Navigation Concept

Xavier Perrin* Ricardo Chavarriaga† Roland Siegwart* José del R. Millán†
*Autonomous Systems Lab, ETHZ, Zürich, Switzerland
†IDIAP Research Institute, Martigny, Switzerland

***Abstract*— This paper presents a novel concept of semi-autonomous navigation where a mobile robot evolves autonomously under the monitoring of a human user. The user provides corrective commands to the robot whenever he disagrees with the robot's navigational choices. These commands are not related to navigational values like directions or goals, but to the relevance of the robot's actions to the overall task.**

A binary error signal is used to correct the robot's decisions and to bring it to the desired goal location. This simple interface could easily be adapted to input systems designed for disabled people, offering them a convenient alternative to existing assistive systems. After a description of the whole concept, a special focus is given to the decisional process, which takes into account in a Bayesian way the environment perceived by the robot and the user generated signals in order to propose a navigational strategy to the human user. The strength and advantages of the proposed semi-autonomous concept are illustrated with two experiments.

***Index Terms*— Semi-autonomous navigation, error signal, probabilistic reasoning, human-machine interaction.**

I. Introduction

Despite substantial advances in the field of robotics, a small category of end-users could benefit more from intelligent assistive systems designed for them, namely elderly or disabled persons. Today, most of these systems are focused on people able to manipulate joysticks, which cannot be properly controlled for paralysed or may present difficulties for elderly people.

Shared-control, collaborative control and semi-autonomous control are available strategies in order for a human user to operate a robotic device (see section II). Together with an appropriate protocol for action selection, these control architectures and the user input system could be optimised for elderly or disabled persons.

But the simpler the interface in terms of information flow from the human to the machine, the more steps are required to select the desired command. In this paper, we propose a novel system for an efficient asynchronous human-machine interaction designed for simple interfaces like single buttons, sip and puff systems and even the promising non-invasive brain-computer interfaces (BCIs). We want to rely mainly on the machine and give instructions only at key-points during the execution of a task. Instead of providing navigational commands, like in current semi-autonomous systems where the robot is autonomous on a relative short path but then requires a user input for the next movement to execute, we will provide monitoring signals about the robot's performance at solving the wished navigational task.

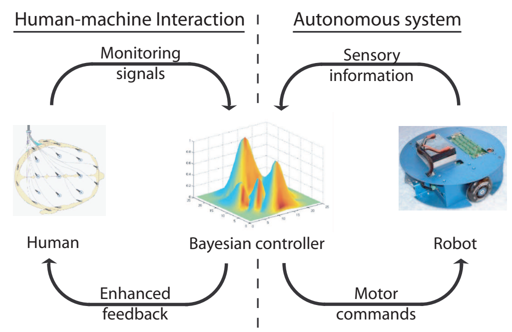


Fig. 1. Scheme of the proposed semi-autonomous navigation concept.

We define our semi-autonomous framework based on monitoring signals as follows:

> A semi-autonomous system is a robotic device, endowed with autonomous capabilities, interacting with a human user who emits corrective monitoring signals whenever necessary to achieve the goal.

This definition implies to have a fully autonomous agent able to execute navigational movements, as depicted on the right part of figure 1. Depending on the local perceived environment, the system chooses what action to execute. This controller's decision will be communicated to the human user by the mean of visual, audio or tactile cues. Based on this information, the user will have the possibility to emit a corrective signal in case of disapproval, which will prevent the execution of the proposed action and trigger a new choice from the controller. The human-machine interaction is shown on the left part of figure 1.

A binary error-related signal will be first provided through a keyboard interface. In future research, we plan to use an equivalent BCI signal. This paper describes our semi-autonomous navigation system and the related controller able to drive the user to the desired location in a efficient way based solely on error signals. In order to face incomplete knowledge and anticipate the uncertainty inherent with the future brain computer interface, the whole system and especially the controller are probabilistic and designed within a formal Bayesian Programming framework.

