

Probabilistic approach to recognize local navigation plans by fusing past driving information with a personalized user model

Alexander Hüntemann, Eric Demeester, Emmanuel Vander Poorten,
Hendrik Van Brussel and Joris De Schutter
Department of Mechanical Engineering, KU Leuven,
Celestijnenlaan 300B, B-3001 Leuven, Belgium
Alexander.Huntemann@mech.kuleuven.be

Abstract—Navigating an electrical wheelchair can be very challenging due to its large size and limited maneuverability. Additionally, target users often suffer from cognitive or physical disabilities, which interfere with safe navigation. Therefore, a robotic wheelchair that helps to drive can prove invaluable. Such a wheelchair shares the control with its human operator. Typically, robots excel in fine-motion control whereas users want to remain in charge. Hence, the robot should focus its help locally and let the user decide about global behavior. Further, an effective robot should understand the navigation plans of its user. It needs to consider the user’s abilities to avoid frustrating the user with wrong assistance.

In order to address these requirements, we propose a probabilistic framework to recognize local navigation plans in a user-specific way. The framework infers navigation plans online and provides a method to calibrate all model parameters from real driving data. It fuses past local information with a user-specific model to reason about how and where the user intends to navigate. We illustrate the validity of our approach by recognizing the local navigation plans of a spastic user driving in a daily environment.

I. INTRODUCTION

Mobility enhancing technology, such as electrical wheelchairs, is helping disabled and elderly people regain their lost independence. Remaining independent clearly has a positive impact on a person’s well-being and self-esteem. Despite their benefits, electrical wheelchairs remain difficult to control. Dangerous accidents with the environment and other people occur frequently due to the large size of a typical wheelchair. Therefore, many potential users are forbidden to drive if they lack the necessary cognitive or physical skills. Even able users find it difficult and tiresome to navigate safely.

Improving wheeled mobility becomes possible by equipping an electric wheelchair with sensors and computing to assist during navigation. An important requirement of this assistance is to let the user feel in control. Actually, the robotic wheelchair shares the control with its operator. The user profits from the precise and safe movement of the robot, whereas the robot relies on the user for global planning and coarse control.

A potential safety risk exists, though, when human and robots collaborate. The robot might misunderstand its user and drive into a wrong direction. This can happen if the robot takes decisions without considering the user’s abilities and needs. In order to avoid frustrating its user, the robot must

recognize navigation plans in a way that considers his or her driving abilities. We model such plans as local trajectories that the robot should execute. Further, to increase robustness, the robot measures its confidence about the estimated plans and bases its actions on this knowledge. While reasoning about the probability of each plan, the robot considers the user’s abilities explicitly. We refer to this process as *plan recognition* or *intention estimation* interchangeably. 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.

A. Overview and contributions

This paper describes a probabilistic approach, which combines spatial and temporal user-specific information to infer local navigation plans. We first detail the process of recognizing local plans in section II. Starting with an example, we illustrate the complete procedure and highlight the challenges. Then, we translate the problem of plan recognition into a Dynamic Bayesian Network and explain our underlying assumptions. Section III describes the procedure to learn the plan recognition models from driving data. The experimental section, section IV, applies the framework to recognize the navigation plans of a spastic user driving a wheelchair in a daily environment. Section V compares our approach to related systems in literature.

The main contribution of the paper is to formulate local plan recognition in a framework, which allows inference and learning (section II and III). Further, recognizing local navigation plans requires a procedure to link the past to the present without any fixed global references. The crucial insight is that recorded driving data can provide global information, and, hence allow learning the required local models (section III-A). We formalize this insight as a set of soft probabilistic constraints (section III-C) that define the solution to local plan recognition (section III-D). To demonstrate the validity of our approach, we predict the local navigation plans of a real wheelchair user in section IV.

II. LOCAL PLAN RECOGNITION

The following example illustrates the different aspects of recognizing the navigation plan of a user. Let us assume that the user starts from the position shown in the left side of figure 1.

1) *Plan generation*: The robot first generates all possible user plans for a given situation (the plan library \mathcal{I}_k). These plans represent local navigation intents in an environment. In practice it might be necessary to discretize, while preserving a locally dense set.

2) *User model*: Next, the robot reasons about the user's abilities to determine the probability of each plan. Calculating the probability of docking at the table (estimating the posterior) directly is more difficult than obtaining the probability of the signals required to reach the table, when the user has a plan to reach the table in mind. We call the latter probability *user model*. Predicting user plans from user signals is generally more difficult than the inverse, because the user-plan space is higher dimensional. A sequence of input signals generates a plan.

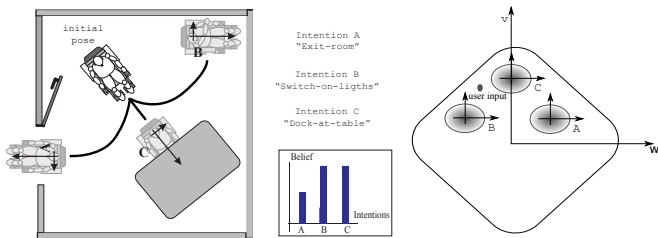


Fig. 1. The left side of the figure shows the wheelchair and three potential destinations. The user model is represented on the right. Darker colors indicate the region where user inputs are more likely for a given intention.

