Robotics and Autonomous Systems 58 (2010) 1246–1255

Contents lists available at ScienceDirect

Robotics and Autonomous Systems

journal homepage: www.elsevier.com/locate/robot

Brain-coupled interaction for semi-autonomous navigation of an assistive robot[☆]

Xavier Perrin[a,*], Ricardo Chavarriaga[b,**], Francis Colas[a], Roland Siegwart[a], José del R. Millán[b]

[a] Autonomous Systems Laboratory, ETHZ, Zürich, Switzerland

[b] CNBI, Center for Neuroprosthetics, Ecole Polytechnique Fédérale de Lausanne (EPFL), 1015 Lausanne, Switzerland

ARTICLE INFO

Article history:
Available online 19 June 2010

Keywords:
Semi-autonomous navigation
Assistive robotics
Low throughput user interface
Bayesian programming
Brain–computer interface
Neurorobotics
EEG
Error-related potentials

ABSTRACT

This paper presents a novel semi-autonomous navigation strategy designed for low throughput interfaces. A mobile robot (e.g. intelligent wheelchair) proposes the most probable action, as analyzed from the environment, to a human user who can either accept or reject the proposition. In the case of refusal, the robot will propose another action, until both entities agree on what needs to be done.

In an unknown environment, the robotic system first extracts features so as to recognize places of interest where a human–robot interaction should take place (e.g. crossings). Based on the local topology, relevant actions are then proposed, the user providing answers by means of a button or a brain–computer interface (BCI). Our navigation strategy is successfully tested both in simulation and with a real robot, and a feasibility study for the use of a BCI confirms the potential of such an interface.

1. Introduction

Robots are present more and more in our daily life, not only in industry but also at home as toys or as service robots like vacuum cleaners or intelligent wheelchairs. They can be either controlled by the human or autonomously perform a predefined task. Shared-controlled, semi-autonomous and autonomous techniques have been proposed, but they are often not adapted to low throughput devices made for disabled or elderly people, such as sip and puff systems, single switches or even the electroencephalogram-based (EEG) brain–computer interfaces (BCI).

In this paper, a novel semi-autonomous navigation strategy is proposed, with the aim of minimizing the user involvement. Instead of requiring user control commands at each step, the robot proposes actions (e.g. turning *Left* or going *Forward*) based on environmental information (Fig. 1). The human monitors the activity of the robot, rejecting propositions he disagrees with. Given human rejection, the robot has to take a different decision based on this additional information. The human workload is reduced to a yes/no input instead of having to select among possible actions. In addition, human–robot interaction only takes place at automatically recognized key locations. These characteristics make our approach suitable for low throughput devices.

Besides testing the system using a button interface, we performed a preliminary study of the feasibility of using BCI systems that exploit error-related EEG signals (ErrP); these signals indicate human awareness of an erroneous system decision [1,2]. With respect to our application, the ErrP is a convenient input for paralyzed persons, as we could exploit the brain as an intuitive communication channel. Nevertheless, due to the complex nature of the brain signals in general, such interfaces are prone to classification errors and the decoded commands thus require adequate processing in our robotic system.

Moreover, efficient navigation in real environments requires the robot to be able to deal with noisy information captured by its sensors or delivered by the user interface, while inferring the most appropriate actions to take. In order to cope with these requirements, we develop a robotic controller that relies strongly on Bayesian reasoning techniques.

As exposed above, we have a semi-autonomous artificial agent, the robot, which analyzes its sensory data in order to extract the required information for a proper human–robot interaction. The resulting dialog is then composed of action propositions which requires only high-level "binary" answers, as opposed to low-level control commands encountered e.g. in common shared-controlled systems. Furthermore, the use of an EEG-based BCI allows to extract the user intended answer directly from his brain activity.

The structure of the paper is the following. Section 2 details the semi-autonomous navigation system from the features extraction

[☆] The authors thank Dizan Vasquez for providing the Morsel simulation environment and for useful discussions. This work has been supported by the EC-contract number BACS FP6-IST-027140. This paper only reflects the authors' view and funding agencies are not liable for any use that may be made of the information contained herein.

[*] Corresponding author.

[**] Corresponding author.

E-mail addresses: xavier.perrin@a3.epfl.ch (X. Perrin), ricardo.chavarriaga@a3.epfl.ch (R. Chavarriaga), francis.colad@mavt.ethz.ch (F. Colas), rsiegwart@ethz.ch (R. Siegwart), jose.millan@epfl.ch (J.d.R. Millán).

0921-8890/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.robot.2010.05.010

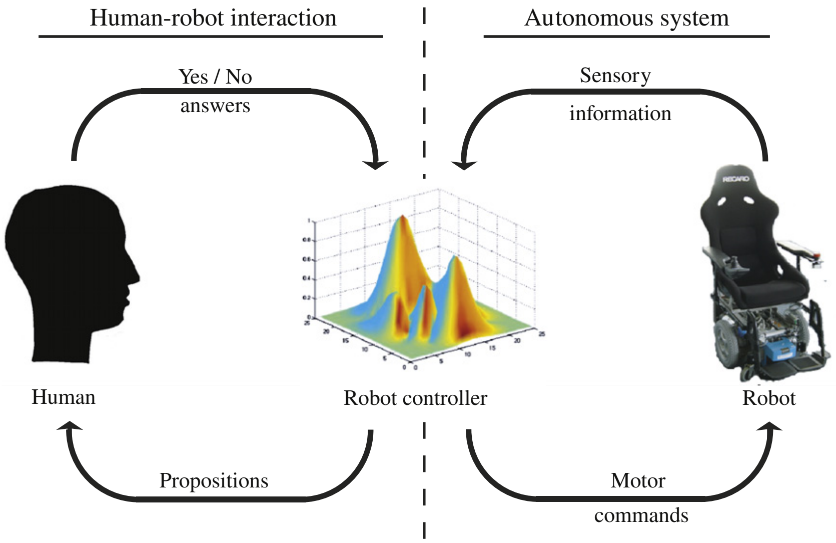

Fig. 1. Overview of our semi-autonomous navigation system, with two loops in interaction. On the right, the autonomous robot evolving safely in the environment, but without knowing the user intended goal. On the left, the human–robot interaction in order to select which action to be executed at a given place.

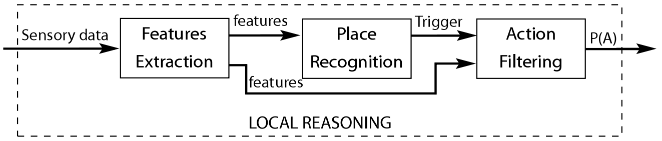

Fig. 2. Semi-autonomous navigation (SAN) system: details of the processing of the local sensory data, composed of three sub-tasks: the *features extraction*, the *recognition of places of interests* such as crossings or door openings, and the computation of a first probability distribution over the possible actions $P(A)$ (*action filtering*).

to the inference of the probability distribution over the possible actions. Then, the human–robot interaction used for action selection is described in Section 3. Section 4 presents the results of experiments both in simulation and with a real robotic platform, as well as the preliminary study with the BCI interface. Finally, we conclude this paper in the discussion of Section 5.

2. Semi-autonomous navigation

Aiming at minimizing the required user involvement, we developed a semi-autonomous approach where the robot evolves autonomously most of the time, only requiring user inputs when a navigational decision has to be made, e.g. when reaching a crossing. This is illustrated in Fig. 1, where the intelligent system – the robot controller – supervises two distinct loops. On the right part, the autonomous part of the system is depicted, where the sensory values are processed in order to compute coherent motor commands. On the left part, the human–robot interaction loop shows our intended dialog mechanism between the human and the machine: the system will make propositions to the user, e.g. Turning left or right, who will then either agree or disagree.