In section II, we will present related work. We will then describe our semi-autonomous concept and the Bayesian controller in section III. After showing some preliminary results in section IV, we will conclude by a summary and an outlook about the future work.

II. Related work

A. Humans controlling robotic devices

There are numerous applications of *shared-control strategies* for telemanipulated robots [8], surgical operations [16] and powered wheelchairs (review in [17]), which are widely used robotic platforms for researches in this field.

Robots and robotic wheelchairs can be distinguished by two major components:

a) Motion decision: A widely used technique is to take a decision given the sensory information and the user's commands using *Bayes' rules* [5], [19]. Some systems [2], [23] use a *semi-autonomous* framework, yet different from our definition: the user provides to the robot a direction for the next movement at each relevant position in the environment. The TAO wheelchair [10] has a *subsumptive reasoning* system that allows the most appropriate reactive behavior to emerge.

b) Motion generation: Besides the purely *reactive behaviors* of the TAO wheelchair, the are two main methods. The *behavior-based* motion generation matches sensory inputs to motor commands [13], [20]. The *planner-based* one takes into account the vehicle's kinematics and the sensory inputs to generate the best trajectory leading to a provided or inferred goal [5].

In general, the user has significant control over the wheelchair, but the user's commands are overridden when a danger of collision is detected, thus forbidding the wheelchair to approach an obstacle even if wanted.

On the contrary, *collaborative control* systems [9] use a dialog-based coordination strategy, where the robot evolves autonomously and asks the human for assistance when needed.

B. Human-machine interaction

Common input systems for human-machine interaction range from keyboards, joysticks and touch screens up to devices more adapted to disabled persons, like voice command, eye-tracking or sip and puff systems [18], [23].

In recent years, a novel technology has been studied, namely brain-computer interfaces (BCIs). The non-invasive, electroencephalography (EEG) based BCIs rely on the decoding of the brain activity in order to manipulate robotic devices, virtual keyboards or more general computer application [15], [22].

The work done by Ferrez and Millán [7] about the error potential is a recent addition to the available decoded brain-commands for human robot interaction. This potential indicates the human's awareness of an erroneous response made by the system when classifying the user intent. We will incorporate it into our system in the course of our future research.

III. Novel semi-autonomous concept

A. Concept overview

Our semi-autonomous system is divided into different interacting layers, as depicted in figure 2.

- **Interaction Layer**. This layer is in charge of the interaction between the human and the machine (decoding the user's signals) and between the machine and the human (providing a feedback of the system's status).
- **Sensory Layer**. This layer fuses in a probabilistic way multisensory information in order to extract the relevant features for the control of the system.
- **Behavioral Layer**. This layer implements a collection of a-priori or learned behaviors[1] for dealing with most navigational issues such as "corridor following", "door traversal" or "approaching a specific place".
- **Decision Layer**. This layer is responsible of selecting the next best behavior to adopt, given the perceived environment, the present used behavior and the signals coming from the user.

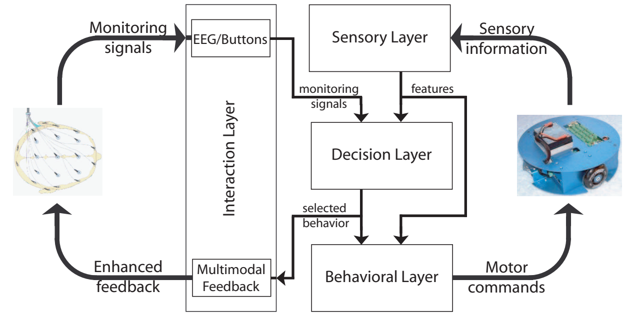


Fig. 2. Scheme of the different layers and their relations within the proposed semi-autonomous navigation concept.