Normally, the user controls an electric wheelchair with a joystick-like device, where the vertical axis changes the linear speed and the horizontal axis controls the angular speed. The user model in figure 1 is a probability distribution in the space of linear and angular speeds rather than in the space of possible user plans. Reasoning about navigation plans given user inputs becomes possible by considering what inputs a user would produce for each plan. For instance, docking at the table in figure 1 requires inputs along the vertical axis, whereas leaving the room yields inputs in the shaded area on the right half-plane.

3) *Temporal reasoning*: The information available to the robot from the user model is quite limited. In figure 1 the robot would be unable to disambiguate intentions *B* and *C*. Hence, it becomes essential to consider past driving behavior.

Relating past driving to the present means to link past plans to current plans in order to initialize their probabilities. If the robot recomputes all plans to new local destinations each time it moves, then it needs to connect past plans to the present without employing their destinations as fixed anchor points. Figure 2 shows why this can be difficult. Connecting plans whose end destinations remain closest can fail. In figure 2 trajectories advancing forwards would be linked to plans leading backwards.

The robot discards global references in order to provide richer assistance, since the plans can be denser locally. However, it must accept additional uncertainty as a price for enhanced navigation. Absolute certainty when linking past and present plans is only possible if their goal locations remain fixed.

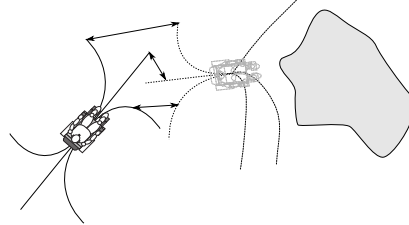


Fig. 2. Connecting past and present driving without global references.

A. Representation of user plans

The purpose of our probabilistic plan recognition framework is to recognize *all* possible user navigation intents, to estimate the uncertainty on these intents, and to make the estimation adaptive to the specific user that is interacting with the robot. In order to realize this, a representation of user intents is chosen that is different from previous approaches, which nearly all represent intents in terms of specific assistance algorithms such as *follow-corridor* or *avoid-obstacle*.

User intentions or user plans i can be generically described as a certain goal twist that the user wishes to achieve at a certain goal pose. The trajectory that the user has in mind to achieve this goal pose and goal velocity is also part of the intention representation. It is imperative to consider the trajectory as part of the user plan since a user can reach the same goal pose in a myriad ways.

Any intention can be modeled with this representation in a precise way, to *any* place in the robot's environment close by or far away.

B. Probabilistic model

In order to reason about the intentions of its user, the robot calculates the posterior probability of a set of navigation hypotheses, $i_k \in \mathcal{I}_k$, given past driving behavior $\mathcal{H}_{0:k}$ and the present input of the user, u_k . Bayes' rule inverts the estimation of this posterior by combining a user-specific likelihood, $p_{user}(\cdot)$, with a prior probability, $p_{k-1}(\cdot)$, as follows:

$$\underbrace{p_k(i_k | u_k, \mathcal{H}_{0:k})}_{\text{posterior over user plans}} = \underbrace{p_{user}(u_k | i_k, \mathcal{H}_{0:k})}_{\text{user model}} \cdot \underbrace{p_{k-1}(i_k | \mathcal{H}_{0:k})}_{\text{prior over user plans}} \cdot \eta \quad (1)$$

where η normalizes the posterior distribution. The history $\mathcal{H}_{0:k}$ includes all previous user inputs up to time $k-1$, the sequence of robot actions $a_{0:k}$, the sequence of robot positions, $x_{0:k}$, and any external sensor readings, $z_{0:k}$.

The user model, p_{user} , quantifies how much information the input u_k reveals about the user's hidden navigation plan. The prior, p_{k-1} , in equation (1) represents the temporal component of plan recognition. Hypotheses, which are linked across time, inherit posterior probability.

C. Probabilistic dynamics

We assume that the user has plan i_0 in mind when she starts interacting with the robot. However, she only

communicates her intentions through an input device, which produces input \mathbf{u}_0 . The robot observes \mathbf{u}_0 and moves to pose \mathbf{x}_1 in the environment after executing action \mathbf{a}_1 . When the robot changes its pose after executing action \mathbf{a}_1 , the user adapts by devising a new plan \mathbf{i}_1 , which results in user interface signal \mathbf{u}_1 , and so forth.

The Dynamic Bayesian Network (DBN) [1], [2] in figure 3 represents the robot's state estimation problem graphically. It models the probabilistic relations between random variables, which the robot can observe or has to estimate, and it also incorporates the flow of time from left to right in the model.

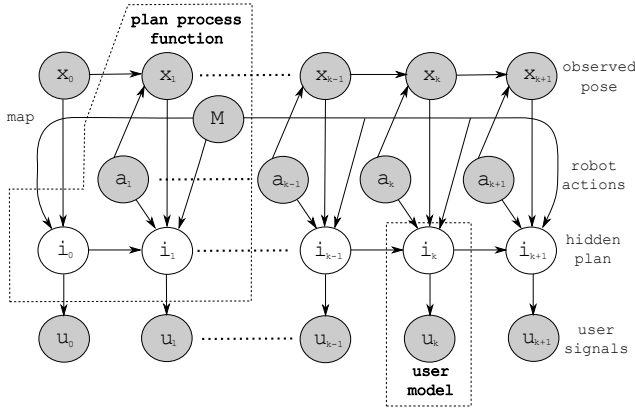


Fig. 3. This Dynamic Bayesian Network (DBN) represents the evolution of user plans during navigation. The robot tracks this evolution in order to understand how the user interacts with the wheelchair. A random variable is represented inside a circle or node, which is either shaded for observed variables or uncolored for hidden variables. The complete Dynamic Bayesian Network represents a factorization of the joint probability over all variables present in the model. Each individual factor is a conditional probability of a child node conditioned on its parent nodes, as specified by the direction of the edges. Therefore, directed edges indicate probabilistic dependencies between variables.