In our semi-autonomous navigation (SAN) approach, a module termed *Local reasoning* (Fig. 2) is in charge of analyzing the robot sensory input in order to recognize places where a navigational decision has to be made (e.g. corridor crossing). Once an interesting place has been reached the probability distribution $P(A)$ of possible actions is inferred by the *Action filtering* module. This distribution reflects the robot analysis of the unknown surrounding environment, where actions leading toward open spaces are more probable than actions leading to an obstacle. In this implementation the possible actions are *Stop*, *Turn Right*, *Move Forward*, *Turn Left*, and *U-Turn*.

Once the action probability distribution has been computed, a *Human–Robot dialog* is established in order to choose the action intended by the user (see Section 3). That action is then executed and the robot resumes autonomous behavior until a new interaction is required. This section describes the robotic platform used to test the proposed approach, and then we introduce the modules presented in Fig. 2. Starting from the features extraction and place recognition to the computation of an initial probability distribution for initializing the dialog with the user.

2.1. Experimental setup

We test the developed system on a robot named BIBA and its simulated equivalent. The BIBA robot (Fig. 3(a)) is a differential-driven robot built by the Bluebotics company (http://www.bluebotics.ch). The only used sensor is a SICK laser range finder (http://www.sick.com), providing robot-centric measurements on 180° (361 measurements). The provided robot software is used for implementing the navigation methods and automatic obstacle avoidance, so that our system only provides local navigation sub-goals (x, y, θ) location in a robot-centric coordinate system. This emphasizes the fact that the proposed approach for semi-autonomous navigation is independent of the actual robotic platform and motion planning implementation.

In addition to using the real robot, we also performed tests in a simulation platform endowed with sensor emulation capabilities. This allow us to extensively test different system configurations and experimental conditions. For our work, we relied on an in-house simulator named "morsel", developed at ETHZ (Fig. 3(b)). Based on a 3D game engine, it allows to easily create a realistic environment (with static and dynamic objects) where the robot will evolve. Several sensors like laser scanners and cameras are also available, enabling to have a full robotic simulation. For the simulated navigation experiments, we rely on a simple proportional controller with saturation conditions taking advantage that the robot position in an absolute coordinate system is continuously available.

2.2. Feature extraction

Based on the distance measurements coming from the laser scanner (Fig. 4 first column), features are extracted for two different purposes: first, to recognize places of interest where

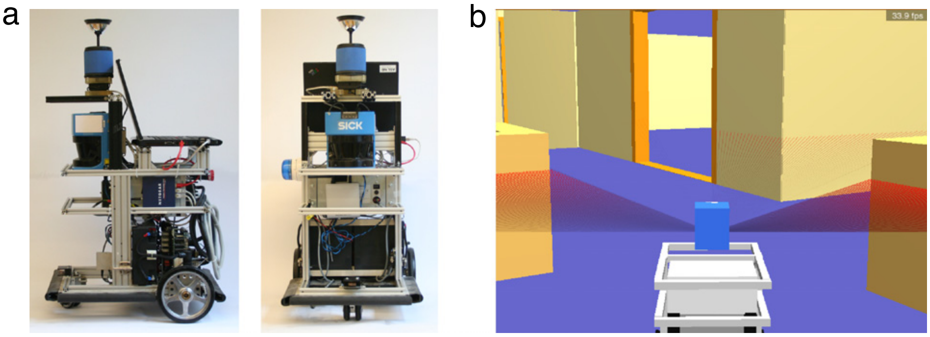

Fig. 3. (a) BIBA robot. (b) Morsel simulator.

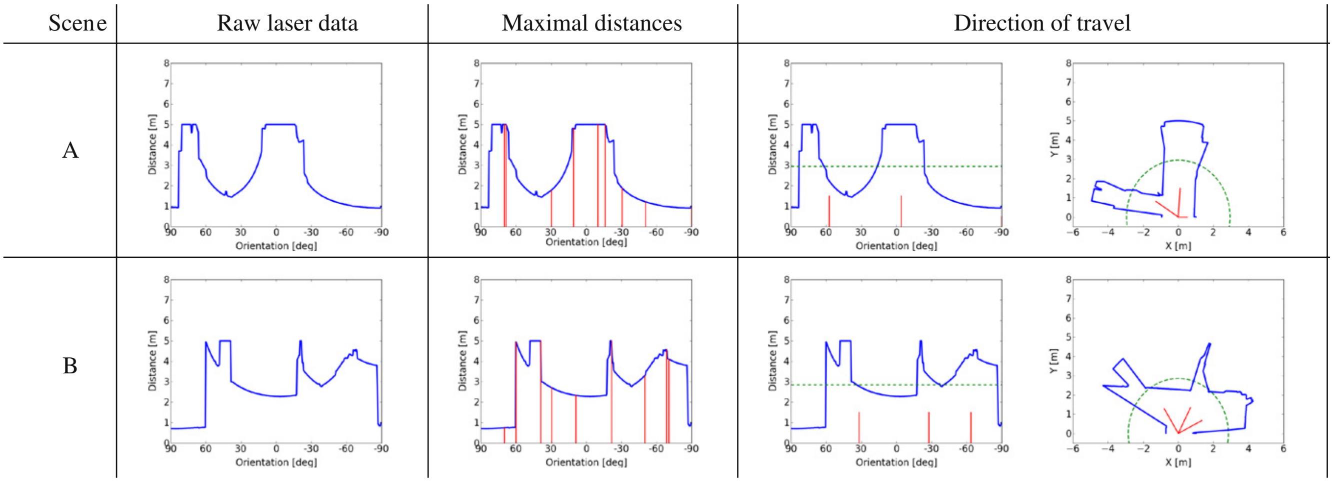

Fig. 4. Examples of feature extraction in a polar view. Left column, the raw laser data. Middle and right columns, the maximal distance in 9 sectors and, respectively, the directions of travel (also in Cartesian coordinate). The green dotted line in the third column indicates the threshold value for extracting the openings.

human–robot interaction is required and, second, to identify possible directions of travel.

For place recognition, the sensory data on 180° are divided in N sectors of equal size and the maximal distance in each sector is extracted (Fig. 4 second column). This allows to find all the openings in the environment in front of the robot (the explanation of this choice as well as the optimal number N of sectors will be discussed in Section 2.3). Besides extracting the maximal distances, the minimal measured distance is also stored for safety issues (see Section 2.4 for details).

Concerning the navigation features, we need to find the directions of travel corresponding to the three commands "Right", "Forward", and "Left".[1] First, we extract the openings present in the environment by searching for the peaks in the polar representation of the distance measurements (Fig. 4 third column, portion above the threshold value). As the robot could encounter narrow cluttered environments or wide obstacle-free ones, a fix threshold method would not be appropriated. Some empirical tests showed that the mean between the minimal and maximal measured distances is a good compromise for extracting the desired directions without being stuck into a local maximum.

Then, we remove the openings being too narrow for the robot and assign the remaining directions to the corresponding navigation commands. If there is no direction associated to one of these commands, the direction corresponding to the minimal measured distance in the related sector is used (making the robot face the nearby obstacle). If there are more than one direction associated to a command, the direction pointing the furthest away is chosen. Finally, we check for possible collision and, if necessary, modify the orientation accordingly. The final result is shown in Fig. 4 in the rightmost column.