In the Sensory Layer, information coming from the robot's sensors are fused together into a Bayesian occupancy grid providing an estimation of the obstacle poses [4]. Out of this local map of the environment, some basic features are extracted. As shown in figure 7b, they represent the directions and the associated distances of the closest obstacles or of the middle of the free traversable space in three regions around the robot: in front, on the left and on the right. We assume that the robot cannot go backwards. Some details about the Interaction Layer and the feedback modalities are given in section IV. For a description on how the features are associated to motor commands in a Bayesian way within the Behavioral Layer, please refer to [12].

After a presentation of the Bayesian programming framework, we will describe in more detail the Decision Layer, starting with the implementation of an autonomous controller and then enhancing it with semi-autonomous capabilities.

B. Bayesian programming

The Bayesian programming framework (BP) [6], [12] has been developed for designing robust robotic systems facing uncertain or incomplete knowledge. This framework provides both formal and computational tools for designing applications in a systematic way, as robot [4], [12] and game programming [11] or CAD modeling [14]. Sensor fusion with Bayesian occupancy grids, object tracking under partial occlusion and danger estimation have also been done [4]. A Bayesian program, as represented in figure 3, is made up of two parts: a description and a question.

[1] A behavior is a learned sensory-motor association [12].

$$\text{Prog} \begin{cases} \text{Desc} \begin{cases} \text{Spec } (\pi) \begin{cases} \text{Relevant variables} \\ \text{Decomposition} \\ \text{Forms} \end{cases} \\ \text{Identification based on Data } (\delta) \end{cases} \\ \text{Question: } P(S|K)? \end{cases}$$

Fig. 3. Structure of a Bayesian Program.

Description. In the description part, we define all the known information about the problem given a set of experimental data δ and preliminary knowledge π. It represents a joint probability distribution specified by the following components:

- A set of relevant variables (sensory, motor or internal state variables) on which the joint distribution is defined.
- A decomposition of the joint distribution as a product of simpler terms, respecting the Bayesian rules.
- The parametric forms assigned to each of the terms appearing in the decomposition.

Question. Given a distribution, it is possible to ask probabilistic questions by partitioning the set of variables into ”Search” (S), ”Known” (K) and ”Free” (F) variables.

C. Autonomous Controller

Inspired from the work of Le Hy [11], we will describe our autonomous controller by the following model in the BP framework:

a) Relevant variables:

F_i^t : discretized distance features at time t, computed in the $i \in [1, N_f]$ regions around the robot;

$B^t \, and \, B^{t+1}$: the set of different behaviors (N_b behaviors like *Forward*, turning *Left*, turning *Right* and *Stopping*) available at time t and $t+1$.

The general task the robot has to accomplish for the present study is to go where there is the most free space until it cannot go further. That is the reason why we care only about the distances inside of the three regions and not about the directions. Note that the discretized distances, allocated in five classes, are not measured metrically but are relative to each other by taking into account the surrounding traversable space.

b) Decomposition of the joint distribution: The resulting joint distribution is decomposed into probability distributions according to the Bayes rules and some conditional independence assumptions explained later:

$$P(F_i^t \ B^{t+1} \ B^t) \ = \\ P(B^t) \ P(B^{t+1}|B^t) \ \prod_{i=1}^{N_f} P(F_i^t|B^{t+1})$$

$P(B^t)$ represents the prior knowledge about the behaviors at the present time. $P(B^{t+1}|B^t)$ represents the probability of keeping the same behavior or switching to another. The $P(F_i^t|B^{t+1})$ terms link the features to the choice of the next behavior. These distributions allow us to simplify the dependencies between features. This so-called ”inverse programming” method works in the opposite way as Finite State Machine, where the selection of a behavior would depend on the combination of all features. Here, it consists in giving

B^{t+1} / B^t	Stop	Right	Forward	Left
Stop	0.25	0.10	0.10	0.10
Right	0.25	0.36	0.25	0.24
Forward	0.25	0.30	0.40	0.30
Left	0.25	0.24	0.25	0.36

TABLE I
$P(B^{t+1}|B^t)$.