In human-robot navigation the evolution of user plans governs the time scale of the robot's DBN. The robot needs to query its sources of information, i.e. its external sensors \mathbf{z} , the user signals \mathbf{u} , the localization \mathbf{x} and map-building \mathbf{M} capabilities at the bandwidth of the user. For the experiments with a continuous user interface in section IV we chose a sampling period of $\Delta T_{\text{sampling}} = 0.3 \text{ s}$. This value is inspired by the review of human reaction times during breaking and steering cited in [3]. Under ideal conditions, when the driver reacts to expected events, a lower bound of reaction times ranges from 0.70s to 0.75s .

In the DBN of figure 3, we assume that actions \mathbf{a} are marginally independent, i.e. they are independent of any other variable. Although the robot might choose its actions depending on estimated user intentions, it must assume that this causal link is unknown to the user. The user ignores what the robot knows about her plans. Instead, she only reacts to the consequences of the displacement. These actions could also have been chosen randomly and the user would react in the same way for the same sequence of observed displacements.

The following sections explain each component of the

DBN further, namely the *user model* (section II-D) and the *plan process function* (section II-E). Section II-G introduces the generic algorithm to calculate the posterior over user plans, $bel(\mathbf{i}_k)$, in function of the user model and the plan process probabilities.

D. User model

We model how a user converts navigation plans into inputs to the robot in order to understand how and where she intends to drive. The input of the user depends directly on her intentions. Relations to any other quantity, such as robot actions or the robot's location, are always mediated through the user's plan, which the user adapts to her current circumstances. We model this dependency by a probabilistic *user model* that corresponds to the likelihood of observed signal \mathbf{u}_k if the user has plan \mathbf{i}_k in mind.

$$\text{user model: } p_{\text{user}}(\mathbf{u}_k | \mathbf{i}_k) \quad (2)$$

Note that the user model needs to be adapted to every user interface and to every user. It is the personalized component of the framework, which enables intuitive and user-specific driving assistance. Our probabilistic user model quantifies the inherent uncertainty related to driving. For instance, users execute maneuvers imperfectly or drive differently every time. Further, the model also considers the uncertainty stemming from modeling errors or simplifications.

E. Plan process function

The *plan process function* in the DBN of figure 3 reflects how the user adapts her plans in response to actions that change the position of the robot. We assume that the user changes her plan \mathbf{i}_{k-1} to plan \mathbf{i}_k starting from position \mathbf{x}_k after observing the effects of action \mathbf{a}_k . This plan transition is mirrored by the robot in order to understand how the user's plans are evolving.

Formally, plan transitions are modeled by a *plan process function*, which consists of two parts. The first component adapts all previous user plans at time $k-1$, \mathcal{I}_{k-1} , to trajectories \mathcal{I}_k starting from position \mathbf{x}_k . The second component initializes the probability of the new set of trajectories at time k . This initialization relies on the *plan process probability*, which serves to calculate a prior over user plans:

$$\text{plan process probability: } p_{\text{ppf}}(\mathbf{i}_k | \mathbf{i}_{k-1}, \mathbf{a}_k, \mathbf{x}_k, \mathbf{M}) \quad (3)$$

Equation (3) makes the dependence of the plan process probability on current pose \mathbf{x}_k and the extended map \mathbf{M} explicit. This extended map \mathbf{M} summarizes all available information about an environment (e.g. a-priori maps, online constructed maps and the history of sensor readings $\mathbf{z}_{0:k}$). In practice, the dependency on map \mathbf{M} and pose \mathbf{x}_k implies that each user plan \mathbf{i}_k considers obstacles in the environment and starts from the robot's current pose.

F. Differences user model and plan process function

It is important to highlight the difference between the *plan process probability* and the *user model*. The *plan process probability* quantifies how information from the past propagates to the present without reference to the user's current input \mathbf{u}_k . As a consequence, the plan process probability is largely independent of a particular user. Instead, it reflects how robot actions influence probability assignments to the plan library, \mathcal{I}_k . On the other hand, the *user model* is user-specific. It is a probabilistic model of how a particular user translates her plans into inputs for the robot. The user model corrects the prior over user plans, which originates from the plan process probability, by incorporating user-specific knowledge.

G. Probabilistic computations in plan recognition

The Dynamic Bayesian Network of figure 3 entails a series of conditional independence assumptions, which were explained in the preceding sections. They allow a recursive formulation of the *plan recognition* probability in equation (1), consisting of a *time-update* and a *measurement-update* steps.

$$\text{posterior belief: } \text{bel}(\mathbf{i}_k) \equiv p_{\text{post}}(\mathbf{i}_k \mid \mathbf{u}_{0:k}, \mathbf{a}_{1:k}, \mathbf{x}_{0:k}, \mathbf{M}) \quad (4a)$$

$$\begin{aligned} \text{time-update: } & p_{\text{prior}}(\mathbf{i}_k \mid \mathbf{u}_{0:k-1}, \mathbf{a}_{1:k}, \mathbf{x}_{0:k}, \mathbf{M}) \\ &= \sum_{\mathbf{i}_{k-1}} p_{\text{ppf}}(\mathbf{i}_k \mid \mathbf{i}_{k-1}, \mathbf{a}_k, \mathbf{x}_k, \mathbf{M}) \cdot \text{bel}(\mathbf{i}_{k-1}) \end{aligned} \quad (4b)$$

measurement-update:

$$\text{bel}(\mathbf{i}_k) = \frac{p_{\text{user}}(\mathbf{u}_k \mid \mathbf{i}_k) \cdot p_{\text{prior}}(\mathbf{i}_k \mid \mathbf{u}_{0:k-1}, \mathbf{a}_{1:k}, \mathbf{x}_{0:k}, \mathbf{M})}{p_{\text{norm}}(\mathbf{u}_k \mid \mathbf{u}_{0:k-1}, \mathbf{a}_{1:k}, \mathbf{x}_{0:k}, \mathbf{M})} \quad (4c)$$