We also have to define an associated distance to travel to each navigational command. Besides ensuring a minimal security distance to the obstacles, the distances are also bounded to a maximal value of 1.5 m, which the robot travels autonomously.[2] This comes from the need to pass a crossing without triggering unnecessary interactions with the user (topology recognition turned off) as the local topology changes dramatically with the movement, especially when turning right or left.

2.3. Place recognition

The next step in our SAN system is to recognize places of interests where an interaction with the human is required, such as crossings. We applied the same approach as in the work of Tapus et al. [3] to detect interesting topologies ranging from corridors to X-crossings, through the different L- and T-crossings and dead ends. Therefore, we developed a Bayesian model in order to recognize the different topology types given the N maximal distances features. In the Bayesian Programming (BP) formalism [4], the probabilistic relations between these distance features $F_{1\ldots N}$ and the topology classes T are described with

[1] For the commands "Stop" and "U-Turn", the position of the robot remains the same, the only possible movement being to turn on spot in case of a 'U-Turn' instruction.

[2] This value was found to be adequate for an indoor, furnished environment.

Table 1

Results of the topology recognition for the classifier with 9 features. Corridor (‖), left *L*-(╗), right *L*-(╔), left *T*-(╣), right *T*-(╠), front *T*-(╦), and *X*-crossing (╬), dead end (╥).

	‖ (%)	╗ (%)	╔ (%)	╣ (%)	╠ (%)	╦ (%)	╬ (%)	╥ (%)
‖	94.9			3.6	1.2			0.3
╗	0.1	99.2				0.7		
╔		0.1	99.1			0.8		
╣	0.3		0.1	98.5	0.4		0.7	
╠				0.5	95.0		4.5	
╦		0.1			0.1	99.8		
╬				3.2	2.2	0.1	94.5	
╥							0.1	99.9

the following joint probability decomposition: $P(F_{1\dots N}\ T) = P(T)\prod_{i=1}^{N} P(F_i \mid T)$. The prior $P(T)$ is set to a uniform distribution and the $P(F_i \mid T)$ are Gaussian distributions learned using training data. Then, the Bayesian model computes $P(T \mid F_{1\dots N})$, i.e. the most probable topology class given the extracted features.

The optimal number N was determined experimentally by testing this parameter in the range from 1 to 20. In a simulated environment, the robot was randomly positioned and oriented in front of a variety of crossings in order to gather learning data for estimating the different $P(F_{i,N=n} \mid T)$ and then to test the resulting classifier. This random positioning occurred in a rectangle of 1.3 × 1.8 m (depth × width) before the crossing.[3] The orientation varied of ±20° from the corridor's central direction. Both the training and testing sets contained around 700 laser scan data for each topology type.

We then compare the performance of the classifier for the different number of features taking into account not only the correct classifications, but also the possible effects of misclassifications. For example, a wrongly classified topology type may prevent the system to go to a given free traversable direction in reality. Such a case would have thus a major impact on the performance of a classifier. On the contrary, allowing a possible sub-optimal action may have a smaller impact in performance. For this, we considered a comparison method were wrong recognitions are assigned a negative cost as described here:

$$P(\text{Correct}) = \sum_{T} P(T) \sum_{T'} P(T' \mid T)\, P(\text{Correct} \mid T\,T')$$

$P(T)$ is the distribution of the testing data for each topological class (appearance frequency); $P(T' \mid T)$ is the obtained confusion matrix in the testing set (e.g. Table 1). $P(\text{Correct} \mid T\,T')$ is the cost matrix used for weighting the different classifier outcomes. In our situation of topology recognition, we gave a positive score of +1 for a correct classification. For incorrect classifications, negative scores are given as follows: −1 for each action we should have been able to execute given the true topology but which was not available for the inferred topology, and −0.5 for each action we could wrongly do according to the inferred topology as compared to the true one. For example, a right T-crossing classified as a left T-crossing would have a reward of −1.5, given that we cannot go on the right (−1) but are wrongly able to go left (−0.5).

This performance metric shows that 7 features or more yield good, stable classification performances. Peak performance was obtained in the case of 4, 9, and 15 features (performances of 96.19%, 95.80% and 95.81%, respectively). We chose a topology classifier relying on 9 features, as a good compromise between 4 and 15 features. Given the fact that we trained the system in an ideal simulated environment, 4 features may not be able to cope with the complex distance measurements in real, cluttered environments, while using 15 features may lead to overfitting.

Table 2

Examples of relations between the topology type and the actions $P(A \mid T)$.

$P(A \mid T)$					
Stop	Right	Forward	Left	U-Turn	T
0.06	0.01	0.80	0.01	0.12	‖
0.06	0.01	0.01	0.80	0.12	╗
0.06	0.01	0.40	0.40	0.13	╣
0.04	0.28	0.28	0.28	0.12	╬
0.50	0.01	0.01	0.01	0.47	╥

The classification accuracy for the classifier with 9 features is shown in Table 1. Correct performance is above 95% in most cases, the lowest one being 94.5% for the X-crossing. The navigation could only be strongly affected in the two cases where a place of interest is misclassified as a corridor, but they occur rarely (less than 0.5%). Furthermore, these situations are supposed to rapidly change due to the robot motion, resulting in a classification other than a corridor.

Now that the system is able to recognize the topological nature of the environment, the corresponding actions at a topological change have to be deduced. This is done in the next processing step, which is called once the system recognizes with a clear confidence a new topology type (i.e., when the probability difference in the two most probable topology types exceeds 0.5).

2.4. Action filtering

The final purpose of the local reasoning system is to get an initial probability distribution over the actions the robot could execute, so as to engage a dialog with its user. Given the previously recognized topology, this probability distribution could be easily obtained by relating each possible crossing topology t with a given distribution $P(A \mid T = t)$, filtering out unprobable actions. By doing this, we would break up the uncertainty propagation from the place recognition stage to the later processing steps, as we would maintain only the value with the maximal probability. Such a simplification might also discard valuable information, as the second most probable topology could be related to alternative actions.

Nevertheless, in the action filtering step, we take into account possible topology misclassification as well as some constraints related to security issues. From the place recognition step, we obtain the probability distribution $P(T)$. Then, the topology is related to the actions A through the probability table $P(A \mid T)$, given *a*-priori, as partly illustrated in Table 2.

Concerning safety issues, we constraint the system to either stop or do a U-turn if the minimal measured distance to an obstacle is below a given safety distance. In probabilistic approach, this is done by adding a constraint variable C which takes the value of 1 if the minimal distance $D_{\min}$ is above a fixed security distance. Depending on the value of C, the actions will either reflect the ones predefined in Table 2 or the stop/U-turn alternatives.