Front distance / B^{t+1}	Stop	Right	Forward	Left
Low	0.2	0.35	0.06	0.35
Mid low	0.2	0.32	0.15	0.32
Medium	0.2	0.11	0.20	0.11
Mid high	0.2	0.11	0.29	0.11
High	0.2	0.11	0.30	0.11

TABLE II
$P(Distance\ in\ front|B^{t+1})$.

probabilities to the system about how a particular feature should look like, independently from the others, if we choose a given next behavior. Powerful and easily maintainable, this selection method only adds one probability table for each new feature, which reduces the computational complexity [11].

c) Forms and identification: All probability distributions are given as tables, except $P(B^t)$ which is a uniform distribution over all the behaviors. This is because we have no a priori information about this value when building the model. The content of the tables is set a priori by the programmer for the simple example shown in section IV and no identification phase took place. We want the robot to drive towards the most free space until it cannot go further. More complex applications may require learning techniques in order to capture probability distributions that reflects the desired robot's general behaviour [11].

Table I shows the transition probabilities between the behaviors ($P(B^{t+1}|B^t)$). One can see that the probability of staying in the same behavior is the highest and that when turning, there is a higher probability to return to *Forward* than turning in the other direction. Note that each column of the tables sums up to 1, as needed by the Bayes' rules.

Table II is an example of a probabilistic table describing the influence of a distance measure ($P(F_i^t|B^{t+1})$). The column corresponding to the *Forward* behavior should be read as follows: given that the chosen behavior is *Forward*, there is a high probability that the distance in front of the robot is between medium and high. Similarly, if the robot chose to go *Left* (or *Right*), there is a high probability that an obstacle is relatively close in front.

The question we ask to the Bayesian program is $P(B^{t+1}|F_i^t \ B^t)$, i.e. what is the next behavior given the present behavior and features. The Bayesian program for the autonomous controller is summarized in figure 4. This controller is able to drive the robot towards the most free space without taking into account the user's destination.

Program
- Description
  - Specification
    - Relevant variables:
      - F_i^t : discretized distance features
      - B^t, B^{t+1} : the set of current and next behaviors
    - Decomposition:
      - $P(F_i^t\ B^{t+1}\ B^t) =$
      - $P(B^t)\ P(B^{t+1}|B^t)\ \prod_{i=1}^{N_f} P(F_i^t|B^{t+1})$
    - Forms:
      - $P(B^t)$: uniform distribution
      - others : probability tables
  - No identification (tables given)
- Question:
  - $P(B^{t+1}|F_i^t,\ B^t)$

Fig. 4. Autonomous controller described in the BP formalism.

Authorisation Forward / B^{t+1}	Stop	Right	Forward	Left
0	0.5	0.5	0.0	0.5
1	0.5	0.5	1.0	0.5

TABLE III
$P(Authorisation\ Forward|B^{t+1})$.

$P(B^{\text{chosen}*})$	Stop	Right	Forward	Left
$A_{\text{fwd}}^t = 1$	0.02443	0.20436	0.76636	0.00485
$A_{\text{fwd}}^t = 0$	0.10458	0.87468	0.00000	0.02074

Fig. 5. Comparison between two controller's output when asking $P(B^{\text{chosen}*}) = P(B^{t+1}|F_i^t,\ B^t,\ A_k^t = \{1\})$, $k \in \{\text{Stop, Right, Left}\}$, using a set of features coming from experimental data. When $A_{\text{fwd}}^t = 1$, all behaviors are authorised; the selected behavior is *Forward*. When $A_{\text{fwd}}^t = 0$, the *Forward* behavior has been forbidden; the selected behavior is *Right*.

Program
- Description
  - Specification
    - Relevant variables:
      - F_i^t : discretized distance features
      - B^t, B^{t+1} : the set of current and next behaviors
      - A_j^t : authorization for each behavior
    - Decomposition:
      - $P(F_i^t\ B^{t+1}\ B^t\ A_j^t) =$
      - $P(B^t)\ P(B^{t+1}|B^t)\ \prod_{i=1}^{N_f} P(F_i^t|B^{t+1})$
      - $\prod_{j=1}^{N_b} P(A_j^t|B^{t+1})$
    - Forms:
      - $P(B^t)$: uniform distribution
      - others : probability tables
  - No identification (tables given)
- Question:
  - $P(B^{t+1}|F_i^t,\ B^t,\ A_j^t)$

Fig. 6. Semi-autonomous controller described in the BP formalism.