III. LEARNING THE MODELS OF LOCAL PLAN RECOGNITION

This section describes our approach to learn the *plan process probability* and the *user model* from driving examples.

A. Learning the plan process probability

Only global knowledge allows absolute certainty when linking past plans to the present. This key insight opens a way to acquire a local *plan process probability*. Figure 4 shows the essence of the method. During calibration the complete trajectory of the robot represents a source of global knowledge. At each sampling step k , different user plans of the plan library \mathcal{I}_k resemble the future path of the robot $\mathbf{x}_{k:\text{end}}$ to varying extents. Actually, each user plan \mathbf{i}_k is a potential future trajectory and can therefore be compared to the actual future trajectory of the robot with the benefit of hindsight.

The comparison between user plans and the robot's future path is independent of a particular user, or of where and how a particular user really intended to navigate. It is merely a comparison between potential and actual future navigation

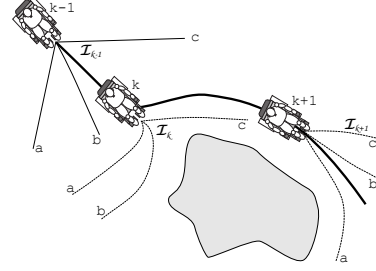


Fig. 4. Plan process probability for local trajectories calibrated with the robot's future path. The bold trajectory represents the path of the robot. At each time step, a set of local trajectories \mathcal{I} emanates from the robot. Similar trajectories to the robot's path inherit probability across time.

outcomes, i.e. between user plans \mathcal{I}_k and the truly executed future robot trajectory $\mathbf{x}_{k:\text{end}}$.

In order to calculate the plan process probability, it seems reasonable to assume that equivalent user plans in terms of future path of the robot should inherit probability from their ancestors from previous times. Next we introduce a metric to measure the resemblance between trajectories.

B. Resemblance between trajectories

1) *Hausdorff distance*: If \mathcal{P} and \mathcal{Q} are curves, then the Hausdorff distance is defined as:

$$d_{HD}(\mathcal{P}, \mathcal{Q}) = \max\{\vec{\delta}(\mathcal{P}, \mathcal{Q}), \vec{\delta}(\mathcal{Q}, \mathcal{P})\} \quad (5a)$$

$$\vec{\delta}(\mathcal{P}, \mathcal{Q}) = \sup_{\mathbf{x} \in \mathcal{P}} \inf_{\mathbf{y} \in \mathcal{Q}} d(\mathbf{x}, \mathbf{y}) \quad (5b)$$

where $\vec{\delta}(\mathcal{P}, \mathcal{Q})$ is the directed Hausdorff distance and $d(\mathbf{x}, \mathbf{y})$ is a secondary point-metric. The Hausdorff metric measures the maximal displacement, which is possible when travelling on the shortest path from one trajectory to the other. Besides the brute-force approach of comparing all points pair-wise, an algorithm exists to compute the metric in linear time for any L_p point distance [4].

2) *Pose metrics*: It remains to define an appropriate metric to compare points of a user plan and the robot's future trajectory. Normally, the Hausdorff metric assumes a point-particle motion model. Then, the Euclidean distance is a proper measure of metric displacement between curves. However, this assumption is unreasonable for a robot that can turn on-the-spot. Its point displacement can be zero although it has turned. The local plan process function should also consider such type of motions.

Martinez and Duffy [5] propose a frame-invariant metric to compare rigid body motion that is well-defined. Let ν_B be the set of poses of rigid body B , and let B_1 and B_2 be two poses, then a metric to measure the distance from B_1 to B_2 is:

$$d_{MD_1}(B_1, B_2) = \|\delta_M\| = \max_{Q \in B} \|\delta_Q\| = \max_{Q \in B} \|\mathbf{r}_{Q1} - \mathbf{r}_{Q2}\| \quad (6)$$

where $\|\cdot\|$ is the standard Euclidean norm and δ_Q is the displacement vector of an arbitrary point of the finite rigid body B , which at configuration B_1 is located at \mathbf{r}_{Q1} and at configuration B_2 at \mathbf{r}_{Q2} . δ_M represents the displacement of the point farthest away from the screw axis defining the motion. This point is not necessarily unique, but its norm is. In case the motion is a pure translation, all displacements are equal and any point can be chosen to calculate the metric.

C. Distance likelihood

It is still necessary to quantify what the resemblance of a certain plan \mathbf{i}_k to the robot's future path *means* in probabilistic terms, and what this information implies for the temporal evolution of user plans. This information represents domain knowledge, which we elicit from the properties of the problem. The Bayesian formulation of *plan recognition* of section II allows to fuse such subjective information with other sources of uncertain knowledge in a probabilistically sound way.