We could have build a specific Bayesian program for our action filtering procedure, having the following joint probability distribution,

$$P(T\,A\,D_{\min}\,C) = P(T)\,P(D_{\min})\,P(C \mid D_{\min})\,P(A \mid T\,C).$$

However, instead of having a separate program relying on $P(T)$, we included the different terms we presented here into the place recognition step. Fig. 5 shows the complete Bayesian program

[3] For our SAN system, it is desirable to recognize a new place of interest before having gone too far through it so as to better plan a motion sequence given the specific kinematics of the robot. Therefore, we did not placed the robot in the middle of a crossing.

$$
\text{Program}\begin{cases}
\text{Description}\begin{cases}
\text{Specification}\begin{cases}
\text{Relevant variables:}\\
\quad F_{1\,\ldots\,N}:\ \text{the features in } N \text{ sectors}\\
\quad T \in \{\text{‖},\text{╗},\text{╔},\text{╣},\text{╠},\text{╦},\text{╬},\text{╥}\}:\ \text{the different}\\
\qquad \text{topology types}\\
\quad A \in \{S,R,F,L,U\}:\ \text{the actions}\\
\quad D_{min}:\ \text{minimal measured distance}\\
\qquad \text{to obstacles}\\
\quad C \in \{0,1\}:\ \text{safety constraint}\\
\text{Decomposition:}\\
\quad P(F_1 \ldots F_N\, T\, A\, D_{min}\, C) =\\
\quad P(T)\, P(D_{min})\, P(C \mid D_{min})\\
\quad \prod_{i=1}^{N} P(F_i \mid T)\, P(A \mid T\, C)\\
\text{Forms:}\\
\quad P(T),\ P(D_{min}):\ \text{uniforms}\\
\quad P(F_i \mid T):\ \text{Gaussians}\\
\quad P(C \mid D_{min}) = \begin{cases} 1 \text{ if } D_{min} > D_{security}\\ 0 \text{ otherwise}\end{cases}\\
\quad P(A \mid T\, C) = \begin{cases} P(A \mid T) \text{ if } C = 1 \text{(table 2)}\\ P(A) = \begin{cases} 0.5 \text{ if } A \in \{S,U\}\\ 0.0 \text{ otherwise}\end{cases}\end{cases}
\end{cases}\\
\text{Identification: Gaussians learned based on}\\
\quad \text{experimental data, probability table given,}\\
\quad \text{other parameters fixed a-priori}
\end{cases}\\
\text{Questions:}\\
\quad P(T \mid F_1 \ldots F_N),\ P(A \mid F_1 \ldots F_N\, D_{min})
\end{cases}
$$

Fig. 5. Bayesian program for both place recognition and action filtering.

expressed in the BP formalism. As can be seen, besides the inference of the topology type, the new question we ask to the program is $P(A \mid F_1 \cdots F_N\, D_{\min})$, i.e. the probability distribution over the actions given the extracted features and the minimal distance. There is thus a marginalization step over the topology types T.

3. Human–robot interaction

If two or more actions have a similar probability to be executed, the robot cannot know which one fits better the human's navigation intention. There also could be the case when the robot has a wrong belief about the actions (e.g. the user wants to stop in front of a wall to look at a painting). Therefore, the human is involved in the control loop by monitoring propositions made by the mobile robot.

A dedicated process is in charge of selecting which action will be proposed to the human based on the action probability $P(A)$ (from *Action Filtering*). It further interprets the user inputs so as to update $P(A)$. The human–robot dialog continues until an action is likely enough as to trigger its execution by the robot.

An important issue is how to provide feedback to the user about the proposed action. In a previous study we compare several types of feedback principles used for proposing actions to the user [5]. The three major human modalities were covered (sight, hearing, touch) and two kinds of user interfaces were tested (keyboard and BCI). Visual feedback was found to be a good feedback modality both from psychophysics measures as well as elicited brain signals. Furthermore it allows its extension to a large number of possible propositions.

Complementary, the user has to provide an answer to these propositions, e.g. "Yes" or a "No". Common specialized interfaces such as single or multiple switches, or sip and puff systems can be directly used. Interfaces with a richer output, such as joysticks or eye/gaze tracking systems, may be used as well by first processing their output in order to assign a given subregion of their working space to a particular command. Potentially, detection of error-related evoked EEG signals elicited in the user's brain in case of disapproval may be also used to this purpose [2,1].

Table 3
Confusion matrix of the classification of the error potential (ErrP) taken from [1]. I_k stands for the real user intended response and O_k for the observed signal classification (EEG decoded response) at time k. Note the asymmetry of the confusion matrix.

$P(O_k \mid I_k)$	$O_k =$ no ErrP (%)	$O_k =$ ErrP (%)
$I_k =$ no ErrP	92.0	8.0
$I_k =$ ErrP	26.5	73.5

A simple dialog management strategy can be used if an ideal interface is used, e.g. by proposing alternative actions as long as the user answers "No". However, particular attention has to be taken if the interface is prone to errors, as is the case of BCI systems. In this case, the actual user intend has to be inferred during the whole dialog process, possibly composed of multiple propositions and answers. Inspired from studies in spoken dialog systems, we designed a Dynamic Bayesian Network (DBN) in order to track the user intended action [6]. The proposed action, the user answer obtained from the user interface, as well as the interface accuracy are the three inputs to the DBN. The interface accuracy is represented as a probability table in the form of $P(O_k \mid I_k)$, i.e. the probability that the interface observes a human answer O_k at step k given the true intended answer I_k (Table 3). It serves to correct the user answer. This adapted user answer, together with the proposition, allows then for an adequate update of the initial probability distribution over the actions $P(A)$.

Then, a strategy is required for selecting which action to propose and when the system has to stop the dialog and proceed to the execution of the selected action. Several strategies were tested during a user study, both from a theoretical and a user point of view. Unsurprisingly, the preferred solution was to simply propose the most probable action, as it is more logical for the user. Executing an action after a single acknowledgment would be a risky strategy if an uncertain interface such as a BCI is used. We thus wait until the probability difference between the two most probable actions exceeds a given threshold.

4. Results

We perform several experiments with the simulated and real robot in order to evaluate the ability of the proposed system to efficiently navigate based on our human–robot interaction framework. We assess the performance of the place recognition process as well as the interaction mechanism. Moreover, preliminary results of the EEG potentials evoked during navigation are also presented.

4.1. Simulations

We evaluated the system performance in a navigational task for different conditions. In all cases, the goal is to take the robot from a start position (S) to a goal location (G). The environment was designed so that a place could be reached through different paths, which is useful for simulating unforeseen situations such as blocked passages (Fig. 6). The goal of these experiments is to show the performance of the SAN system evolving in unknown environments (i.e. the ability to identify locations where an action has to be selected) and to assess the benefit of our Bayesian-based action selection process. For the latter case, two distinct parameters will be varied: the actual interface accuracy and the knowledge the SAN system has about the interface accuracy (i.e. the $P(O_k \mid I_k)$ distribution). For proper assessment of these

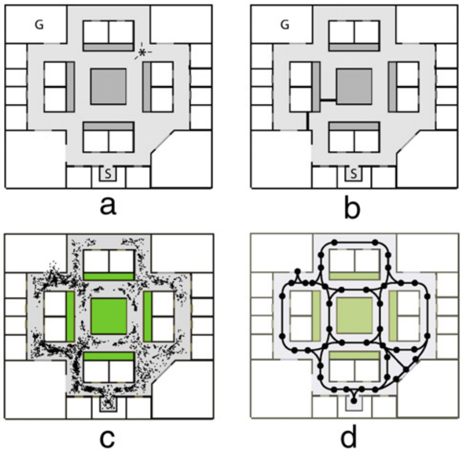

Fig. 6. Simulated environment. In white, the possible rooms; in light grey, the traversable areas; in dark grey, obstacles (plants, holes, etc.); S, the start position; G, the place to be reached; and * an example of crossing with non-perpendicular directions. Environment with (a) all paths allowed and (b) with some corridors blocked (represented by a thick black line). (c) Places where a topological change was automatically identified across all the conditions. (d) Manually built topological map, where nodes correspond to the places of interest.