D. Semi-Autonomous Controller

We will now present the modifications made to the previous controller for converting it into a semi-autonomous controller where the human can interact with the robot.

The human user generates monitoring signals whenever the autonomy of the robot needs to be restricted. As the monitoring signal is related to an error signal, we can add the notion of behavior's authorisation to the autonomous controller. The recognition of an error signal would prevent the execution of the corresponding selected behavior, therefore reducing the set of available behaviors. Given this additional information, the Bayesian controller will be asked for a new solution, corresponding to the next best behavior.

In other terms, the user has to authorise the behavior proposed by the controller. In our probabilistic formulation, this notion of behavior authorisation corresponds to additional A_j^t boolean variables, one for each possible behavior. $A_j^t = 1$ means that the j^{th} behavior is authorised at time t, $A_j^t = 0$ meaning the contrary. The influence of the A_j^t terms on the choice of the behavior will be described in probabilistic tables of the form $P(A_j^t|B^{t+1})$, as the example given in table III. One can see that the authorisation for the *Forward* behavior has no influence on the other behaviors (probability of 0.5 in both cases) but that it strictly allows (probability of 1) or prohibits to go forward.

Figure 5 shows a comparison between two controller outputs, the first one without any restriction regarding the authorised behaviors and the second one after the processing of a user-generated error signal. The authorisation is then reset to 1 after a fixed time or after the execution of the allowed behavior.

The resulting version of the Bayesian controller for our proposed semi-autonomous navigation system using monitoring signals is described in figure 6.

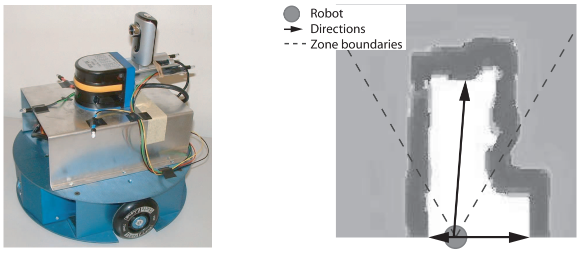
IV. Preliminary results

The semi-autonomous navigation (SAN) system was implemented and tested on an real robotic platform. The Smartease Robot, depicted on figure 7a, is a differential-drive mobile platform designed for educational purposes [3]. A Hokuyo PBS-03JN infrared range-finder was used as unique input sensor (99 values covering a field of view of 180° and ranging up to 3 meters [1]). The robot is covered with several LEDs, three of them, placed in front and on the two sides, giving a feedback of the controller's choice to the human user. Once the human user disagrees with this choice, he presses a key to send an error signal. An example of the robot sensory information and the extracted features is presented in figure 7b.

We designed three experiments in order to show progressively the capabilities of our SAN system. We recorded 50 trials for each experimental condition and then compared the duration of each trial and the number and nature of the user interventions. The translational and rotational speed limits were the same for all conditions.

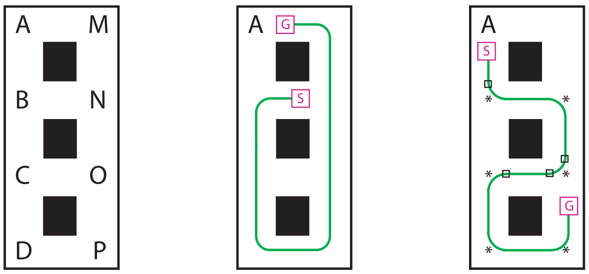
1) Experiment A: A maze-like environment (figure 8a) is used for experiment A in order to show the resulting general behavior of the SAN system when driving alone with no user intervention (similar as in figure 4).