Let $\mathbf{d}_k = [d_1, \dots, d_m]^T$ be the observed distances of all user plans at time k to the robot's future trajectory. If the user intends to follow the robot's future trajectory, she will have a plan resembling the future trajectory. Therefore, each distance d_j offers an uncertain measurement of the likelihood of each plan \mathbf{i}_k . We can define the following *distance likelihood*, which relates distance measurements \mathbf{d}_k to the plan library \mathcal{I}_k .

$$\text{distance likelihood: } p_{\text{dist}}(\mathbf{d}_k | \mathcal{I}_k) \quad (7)$$

The *distance likelihood* should decrease in function of distance, as far-away plans are less likely to correspond to the robot's future path. Equation (8) defines the functional form of a candidate distance likelihood.

$$p_{\text{cos}}(d | \mathbf{i}_k) = \frac{1 + \cos\left(\pi \frac{d - d_{\min}}{d_{\max} - d_{\min}}\right)}{\kappa_{\text{cos}}(d_{\max}, d_{\min})} \quad (8)$$

where $\kappa_{\text{cos}}(d_{\max}, d_{\min})$ is a normaliser that ensures that the likelihood obeys the laws of probability. The likelihood is defined over the range of distances $[d_{\min}, d_{\max}]$ for the observed robot trajectory.

The rationale behind introducing a *distance likelihood* is to reduce the amount of data needed to calibrate the *plan process probability*. Normally, in order to calculate a probabilistic model of plan transitions, one would require sufficient data about *all* possible transitions. The distance likelihood can reduce the amount of training data because information spreads from observed transitions to unobserved transitions between plans.

D. Calibration of the plan process function

The *distance likelihood* imposes soft probabilistic constraints on the user's true intention, which is hidden for the robot. Therefore, the *distance likelihood* is a type of uncertain evidence called *virtual evidence* [6]. It only offers direct, albeit uncertain, information about the user's plan \mathbf{i}_k . This information can propagate inside the Dynamic

Bayesian Network of figure 3 to learn the parameters of local plan recognition. These parameters are the *plan process probability*, and if the user interface is discrete, also the *user model*. User models for continuous interfaces are discussed in the next section.

Please note that the *distance likelihood* is only available during training, when the robot's full trajectory is known. In operation, the robot lacks access to its future path.

We compute the solution for the *plan process probability* as the Maximum A Posteriori (MAP) distribution obtained by the Expectation Maximisation (EM) algorithm [7]. The exact procedure for a Bayesian Network is described in [8]. In broad terms, during the Expectation (E) step the expected sufficient statistics of each component distribution of the Bayesian Network are calculated by propagating evidence with, for instance, the Junction Tree Algorithm [9]. Then, in the Maximization step (M), the expected sufficient statistics are assumed to be the actual statistics, which allow to compute each component's MAP distribution.

The MAP distribution allows placing a prior on the learning procedure, which makes unobserved data still possible. If the robot were using the Maximum Likelihood *plan process probability*, unobserved plan transition during training would remain impossible in operation.

E. Learning the user model

If the user interface is continuous, which is approximately true for conventional wheelchair joysticks, it is possible to extract a continuous probabilistic mapping from user plans to user inputs. In [10] we propose a learning framework based on Gaussian Process Regression [11], which predicts user inputs in function of continuous user plans.

Conventional joysticks consist of two independently controllable axes that define the linear and angular speeds respectively. Therefore, it is interesting to split the user model p_{user} into two components, one for the linear speed and another one for the angular input. Such factorisation follows from the product rule in probability and is stated in equation (9).

$$p_{\text{user}}(\mathbf{u}_k | \mathbf{i}_k) = p_{\text{user}}(v_k^u | \omega_k^u, \mathbf{i}_k) \cdot p_{\text{user}}(\omega_k^u | \mathbf{i}_k) \quad (9)$$

Each component in equation (9) defines a probability distribution over user inputs v_k^u and ω_k^u , which we represent as a Gaussian Process, \mathcal{GP}_v or \mathcal{GP}_ω .

$$v_k^u \sim \mathcal{GP}_v(\mathbf{0}, \text{cov}_v(\mathbf{i}_k, \omega_k^u)) \quad (10a)$$

$$\omega_k^u \sim \mathcal{GP}_\omega(\mathbf{0}, \text{cov}_\omega(\mathbf{i}_k)) \quad (10b)$$

where \sim denotes that the user input is distributed according to a specific \mathcal{GP} with covariance function $\text{cov}_v(\cdot)$ or $\text{cov}_\omega(\cdot)$. The mean function of the Gaussian Process is zero for normalized training data.

When learning the parameters of the Gaussian Process \mathcal{GP}_v or \mathcal{GP}_ω it is necessary to calculate a ground truth plan \mathbf{i}_k^{GT} . We employ the same metric of section III-B to obtain the plan with lowest distance to the actually executed

trajectory and use this plan to train the Gaussian Processes. An important requirement for learning is that all user plans are sufficiently represented during training, i.e. that each plan is executed several times.

IV. EXPERIMENTAL RESULTS

The experiments have been conducted in a home-like environment at iRv/ vilans in Hoensbroek, The Netherlands (cf. figure 6(a) for a map of the environment). We collected two datasets on two different days. One dataset will be employed for training the models and the other to evaluate their performance.

The training dataset lasts about 9 minutes during which the user had to solve different navigation tasks. In the test data the user drives for 5 minutes performing similar maneuvers in the same environment.