Table 4
Conditions tested in simulation. *TP*, True Positive, and *TN*, True Negative, rates represent the percentage of correct classification for both error and correct classes. *BF knowledge* is the information the Bayesian filter has about the accuracy of the input system ($P(O_k \mid I_k)$), whereas the simulated interface accuracy is shown in the *Accuracy* column. The last column shows the success rate (goals reached) for each condition.

	BF knowledge		Accuracy		Success rate
	TP (%)	TN (%)	TP (%)	TN (%)	
1	Autonomous behavior				18
2	100	100	100	100	100
3	C2 $+$ blocked corridors (Fig. 6(b))				100
4	100	100	92	73.5	100
5	100	100	75	75	94
6	100	100	50	50	10
7	92	73.5	92	73.5	100
8	92	73.5	85	65	100
9	92	73.5	75	90	100

conditions we simulate the human–robot interface for different performance levels.

For the given task, the shortest path in terms of decision points requires 6 human–robot dialogs with an ideal system, which would recognize perfectly each crossing and would navigate autonomously between them. If the robot was unable to reach the goal in less than 30 steps, the trial was stopped and labeled as a failure. Each experiment was composed of 50 trials.

Fig. 6(c) shows the places where the system triggered an interaction with the human user, i.e. each dot corresponds to a topological change as described in Section 2.3. As expected, they are largely located around crossings and corners with a higher density in the left part of the environment due to the trajectories taken by the robot (see below). Comparison of these points – identified automatically by the system – with an ideal topological map of the environment (Fig. 6(d)) confirms the good performance of the place recognition module.

We simulated nine experimental conditions as summarized in Table 4. They can be separated in three different groups. The first three conditions compare the contribution of the user in the control loop. The following three conditions compare different levels of reliability of the human–robot interface, for a system that assumes a perfect interface. Finally, the last three conditions emulate a system that takes into account a predefined HRI reliability. In addition to the success rate, the robot trajectories for all conditions are shown in Fig. 7.

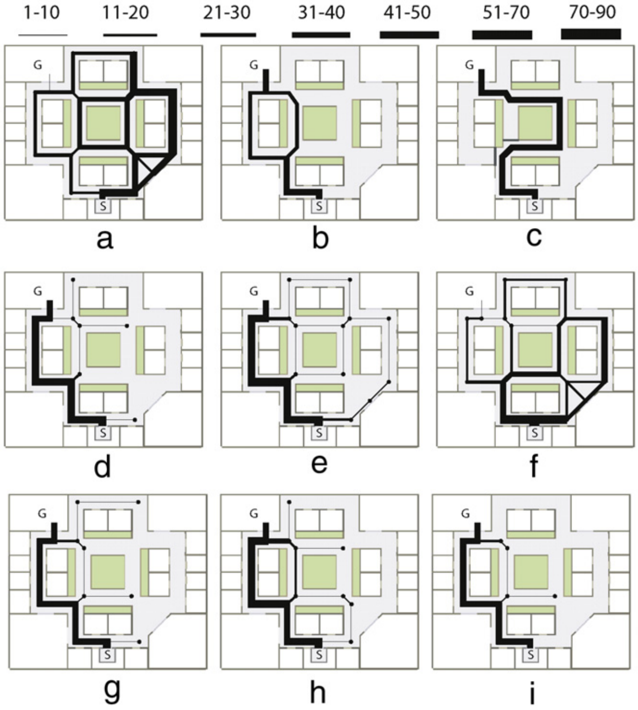

Fig. 7. Robot trajectories for the different input interface accuracies. The thickness of the line represents the number of times the robot crossed a particular segment, as shown on top. (a) Robot driving autonomously (C1). With no specific filter knowledge about the input system accuracy: (b) Ideal user interface (C2); (c) Ideal user interface, blocked scenario (C3); (d) EEG interface (C4); (e) 75% interface (C5); (f) Random interface (C6). Last three plots show performance of a Bayesian filter aware of the performance of the EEG interface: Simulated EEG performance (g) case I (C7); (h) case II (C8); (i) case III (C9).

Condition 1 corresponds to the autonomous navigation of the robot without user's input. In this case, the robot selects the most probable action at each place of interest according to the action probability distribution $P(A)$. Condition 2 uses an ideal user interface that allows to correctly select the desired action with a minimal amount of interactions. Condition 3 uses the same user interface but two passages of the environment are blocked (Fig. 6(b)) thus forcing the user to take a detour. As shown in the last column, autonomous navigation (condition 1 — C1) only reached the goal room in 18% of the time, since the robot moves around the whole environment without knowing the goal location (Fig. 7(a)).[4] In contrast, when the user input is available – indicating the direction to the goal – the system achieves 100% accuracy in both conditions 2 and 3, being able to reach the goal through the shortest available paths.

Then, the influence of the interface reliability is addressed in conditions 4 to 6. To that purpose, different interface accuracies are simulated: the reported EEG performances from [1] (C4), then a 75% condition (C5), and finally a random interface (C6). For these situations however, the SAN system assumes a perfect interface so as to focus our attention on the impact of the actual interface on the navigation performance. The Bayesian filter knowledge about the interface, i.e. the $P(O_k \mid I_k)$ distribution, will therefore describe

[4] Moreover, the robot has a tendency to move towards the right. This phenomenon has its origins in the selection process of the most probable action: when two actions have the same probability, the first one in the tuple (Stop, Right, Forward, Left, U-Turn) is selected.

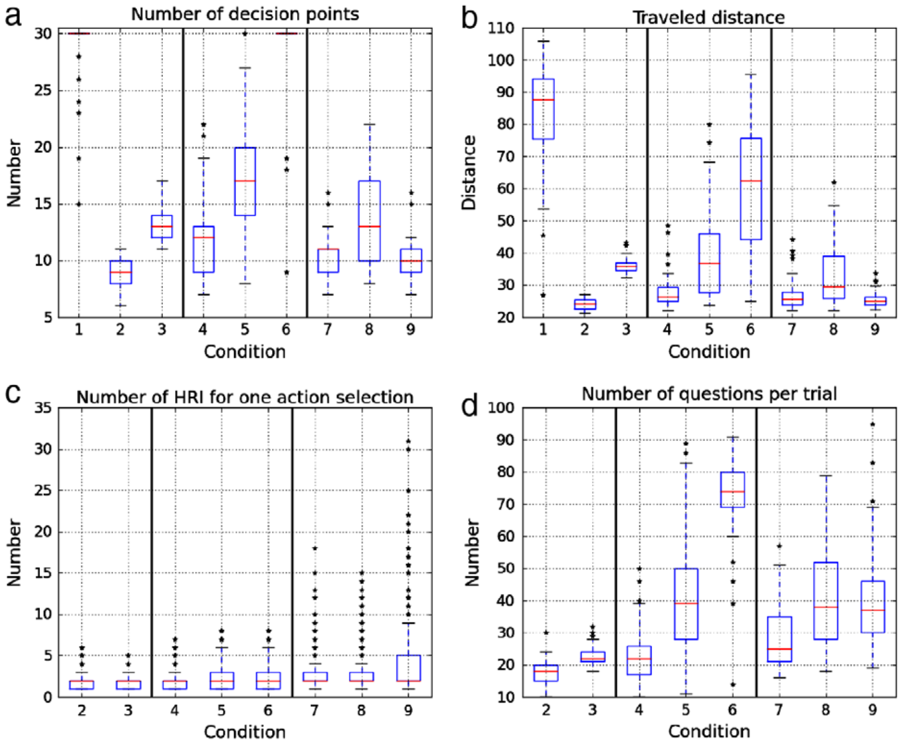

Fig. 8. (a) Number of recognized places of interest, where a navigational decision had to be made (equivalent to the number of steps). (b) Total traveled distance. (c) Number of questions asked to the user in order to select an action. (d) Total number of questions asked to the user in order to go from the start location to the goal destination. The first condition (autonomous robot) was not considered for (c) and (d). Boxplot representation, with stars indicating outliers.