The result corresponds to our expectations: the robot goes always where there is the most free space (figure 8b).



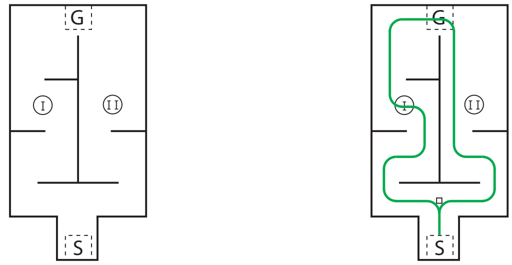
(a) (b)

Fig. 7. (a) The Smartease Robot equipped with the Hokuyo sensor and feedback capabilities. (b) Example of Bayesian occupancy grid with features superimposed (dark grey: occupied, light grey: unknown, white: empty).



(a) (b) (c)

Fig. 8. (a) Maze-like environment for experiments A and B. Graphical representation of the paths for Exp. A (b) and B (c). A square □ indicates where the user provided an error signal to the system and a star * where he provided a direction.



(a) (b)

Fig. 9. (a) Experimental environment for experiment C: two possible ways for going to a same goal location. (b) Graphical representation of the paths. The square □ indicates where the user provided an error signal.

Condition	Time [s]		User interventions	
	mean	std. dev.	mean	std. dev.
Original SAN	46.4	1.4	6	0
SAN with error signals	49.5	3.0	4.0	0.9

(a)

Condition	Time [s]		Percentage
	mean	std. dev.	
SAN driving alone, path I	46.0	2.9	56
SAN driving alone, path II	37.4	3.4	44
SAN with error signals, path I	-	-	0
SAN with error signals, path II	36.9	2.0	100

(b)

TABLE IV

NUMERICAL RESULTS FOR EXPERIMENTS B (a) AND C (b); 50 TRIALS WERE RECORDED FOR EACH EXPERIMENTAL CONDITION.

2) Experiment B: Within the same environment as for Exp. A, the second experiment (Exp. B) compares our SAN with user interventions (figure 6) to an original SAN (i.e. a direction is given at each place of interest) when solving a simple navigational task, represented here as a sequence of places to visit: B-N-O-C-D-P.

As represented in figure 8c, the task is solved by our SAN system in a similar amount of time (table IVa, Student's t-test for independent samples: $t_{99} = -0.9364,\ p > 0.05$) as with an original SAN method, an important characteristic for validating a new concept.

A particular advantage of the proposed system lies in the amount and nature of commands required from the user. While the original SAN requires six interventions (six times a minimum of two bits), the new approach requires an average of four binary error signals. The equivalent of a three-fold decrease of the information requirement may be of importance when dealing with simple interfaces (e.g. sip and puff systems) or low throughput interfaces (e.g. BCIs). Note that at certain intersections, the user may have to provide several error signals (e.g. location O). This is explained as follows: when the robot is in situation O, facing P, and receives an error signal, it turns right. But as it turns, the feature corresponding to the left side of the robot increases and becomes dominant, because it started to see a wall followed by the free space in direction of P, thus making the robot suddenly turn left. In order to go towards C, the user has to provide an additional error signal. Due to the imprecisions of the sensor and the Bayesian nature of the controller, the robot doesn't take twice the absolute same path, thus explaining the difference of time to complete the task and the number of user interventions. Using a short-term memory for saving the local environment together with the corresponding decision should overcome these problems.

3) Experiment C: In this experiment, the robot has to go from a start position (S) to a goal position (G) through two possible paths, the second one (II) being shorter (figure 9a). The robot evolves first autonomously using our SAN system and finds its way from S to G; then, in a second experimental condition, the user can provide monitoring signals (figure 9b). As can be seen in table IVb, there is a probability of about 50% that it takes the longer path I if the user does not intervene (actually, the robot went three times more through path I than II over the fifty trials). This shows that there is no predefined preferred direction when facing a left/right choice with equivalent corresponding features. If the user provides an error signal when the robot is willing to take the path I, the path II is selected as only alternative for completing the task. It is to mention that for this particular environment at most one error signal per trial is needed. The human-machine interaction allows to optimise the task because of the human's knowledge included in the decisional process, letting the semi-autonomous robot choose the optimal trajectory as shown in table IVb.