No driving assistance was provided since the purpose of the experiment is to study the user’s driving style. Typical maneuvers include most of the situations encountered in daily-life such as driving forwards, crossing narrow openings, turning at different radii, and also executing more complex trajectories to reach far-away poses in the environment. Collisions with the environment happened occasionally.

A. Description of the user

Figure 5(a) shows the user participating in experiments within the EU-funded MOVEMENT project [12]. The user, who was 19 at the time of the experiments, is a female presenting symptoms of spastic quadriplegia¹. As a consequence of her condition, she was incapable of steering a conventional electrical wheelchair safely. During the experiments, collisions, especially in narrow locations, abounded. She clearly fits the profile of users who currently are not prescribed power wheelchairs and could benefit enormously from navigation assistance.

B. Plan Generation

The robot constructs a collision-free local plan library by considering the latest sensor data and the position within an a-priori map of the environment (provided the robot is well localized). Efficient collision checking is possible for a large number of paths by employing pre-computed look-up tables [14]². Map features and perceived obstacles are placed inside this pre-computed grid when updating the set of local paths. Figure 5(b) shows an example of local plans, adapted to dynamic obstacles at the robot’s current location.

¹Spasticity generally results from damage to the part of the brain that controls voluntary movement. It may also occur when the nerves traveling from the brain down to the spinal cord have been damaged [13]. Quadriplegia means that it affects all limbs.

²All collision-free trajectories of a plan library with 144 plans are calculated in about 20 – 30ms on an Intel Core Duo Processor @2.6GHz and 4GB of RAM.

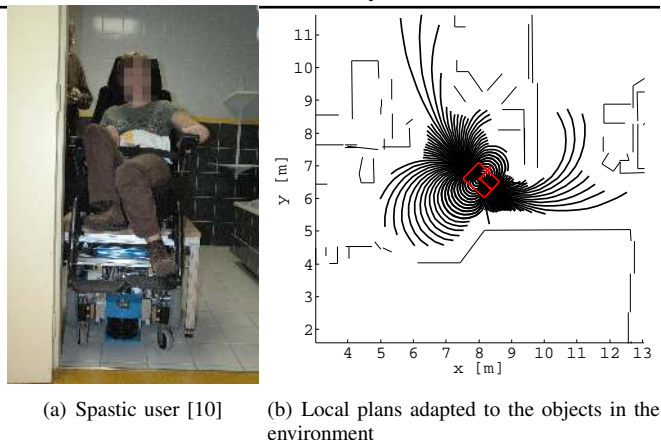


Fig. 5. Snapshot of the navigation experiments at iRv/vilans.

C. Results plan recognition

In this section we describe the results of recognizing the local navigation plans of a spastic user. We combine a user-specific driver model with past driving behavior. The user model for the maneuver in 6(a) is shown in figure 6(b) and the posterior in figure 6(c). The solid line in the probability plots represents ground-truth according to the metric discussed in section III-B. The robot is able to reduce the uncertainty of the user model by including past driving information.

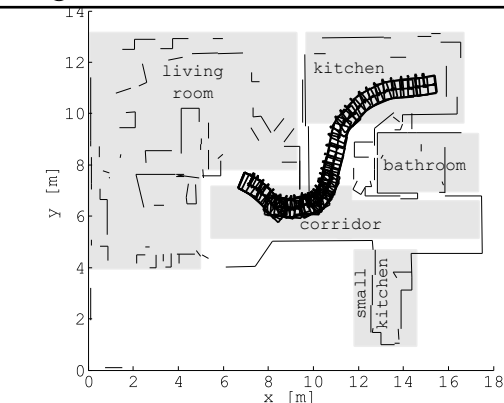
Our performance benchmark is the concentration of probability mass around ground-truth. Visual inspection shows reasonable convergence, which increases from the user model to the posterior. In order to quantify the convergence, we use the Kullback Leibler divergences $KL(p(\mathbf{i}_k)||U(\mathbf{i}_k))$ and $KL(p(\mathbf{i}_k)||q_{GT}(\mathbf{i}_k))$. The first divergence measures the difference between $p(\mathbf{i}_k)$ and a uniform distribution over plans, $U(\mathbf{i}_k)$. Higher values imply a more informative distribution $p(\mathbf{i}_k)$. Conversely, $KL(p(\mathbf{i}_k)||q_{GT}(\mathbf{i}_k))$ quantifies the spread around ground truth and lower values are better. We choose $q_{GT}(\mathbf{i}_k)$ as a normalized, integer-valued, Gaussian that is centered on ground-truth and has a standard deviation of 8 plans.

TABLE I
ANALYSIS OF CONVERGENCE TO GROUND TRUTH

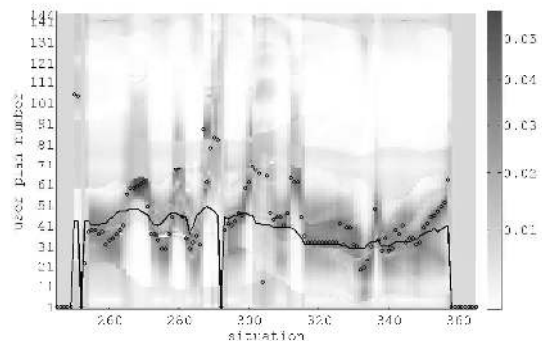
distribution	$KL(p(\mathbf{i}_k) U(\mathbf{i}_k))$	$KL(p(\mathbf{i}_k) q_{GT}(\mathbf{i}_k))$
user model	1.06 ± 0.38	14.92 ± 5.55
posterior	2.12 ± 0.56	3.80 ± 1.27

In table I, the first value represents the mean and the second one standard deviation. To put the values of $KL(p(\mathbf{i}_k)||U(\mathbf{i}_k))$ in perspective, a Gaussian with mean 72 and standard deviation 17 has $KL(\mathcal{N}_{int}(72; 17)||U(\mathbf{i}_k)) = 1.04$, and if $\sigma = 8$, then $KL(\mathcal{N}_{int}(72; 8)||U(\mathbf{i}_k)) = 2.12$.