a 100% correct classification of the user signals. Although the accuracy of the input system was diminished, perfect performance was achieved in condition 4, while condition 5 reached 94% of success. In these conditions, the user was able to mainly keep the robot on the shortest path, having sometimes to make a detour or to recover from a wrongly selected action (see the thin lines in Fig. 7(d)). Finally, a random interface (C6) leads to equally random trajectories thus resulting in the lowest performance, 10%. This condition is also worse than the autonomous motion (C1), as often opposite actions were successively selected (e.g. Forward followed by a U-Turn).

The last 3 experiments also emulate the use of a brain–machine interface, but the Bayesian controller takes into account the reported accuracy of the BCI system ($P(O_k \mid I_k)$ distribution, taken from [1]). Condition 7 simulates the case where the BCI accuracy corresponds to the controller's settings, while the remaining two conditions emulate different accuracies to simulate intra-subject variability of the brain signals.[5] In condition 8, the accuracy decreases for both recognizing the presence and the absence of the ErrP signal. In condition 9, a "No" is better recognized than a "Yes". Perfect task performance is achieved in the three conditions, with small differences in the traveled paths (Fig. 7(g)–(i)).

The same trends are observed in Fig. 8. It shows the number of places where human–robot interactions took place, the total distance traveled by the robot, the number of questions asked at each place of interest, as well as along the whole trajectory. It can be seen that for conditions 2 and 3, similar number of questions were asked in order to select one action. Given the increased path length of C3 due to the blocked corridors, the number of decision points is higher.

For condition 4, the number of questions before an action is selected is similar than the two previous conditions. However, given the lower interface accuracy, wrong actions are selected, resulting in longer paths and, accordingly, more encountered decision points. Then, more single questions are asked during conditions 5 and 6, as the user is less accurately understood. This of course leads to an increasing number of wrong selected actions, with the longest paths corresponding to condition 6.

In the last three conditions, where the Bayesian filter had an estimation of the accuracy of the input system, the number of questions asked to the user is similar to all previous conditions, except for some unusual situations where more than 10 successive questions appeared. Nevertheless, the paths traveled were optimal in nearly all cases as can be confirmed by the number of decision points and the traveled distance (conditions 7 and 9 similar to the ideal condition 2). The smaller interquartile range of C9 with respect to C7 when looking at the number of decision points suggest that a better recognition of a "No" than a "Yes" answer has a positive impact on the path length, as the selection of a wrong action is more easily prevented. However, this is at the cost of significantly more single questions. Condition 8, where the user interface had an important decrease in accuracy, performed worse than C7 and C9, as more incorrect actions were selected. But it is still better than C5 (75% interface accuracy with no knowledge from the Bayesian filter) in terms of traveled paths.

Comparing conditions 7–9 to condition 4 (simulated EEG interface with, resp. without the corresponding knowledge of the Bayesian filter), the benefit of providing the Bayesian filter with the mean accuracy of the input interface resides in the more directed navigation towards the goal destination (note also the line thickness difference between Fig. 7(g)/(i) and (d)). This improvement is at the cost of more human–robot interactions, as more evidence is required from the system in order to select an action.

[5] Note that BCI performance may vary due to changes in the experimental protocol, and feedback variations.

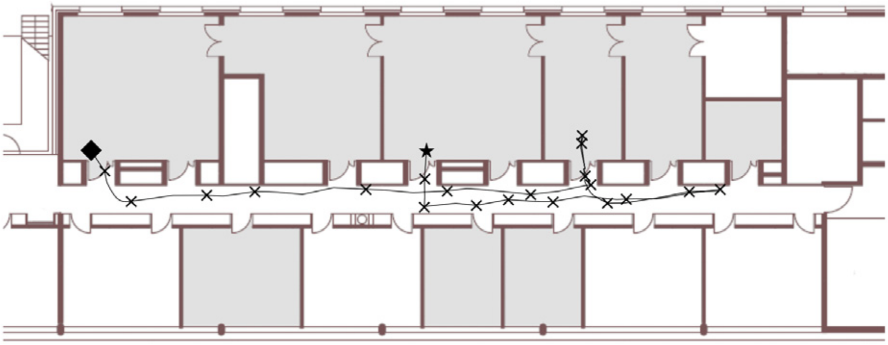

Fig. 9. One example of path driven by the real robot in an office environment. The robot started at the position indicated by the star, went along the corridor and visited two other offices. It ended at the position indicated by a diamond. The grey offices had their doors opened.

4.2. Real robot navigation

With the BIBA robot, we conducted one experiment in an office environment, with a scenario involving the visit of three different rooms. We registered both the traveled path and the locations where a new topology type was recognized, and thus where a human–robot interaction took place.

Fig. 9 shows one path of the robot. As can be seen, the robot successfully completed its task by first going out of the starting room, moving towards the end of the corridor, turn back and then visiting two other rooms. Note that the robot did not went so far into the offices for safety reasons, as the offices were all furnished and as we only sense the environment on an horizontal plane at roughly half the height of the robot; moreover, people in the offices kept working normally as we performed the experiments. We test the developed system in an unmodified environment so as to assess our approach with a real platform and to validate the simulation results.

The scenario was repeated three times and leaded to the following number of decision points: 22, 28, and 30. Numerous office doors were opened (as indicated in grey in the figure). All these openings were correctly recognized by the topology classification module, leading to many decision points. The difference is related to the fact that people were moving in the corridor and some doors were temporally opened or closed. The probabilistic nature of our robotic system also leads to different solutions even if all the trials had the same initial conditions. However, it mainly consists of different trajectories or interaction locations rather than a different amount of decision points.

4.3. EEG-error-related potentials

A potential use of the proposed approach is its coupling with brain–computer interfaces that decode the error-related potentials. However, previous related BCI studies were mainly based on simple stimuli [2,1]. As a further assessment of the viability of the use of EEG signals in real conditions we performed preliminary experiments where we record EEG potentials while the user monitors the navigation of the robot in real environments, driving around the laboratory facilities, as well as in a realistic simulation (9 and 7 possible goals respectively).

The task to be solved by the subjects was to bring the robot to goal locations randomly selected by the operator of the experiment by confirming or rejecting the robot propositions. During the experiment, the brain activity of the human subjects was recorded while the robot navigates through the environment. Two sessions were performed with the simulated robot and one with the real one; at each session the user is asked to navigate to 5 and 10 goal destinations per session, respectively.

During the experiments the user remotely monitors the robot while observing a video stream provided by an on-board high quality camera. When the robot comes to a crossing, actions are proposed with a visual feedback superimposed at the center of the video image (Fig. 10). A *fake user* application (emulating the decoding of EEG signals) was providing answers to the propositions, with a predefined error rate of 20% of the time. EEG signals are recorded at 512 Hz, spatially filtered by removing the average activity across electrodes (i.e. CAR referencing) and time filtered in the range 1–10 Hz. Following previous studies, ErrP analysis is based on the activity of frontocentral electrodes elicited by the feedback onset [2,1].