V. CONCLUSIONS AND OUTLOOK

In this paper, we presented a novel concept for semi-autonomous navigation and illustrated the strength of the approach using preliminary experimental results. Within the proposed concept, the robot evolves autonomously and the human user provides only monitoring signals when necessary. Contrary to prior work in the field of semi-autonomous navigation, these signals are not intended to be directional control commands, but they are related to the evaluation of the performance of the robotic device. Thus, our concept provides a reduced and simplified human-machine interaction and has significantly better applicability for non-trained humans.

Using a well-defined Bayesian Programming formalism, we describe the composition of our general semi-autonomous framework giving a special focus to the process of taking decisions in interaction with the environment and with the human. The proposed approach adequately uses the monitoring signals in order to efficiently bring the robot to the desired destination, without requiring sustained involvement from the human user. The BP formalism also unifies the way of dealing with the uncertainties of the perceived environment and of the inferred human's desired action. Furthermore, the integration of the uncertainties due to the future human-machine interaction is made easier, as the EEG signals classifier we will use in the next stages of this research delivers a probability of having recognised an error signal [7].

Experimental results showed that the proposed semi-autonomous system has similar performances compared to full robot control in terms of completion and completion time of a navigational task, while requiring less information from the user. Furthermore, the human-machine interaction may exploit the user's knowledge to guide the decisions in ambiguous situations (i.e. choosing between path I and II in experiment C using only the robot's local sensory information).

The future improvements of the semi-autonomous Bayesian controller include the teaching of the probability tables to the robot by driving it through the environment and showing it how to behave in order to overcome their actual manual filling [11], [12]. The addition of a short-term memory should allow to be consistent in the chosen behaviors and overcome some contradicting decisions as exposed in experiment B. We will also test our system in complexer environments with more than three alternatives.

In the present implementation decisions are taken based on local sensory readings and no learning occurs when there is an error signal. This can be improved by endowing the system with spatial reasoning capabilities. Thus, when navigating in frequently explored environments (e.g. user's apartment), the robot can build a representation of the environment and learn transition probabilities between places at each human-machine interaction [21], depending on contextual information like the user habits, the time of the day or other external variables. Hence, acquiring relevant information about most probable actions given a particular location that can be directly integrated onto the Bayesian reasoning system.

Acknowledgment

The authors would like to thank Juan Manuel Ahuactzin for his help with the Bayesian implementation and Roland Philippsen for the constructive comments. This work has been supported by the Swiss National Science Foundation NCCR 'IM2' and by the EC-contract number BACS FP6-IST-027140.

References

[1] M. Alwan, M. B. Wagner, G. Wasson, and P. Sheth. Characterization of Infrared Range-Finder PBS-03JN for 2-D Mapping. In *Proceedings of the IEEE International Conference on Robotics & Automation (ICRA '05)*, Barcelona, Spain, 2005.

[2] A. Argyros, P. Georgiadis, P. Trahanias, and D. Tsakiris. Semi-autonomous Navigation of a Robotic Wheelchair. *Journal of Intelligent and Robotic Systems*, 34:315–329, 2002.

[3] D. Burnier, S. Bouabdallah, and R. Philippsen. Smartease Robot. *http://asl.epfl.ch/contests/current/resources.html*, 2002.

[4] C. Coué, C. Pradalier, C. Laugier, Th. Fraichard, and P. Bessière. Bayesian Occupancy Filtering for Multi-Target Tracking: an Automotive Application. *International Journal of Robotic Research*, 25(1):19–30, 2006.

[5] 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. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '06)*, pages 5775–5780, Beijing, China, 2006.