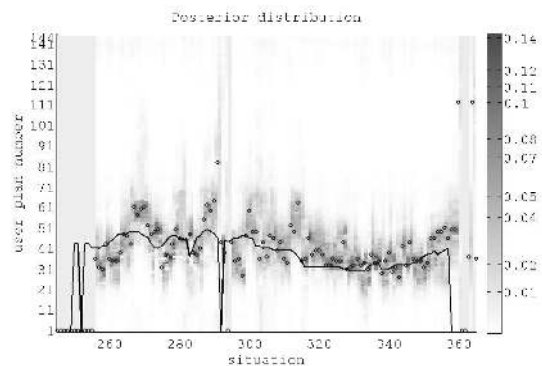
Table I confirms that the posterior converges better to ground truth compared to the user model. However, we believe a moderate level of noise is necessary to avoid an overconfident posterior, which is occasionally wrong. For a



(a) maneuver



(b) user model



(c) posterior

Fig. 6. Recognizing the user's driving intention.

more extensive analysis of the maneuver, please refer to [15].

V. RELATED WORK

A. Navigation assistance

Most assistive robots, such as wheelchairs, robotic guides or walking aids share the control with the user through task-related assistance behaviors. The robot helps to *avoid obstacles*, *dock at tables* or *traverse doorways* [16], [17]. Other approaches rely on environment properties to define the most appropriate navigation assistance, such as *outdoor* versus *indoor* [18], *cluttered* versus *open*, or *dynamic* versus *static* [19].

Due to the broad definition of assistance, the behaviors still need to estimate the actual navigation plan of the user.

It remains the responsibility of the shared controller to decide whether an obstacle should be avoided to the left or to the right, and this estimation is often implicit. In contrast, we propose an explicit estimation of the navigation plans of the user, which considers the user's abilities and needs.

An important difference of our probabilistic framework, is that besides environmental information and user signals, we quantify the inherent uncertainty of driving. Users drive differently every time and the actual navigation plan is hidden for the robot. The framework merges past beliefs regarding user plans with new evidence from user signals.

B. History of the framework

The problem of plan recognition in the context of human-robot wheelchair navigation was first proposed by Demeester et al. [20]. The authors demonstrated how to recognize the user's intention to drive to a goal location. In this early work, user intentions were simply end poses to be reached at an end velocity. Later work [21], [22] recognized the need to include also the user's plan to reach a goal location in the intention representation, as users might reach the same goal location in different ways.

The framework evolved further by representing user plans as local trajectories, since immediate assistance should rely primarily on local information and global maps are not always available. In [10] Hüntemann et al. describe a learning technique to acquire user models based on local trajectories for arbitrary driving styles. This work extends the results of [10] by completing the process of plan recognition for local trajectories.

C. Other frameworks of plan recognition

The field of plan recognition originated in the domain of story understanding. Schmidt et al. [23] define plan recognition as the process of identifying the goals of an agent together with the plan to achieve those goals. Kautz and Allen [24] apply deductive inference on a plan library to identify the intention of an agent. However, their system disregards the plans' prior probabilities. Unlikely hypotheses and very probable plans might explain equally well a set of observations. In order to overcome this shortcoming, Charniak and Goldman [25] propose a system for natural story understanding, which relies on incrementally constructed Bayesian Networks. This system lacks a procedure to learn the parameters of the model in the domain. Dynamic Bayesian Networks have also been applied to adventure games [26].

Although plan recognition has a long history, each application requires different domain-specific models. The designer needs to identify the set of valid plans, which the system should recognize robustly. Further, a procedure to calibrate the parameters of the chosen model is essential. The main contribution of this paper is to describe a complete framework to recognize navigation plans and to make this recognition adaptive to each user.

Approaches that are comparable to our probabilistic framework can be found in the analysis of motion behaviors by

Bennewitz et al. [27] and Glover et al. [28]. In contrast to our framework, however, they disregard user inputs while predicting the motion of humans. Further, they assume implicitly that users can go everywhere autonomously.

VI. CONCLUSIONS AND FUTURE WORK

In this paper we propose a probabilistic framework to recognize local navigation plans. The framework combines present and past information with user-specific driving behavior to derive a probabilistic estimate of the user's navigation plans.

We have shown the importance of employing local trajectories as navigation intents, which offer rich collision information and allow the user to drive anywhere. In order to recognize the user's navigation plans, it was necessary to connect past user intentions to present plans in absence of global references.

The paper also proposes a probabilistic framework to learn the components of local plan recognition, namely the *user model* and the *plan process function*. The former quantifies how a user transforms mental plans into inputs to the robot, whereas the plan process function models how user plans evolve in response to robot actions. The user model is learned via Gaussian Process Regression. The plan process function is calibrated in a Dynamic Bayesian Network by a probabilistic distance likelihood. This likelihood is a domain-specific function that models how much information the distance between user plans and the actual robot trajectory provides about the user's hidden intention.

Future work will compare driver models of users with different abilities. We are also planning to integrate the recognition of local navigation plans with a framework to share the control with the user.