Fig. 11 shows the evoked potentials obtained during both simulated and real robot navigation. For sake of comparison, the ERP reported in a previous monitoring experiment is also shown [1,7]. It can be seen that the waveforms are similar in the three cases, despite the fact that the first two involve a more complex type of interaction, as well as the processing of richer stimuli. Characteristic components of error-related potentials already identified in simpler protocols also appear in the current experiment; namely a first, small positive peak at about 200 ms followed by a larger negative peak around 260 ms and a large positive deflection peaking approx. at 320 ms.

These preliminary results show that error-related signals observed in simpler experimental protocols are also elicited during real human–robot interaction. Moreover, signal waveform is strikingly similar despite the increased complexity of the current task. More experimental recordings are currently undergoing in order to assess the online detection of these potentials and the full integration into the semi-autonomous navigation concept.

5. Discussion

In this paper, we detailed a novel approach for semi-autonomous navigation systems in unknown environments. It relies on the automatic detection of interesting navigational points and a human–robot dialog aimed at inferring the user's intended action. During this dialog, the user can confirm or reject the robot's propositions making the system suitable for low throughput interaction devices, including brain–computer interfaces. Furthermore, building on previous studies on EEG correlates of error monitoring, we show that the proposed system can take into account the reported accuracy of BCI systems in a probabilistic manner, and perform successfully. We assess the performance of the system using experiments in simulation and a real robot. Moreover, preliminary results on the EEG activity generated during semi-autonomous navigation confirm the presence of error-related signals and yield promising evidence towards their future use in online setups.

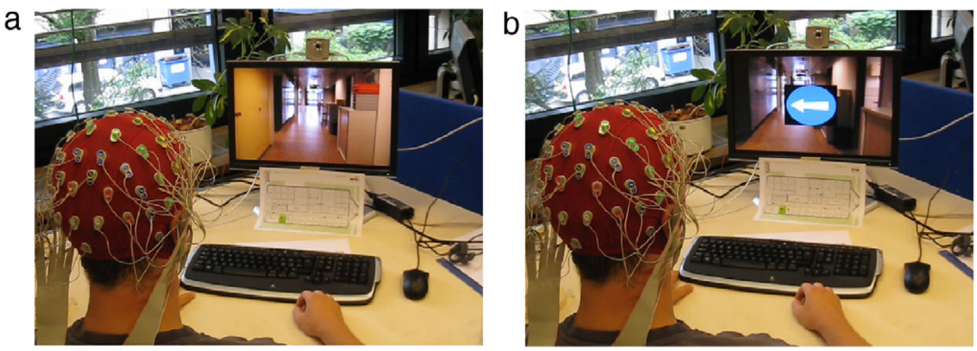

Fig. 10. Experimental setup of EEG recordings in real robot navigation. (a) The subject monitors the robot navigation through a video stream provided by an on-board camera. (b) Action propositions are superimposed on the video stream in the form of visual icons.

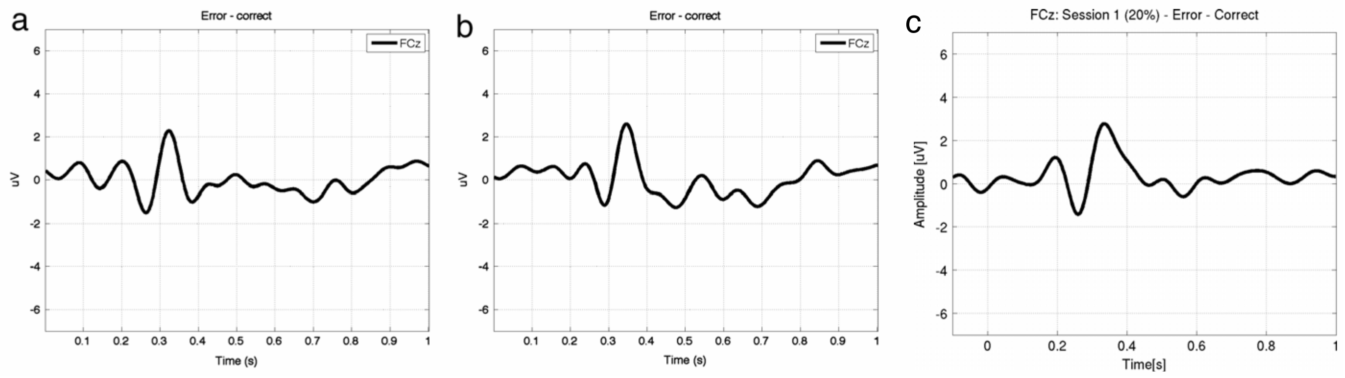

Fig. 11. Grand average ERP signals at FCz electrode (error-minus-correct condition). (a) Monitoring simulated robot. (b) Monitoring real robot. (c) Previously reported ERPs elicited by monitoring simple visual stimuli.

The basis of the whole process is the feature extraction step, which provides pre-processed data to the topology recognition algorithm as well as local directions of travel for the future selected actions. When having recognized a new place of interest, the robot stops, computes an initial probability over the actions, and engages a dialog with the human user. Once an action has been selected, the related local target location is passed to the robot motion controller for execution. The robot then navigates autonomously until a new decision place is recognized and the process starts again until bringing the user to his desired destination.

We tested each part of the local SAN system separately so as to ensure their reliability. Unfortunately, we could not compare the output of each step with a ground truth, as this was not always available. For example, the extraction of directions of travel provides satisfactory results when analyzed – subjectively – by the programmer, but there is no objective optimal output. On the contrary, we were able to analyze the topology recognition algorithm thoroughly and the results were therefore reported.

Then, we reported experiments done in simulation as well as with a real robot which mainly focused on the interaction of all parts rather than their behavior taken separately. It was shown that the SAN system is able to recognize the major places of interest, as if we would have provided a topological map, and to efficiently perform goal-directed navigation. Different user interface accuracies were tested in simulation in order to see their impact on the performance of our robotic system. For a BCI (reported performances are 92% for true positive and 73.5% for true negative classifications) or an ideal interface, the user could fulfill the tasks in nearly all the cases. This was even possible with a lower accuracy of 75% correct interpretation, with sometimes some detours, which constitutes an encouraging result towards the future use of BCI-based systems.

As compared to shared-control approaches based on constant input commands [8–10], our SAN system requires a low user involvement thanks to the high-level commands provided at key locations. Common semi-autonomous approaches [11–13] might be competitive, however with the drawback of having to provide timely precise commands in cruise control mode (risk of missing an intersection). Furthermore, the selection of a command is generally done on a predefined menu with a fixed amount of available commands, which is not always optimal. On the opposite, our approach is based on the proposition of meaningful actions, accepted or rejected by a simple "binary" signal.

As extension of this work, we are designing a system for known environments, where a map of the environment can be learned online. Additionally, the robot can learn the human habits according to contextual information such as the time of the day or the ring of a phone. With such a knowledge, the robot could propose better suited actions, or even goal destinations. We will also continue to work on the online use of the BCI interface assessing its accuracy in the real setup, as well as the required reliability according to the task demands. It would thus represent the first use of such brain signals for the steering of real robotic applications. In parallel, we will study the possibility to dynamically evaluate the need of the human user in terms of robot control, i.e. allowing to choose between the SAN system presented here or more traditional steering method (e.g. shared control) according to the particular situations. To this end, additional user signals could be required, e.g. continuous input commands, level of control, or emergency signals.