[6] J. Diard, P. Bessière, and E. Mazer. A survey of probabilistic models, using the bayesian programming methodology as a unifying framework. In *The Second International Conference on Computational Intelligence, Robotics and Autonomous Systems (CIRAS '03)*, Singapore, 2003.

[7] P.W. Ferrez and J. del R. Millán. You Are Wrong!—Automatic Detection of Interaction Errors from Brain Waves. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence*, Edinburgh, UK, 2005.

[8] T. Fong and C. Thorpe. Vehicle Teleoperation Interfaces. *Autonomous Robots*, 11:9–18, 2001.

[9] Terrence W Fong. *Collaborative Control: A Robot-Centric Model for Vehicle Teleoperation*. PhD thesis, Carnegie Mellon University, 2001.

[10] T. Gomi and A. Griffith. Developing intelligent wheelchairs for the handicapped. In Mittal et. al., editor, *Technology and Artificial Intelligence*, pages 150–178. Springer-Verlag, 1998.

[11] R. Le Hy, A. Arrigoni, P. Bessière, and O. Lebeltel. Teaching Bayesian Behaviours to Video Game Characters. *Robotics and Autonomous Systems*, 47:177–185, 2004.

[12] O. Lebeltel, P. Bessière, J. Diard, and E. Mazer. Bayesian Robot Programming. *Autonomous Robots*, 16:49–79, 2004.

[13] S. Levine, D. Bell, L. Jaros, R. Simpson, Y. Koren, and J. Borenstein. The NavChair Assistive Wheelchair Navigation System. *IEEE Transactions on Rehabilitation Engineering*, 7:443–450, 1999.

[14] K. Mekhnacha, E. Mazer, and P. Bessière. The design and implementation of a bayesian CAD modeler for robotic applications. *Advanced Robotics*, 15(1):45–70, 2001.

[15] J. del R. Millán, F. Renkens, J. Mourino, and W. Gerstner. Non-invasive brain-actuated control of a mobile robot by human EEG. *IEEE Transactions on Biomedical Engineering*, 51(6):1026–1033, 2004.

[16] M. O. Schurr, G. Buess, B. Neisius, and U. Voges. Robotics and telemanipulation technologies for endoscopic surgery. A review of the ARTEMIS project. *Surgical Endoscopy*, 14(4):375–381, 2000.

[17] R. Simpson. Smart Wheelchairs: A Literature Review. *Journal of Rehabilitation Research and Development*, 42(4):423–436, 2005.

[18] R. Simpson and S. Levine. Adaptative Shared Control of a Smart Wheelchair Operated by Voice Control. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '97)*, volume 2, pages 622–626, 1997.

[19] R. Simpson and S. Levine. Automatic Adaptation in the NavChair Assistive Wheelchair Navigation System. *IEEE Transactions on Rehabilitation Engineering*, 7(4):452–463, 1999.

[20] D. Vanhooydonck, E. Demeester, M. Nuttin, and H. Van Brussel. Shared Control for Intelligent Wheelchairs: an Implicit Estimation of the User Intention. In *Proceedings of the 1st International Workshop on Advances in Service Robotics (ASER '03)*, pages 176–182, Bardolino, Italy, 2003.

[21] D. Verma and R. Rao. Goal-Based Imitation as Probabilistic Inference over Graphical Models. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in NIPS 18*, pages 1393–1400. MIT Press, 2006.

[22] J. Wolpaw, N. Birbaumer, D. McFarland, G. Pfurtscheller, and T. M. Vaughan. Brain-Computer Interfaces for Communication and Control. *Clinical Neurophysiology*, 113(6):767–791, 2002.

[23] H. A. Yanco. Wheelesley, A Robotic Wheelchair System: Indoor Navigation and User Interface. In Vibhu O. Mittal, Holly A. Yanco, John M. Aronis, and Richard C. Simpson, editors, *Assistive Technology and Artificial Intelligence*, Lecture Notes in Artificial Intelligence, pages 256–268. Springer-Verlag, 1998.