ACKNOWLEDGMENT

The authors wish to thank the patient participating in the tests, the mother of the patient and also the therapist of the patient, Silvie Vanstipelen, for assisting during the complete duration of the trial. The tests took place at iRv/vilans in Hoensbroek with the kind help of Ger Cremers and Gert Jan Gelderblom. The authors gratefully acknowledge the financial contribution of the EU-funded project MOVEMENT (6th Framework Programme, FP6-511670) and RADHAR (7th Framework Program FP7-ICT-248873).

REFERENCES

- [1] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques*. The MIT Press, 2009.
- [2] K. Murphy, "Dynamic bayesian networks: representation, inference and learning," Ph.D. dissertation, University of California, Berkeley, 2002.
- [3] M. Green, "How Long Does it Take to Stop?" Methodological analysis of driver perception-brake times," *Transportation Human Factors*, vol. 2, no. 3, pp. 195–216, 2000.
- [4] R. Shonkwiler, "Computing the Hausdorff set distance in linear time for any L_p point distance," *Information Processing Letters*, vol. 38, no. 4, pp. 201–207, 1991.
- [5] J. Martinez and J. Duffy, "On the metrics of rigid body displacements for infinite and finite bodies," *Journal of Mechanical Design*, vol. 117, p. 41, 1995.
- [6] J. Pearl, *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann, 1988.
- [7] A. Dempster, N. Laird, D. Rubin, et al., "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 39, no. 1, pp. 1–38, 1977.
- [8] D. Heckerman, *Learning in graphical models*. Kluwer Academic Publishers, 1998, ch. A tutorial on learning with Bayesian networks, pp. 301–354.
- [9] F. V. Jensen and T. D. Nielsen, *Bayesian Networks and Decision Graphs*, 2nd ed. Springer, 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*, pp. 285–292, 2008.
- [11] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*, 2nd ed., ser. Adaptive Computation and Machine Learning. The MIT Press, 2006.
- [12] P. Mayer, G. Edelmayer, G. J. Gelderblom, M. Vincze, P. Einramhof, M. Nuttin, T. Fuxreiter, and G. Kronreif, "MOVEMENT –Modular Versatile Mobility Enhancement System," *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2892 – 2897, 2007.
- [13] D. Kantor. (2007) Spasticity – Medical Encyclopedia. Medical Center, University of Maryland. [Online]. Available: http://www.umm.edu/ency/index/eng_index.htm
- [14] E. Demeester, E. Vander Poorten, J. Philips, and A. Huntemann, "Design and evaluation of a lookup-table based collision-checking approach for fixed sets of mobile robot paths," *Intl. Symp. on Robotics (ISR)*, Taipei, Taiwan, 2012.
- [15] A. Huntemann, "Probabilistic Human-Robot Navigation – Plan recognition, user modelling and shared control for robotic wheelchairs," Ph.D. dissertation, Katholieke Universiteit Leuven, March 2011.
- [16] R. C. Simpson, S. P. Levine, D. A. Bell, L. A. Jaros, Y. Koren, and J. Borenstein, "Navchair: An assistive wheelchair navigation system with automatic adaptation," *Lecture Notes in Artificial Intelligence*, vol. 1458, pp. 235–255, 1998.
- [17] T. Röfer and A. Lankenau, "Architecture and applications of the bremen autonomous wheelchair," *Information Sciences*, vol. 126, no. 1, pp. 1–20, 2000.
- [18] H. Yanco, "Shared User-Computer Control of a Robotic Wheelchair System," Ph.D. dissertation, Massachusetts Institute of Technology, September 2000.
- [19] E. Prassler, J. Scholz, and P. Fiorini, "Navigating a Robotic Wheelchair in a Railway Station during Rush Hour," *The International Journal of Robotics Research*, vol. 18, no. 7, pp. 711–727, July 1999.
- [20] 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)*, pp. 1456–1461, 2003.
- [21] E. Demeester, A. Huntemann, D. Vanhooydonck, G. Vanacker, A. Degeest, H. Van Brussel, and M. Nuttin, "Bayesian Estimation of Wheelchair Driver Intents: Modeling Intents as Geometric Paths Tracked by the Driver," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5775–5780, 2006.
- [22] 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*, vol. 24, no. 2, pp. 193–211, February 2008.
- [23] C. F. Schmidt, N. S. Sridharan, and J. L. Goodson, "The Plan Recognition Problem," *Artificial Intelligence*, vol. 11, pp. 45–83, 1978.
- [24] H. Kautz and J. Allen, "Generalized plan recognition," *Proceedings of the Fifth National Conference on Artificial Intelligence.*, pp. 32–37, 1986.
- [25] E. Charniak and R. P. Goldman, "A Bayesian model of plan recognition," *Artificial Intelligence*, vol. 64, pp. 53–79, 1993.
- [26] D. W. Albrecht, I. Zukerman, and A. E. Nicholson, "Bayesian Models for Keyhole Plan Recognition in an Adventure Game," *User Modeling and User-Adapted Interaction*, vol. 8, pp. 5–47, 1998.
- [27] M. Bennewitz, W. Burgard, G. Cielniak, and S. Thrun, "Learning motion patterns of people for compliant robot motion," *The International Journal of Robotics Research*, vol. 24, no. 1, p. 31, 2005.
- [28] J. Glover, S. Thrun, and J. Matthews, "Learning user models of mobility-related activities through instrumented walking aids," vol. 4, pp. 3306–3312, 2004.