References

[1] R. Chavarriaga, P.W. Ferrez, J.del R. Millán, To err is human: learning from error potentials in brain-computer interfaces, in: R. Wang, F. Gu, and E. Shen, (Eds.), Int. Conf. Cognitive Neurodynamics, Shanghai, China, 2007, pp. 777–782.

[2] P.W. Ferrez, J.del R. Millán, Error-related EEG potentials generated during simulated brain-computer interaction, IEEE Transactions on Biomedical Engineering 55 (2008) 923–929.

[3] A. Tapus, G. Ramel, L. Dobler, R. Siegwart, Topology learning and place recognition using Bayesian programming for mobile robot navigation, in: Proc. of the IEEE/RSJ IROS 2004, vol. 4, Sendai, Japan, 2004, pp. 3139–3144.

[4] O. Lebeltel, P. Bessière, J. Diard, E. Mazer, Bayesian robot programming, Autonomous Robots 16 (1) (2004) 49–79.

[5] X. Perrin, R. Chavarriaga, C. Ray, R. Siegwart, J.del R. Millán, A comparative psychophysical and EEG study of different feedback modalities for human–robot interaction, in: ACM/IEEE Conf. on Human–Robot Interaction HRI08, Amsterdam, Netherlands, 2008.

[6] X. Perrin, Semi-autonomous navigation of an assistive robot using low throughput interfaces, Ph.D. Thesis, ETHZ, Zürich, 2009.

[7] R. Chavarriaga, J.del R. Millán, Learning from error-related potentials in non-invasive brain-computer interfaces, IEEE Transactions in Neural Systems and Rehabilitation Engineering, in press (doi:10.1109/TNSRE.2010.2053387).

[8] E. Demeester, A. Hüntemann, D. Vanhooydonck, G. Vanacker, A. Degeest, H. Van Brussel, M. Nuttin, Bayesian estimation of wheelchair driver intents: modeling intents as geometric paths tracked by the driver, in: Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS'06, Beijing, China, 2006, pp. 5775–5780.

[9] T. Gomi, A. Griffith, Developing intelligent wheelchairs for the handicapped, in: Mittal, et al. (Eds.), in: Technology and Artificial Intelligence, vol. 1458, Springer-Verlag, 1998, pp. 150–178.

[10] S. Levine, D. Bell, L. Jaros, R. Simpson, Y. Koren, J. Borenstein, The NavChair assistive wheelchair navigation system, IEEE Transactions on Rehabilitation Engineering 7 (4) (1999) 443–450.

[11] A. Argyros, P. Georgiadis, P. Trahanias, D. Tsakiris, Semi-autonomous navigation of a robotic wheelchair, Journal of Intelligent and Robotic Systems 34 (3) (2002) 315–329.

[12] H. Seki, S. Kobayashi, Y. Kamiya, M. Hikizu, H. Nomura, Autonomous/semi-autonomous navigation system of a wheelchair by active ultrasonic beacons, in: Proc. of the IEEE ICRA'00, San Francisco, 2000, pp. 1366–1371.

[13] H.A. Yanco, Shared user-computer control of a robotic wheelchair system, Ph.D. Thesis, MIT, 2000.

Xavier Perrin was born in Martigny, Switzerland, in 1982. He received his engineering degree in Microtechnics from the Ecole Polytechnique Fédérale de Lausanne (EPFL), Switzerland in 2006. Since 2006, he is a Ph.D. student at the Autonomous Systems Laboratory at ETH Zürich. His research interests span mobile robotics, Bayesian reasoning, human–computer interaction, brain–computer interfaces and robotic helpers.

Ricardo Chavarriaga is a scientific researcher at the Ecole Polytechnique Fédérale de Lausanne (EPFL), Switzerland. He received an engineering degree in electronics from the Pontifcia Universidad Javeriana in Cali, Colombia in 1998, and a Ph.D. in Computational Neuroscience from the EPFL in 2005. His work is focused on the analysis of brain electrical signals and the design of brain–computer interfaces. In particular, he is interested in the study of neurophysiological correlates of human cognitive processing and its potential use in human–machine interaction.

Francis Colas (1979) is a postdoctoral fellow at the Autonomous Systems Lab. (ETH Zürich) since January 2009. He received a Ph.D. degree in Computer Science from the INPG, in 2006. Subsequently, he joined the College de France, Paris, France as a Postdoctoral Fellow in the Laboratoire de Physiologie de la Perception et de l'Action, from 2007 to 2008. In January 2009, he joined the Autonomous Systems Lab. His research interests include Bayesian modelling of perception and action applied from cognitive sciences to robotics.

Roland Siegwart is full professor for autonomous systems at ETH Zürich. He has a Diploma in Mechanical Engineering (1983) and Ph.D. in Mechatronics (1989) from ETH Zürich. In 1989–1990 he spent one year as postdoctoral fellow at Stanford University. After that, he worked part time as R&D director at MECOS Traxler AG and as lecturer and deputy head at the Institute of Robotics, ETH Zürich. In 1996 he was appointed as associate and later full professor for autonomous microsystems and robots at the Ecole Polytechnique Fédérale de Lausanne (EPFL). During his period at EPFL he was Deputy Head of the National Competence Center for Research (NCCR) on Multimodal Information Management (IM2), coinitiator and founding Chairman of Space Center EPFL and Vice Dean of the School of Engineering. In 2005 he held a visiting position at NASA Ames and Stanford University.

Roland Siegwart is member of the Swiss Academy of Engineering Sciences and board member of the European Network of Robotics (EURON). He served as Vice President for Technical Activities (2004–2005) and is currently Distinguished Lecturer (2006–2007) of the IEEE Robotics and Automation Society. He is member the "Bewilligungsausschuss Exzellenzinitiative" of the "Deutsche Forschungsgemeinschaft (DFG)". He is coordinator of two European Projects and co-founder of several spin-off companies. His research interests include the design and control of systems operating in complex and highly dynamical environments. His major goal is to find new ways to deal with uncertainties and enable the design of highly interactive and adaptive systems. Prominent application examples are personal and service robots, planetary exploration robots, autonomous micro-aircrafts and driver assistant systems.

José del R. Millán is the Defitech Professor at the Swiss Federal Institute of Technology in Lausanne (EPFL) where he explores the use of brain signals for multimodal interaction and, in particular, the development of non-invasive brain-controlled robots and neuroprostheses. In this multidisciplinary research effort, Dr. Millán is bringing together his pioneering work on the two fields of *brain–computer interfaces and adaptive intelligent robotics*.

He received his Ph.D. in computer science from the Univ. Politècnica de Catalunya (Barcelona, Spain) in 1992, where he was an assistant professor for three years. He was also a research scientist at the Joint Research Centre of the European Commission in Ispra (Italy), a senior researcher at the Idiap Research Institute in Martigny (Switzerland), and a visiting scholar at the Universities of Stanford and Berkeley as well as at the International Computer Science Institute in Berkeley.

His research on brain–computer interfaces was nominated finalist of the *European Descartes Prize 2001* and he has been named *Research Leader* 2004 by the journal Scientific American for his work on brain-controlled robots. The journal *Science* has reviewed his work as one of the world's key research in the field of brain–computer interfaces. Dr. Millán is the coordinator of a number of European projects on brain–computer interfaces and also is a frequent keynote speaker at international events. His work on brain–computer interfaces has received wide media coverage around the world.