Bremen autonomous wheelchair. *Information Sciences*, *126*(1), 1–20.

Schmidt, C. F., Sridharan, N. S., & Goodson, J. L. (1978). The plan recognition problem: An intersection of psychology and artificial intelligence. *Artificial Intelligence*, *11*(1–2), 45–83.

Sheridan, T. B. (1992). *Telerobotics, automation, and human supervisory control*. Cambridge: The MIT Press.

Sheridan, T. B., & Ferrell, W. R. (1981). *Man-machine systems: Information, control, and decision models of human performance*. Cambridge: The MIT Press.

Simpson, R. C. (2005). Smart wheelchairs: A literature review. *Journal of Rehabilitation Research & Development*, *42*(4), 423–436.

Simpson, R. C., Levine, S. P., Bell, D. A., Jaros, L. A., Koren, Y., & Borenstein, J. (1998). *Lecture notes in computer science: Vol. 4158*. *Navchair: An assistive wheelchair navigation system with automatic adaptation* (pp. 235–255). New York: Springer.

Spaan, M. T. J., & Vlassis, N. (2005). Planning with continuous actions in partially observable environments. In *Proceedings of the 2005 IEEE international conference on robotics and automation* (pp. 3458–3463). Barcelona, Spain.

STWW project homepage. (2007). www.mech.kuleuven.be/stww/results/d1.1+.doc. Deliverable on user requirements and functional specifications.

Thrun, S. (1999). Monte Carlo POMDPs. In S., Solla, T., Leen, & K. R., Müller, *Advances in neural information processing systems* (Vol. 12, pp. 1064–1070). Cambridge: MIT Press.

Tsui, K. M., & Yanco, H. A. (2007). Simplifying wheelchair mounted robotic arm control with a visual interface. In *Proceedings of the AAAI spring symposium on multidisciplinary collaboration for socially assistive robotics*, Palo Alto, California, USA.

Tzafestas, S. G. (Ed.), (2001). Reinventing the wheelchair—autonomous robotic wheelchair projects in Europe improve mobility and safety. *IEEE Robotics and Automation Magazine*, *7*(1).

Wang, F. Y., Mirchandani, P. B., & Wang, Z. (2002). The VISTA project and its applications. *IEEE Intelligent Systems*, *17*(6), 72–75.

Weir, D. H., & McRuer, D. T. (1970). Dynamics of driver vehicle steering control. *Automatica*, *6*, 87–98.

Yanco, H. A. (2000). *Shared user-computer control of a robotic wheelchair system*. Ph.D. thesis, Massachusetts Institute of Technology MIT, Department of Electrical Engineering and Computer Science.

Yu, H., Spenko, M., & Dubowsky, S. (2003). An adaptive shared control system for an intelligent mobility aid for the elderly. *Autonomous Robots*, *15*(1), 53–66.

Zheng, N. N., Tang, S., Cheng, H., Li, Q., Lai, G., & Wang, F. Y. (2004). Toward intelligent driver-assistance and safety warning systems. *IEEE Intelligent Systems*, *19*(2), 8–11.

Eric Demeester received the M.Sc. degree in Mechanical Engineering in 1999 and the Ph.D. degree in applied sciences in 2007 from the Katholieke Universiteit Leuven. He is currently a postdoctoral researcher at the same university, where he is conducting research in the area of robotics for elderly and disabled people. His research interests include human-machine interaction, user modeling, plan recognition, shared control, and Bayesian inference and decision-making. He has taken part in various robotics projects and conferences. For his Ph.D. dissertation he was given the Robotdalen Scientific Award.

Alexander Hüntemann received his M.Sc. in Telecommunication Engineering from the Polytechnic University of Madrid in 2003. In the year 2004 he was a Marie Curie Fellow on the project DIMENSION at LMS, a world leading Belgian company in Noise and Vibration Engineering. Currently he is working as research assistant of F.W.O. (Research Foundation Flanders) on his Ph.D. at the Katholieke Universiteit Leuven, Belgium. His research interests include adaptive shared control, plan recognition and Bayesian estimation.

Dirk Vanhooydonck received the M.Sc. degree in Mechanical Engineering in 1999 and the Ph.D. in applied sciences in 2007, both from the Katholieke Universiteit Leuven. He is currently a postdoctoral researcher at the same university, where he is working in the area of assistive robotics for people with disabilities. His research interests include shared control, intention estimation, human-machine interaction, user modeling, and learning approaches.

Gerolf Vanacker received a degree in Engineering from GroepT hogeschool in Leuven in 2000 and a Master in Artificial Intelligence from the Katholieke Universiteit Leuven in 2003. He is currently pursuing a Ph.D. in applied sciences at the same university. His research interests include learning approaches, assistive robotics, human-machine interaction and user modeling.

Hendrik Van Brussel is full professor in mechatronics and automation at the Faculty of Engineering, Katholieke Universiteit Leuven (K.U.Leuven), Belgium, chairman of the Division of Production engineering, Machine design and Automation, Department of Mechanical Engineering. He received his B.Sc. ME (Technisch Ingenieur) degree from Hoger Technisch Instituut, Oostende, Belgium, in 1965, and his M.Sc. EE (Burgerlijk Ingenieur) and Ph.D. degrees from K.U.Leuven, Belgium, in 1968 and 1971 respectively. From 1971 until 1973 he was active in Bandung, Indonesia, establishing a Metal Industries Development Centre and as an associate professor at Institut Teknologi Bandung.
He was a pioneer in robotics research in Europe and an active promoter of the mechatronics idea as a new paradigm in machine design. He has published extensively on different aspects of robotics, mechatronics and flexible automation. His present research interest is also shifting towards holonic manufacturing systems, behavior-based robots, and micro and precision engineering, including microrobotics.
He is Fellow of SME and IEEE. He received a honorary doctor degree from Rheinisch-Westfälische Technische Hochschule (RWTH), Aachen, Germany, from 'Politehnica' University in Bucarest, Romania, and from 'Transilvania' University Brasov, Romania. In 2001–2002 he served as President of CIRP (International Institution for Production Engineering Research). He is a Member of the Royal Flemish Academy of Belgium for Sciences and Fine Arts, and Foreign Member of the Royal Swedish Academy of Engineering Sciences (IVA).

Marnix Nuttin focuses on learning approaches to robotics and manufacturing. He contributed to and coordinated several research projects on regional, federal and European level. He worked since 1992 on learning techniques for Programming by Demonstration. Regional and national projects include adaptive shared autonomy for wheelchair control, advanced mechatronic systems and multi-agent systems, autonomic computing for decentralized systems. He participated to the ITEA AMBIENCE project (context-aware environments for ambient services). In FP6 he is principle investigator of the projects MOVEMENT (Modular Versatile Mobility Enhancement Technology), MAIA (Non Invasive Brain Interaction with Robots) and DYNAVIS (Dynamically Reconfigurable Quality Control).