ESANN'2009 proceedings, European Symposium on Artificial Neural Networks - Advances in Computational Intelligence and Learning. Bruges (Belgium), 22-24 April 2009, d-side publi., ISBN 2-930307-09-9.

Augmenting information from Brain-Computer Interfaces through Bayesian plan recognition

Eric Demeester[1], Alexander Hüntemann[1],
José del R. Millán[2] and Hendrik Van Brussel[1] *

1- Katholieke Universiteit Leuven - Department of Mechanical Engineering
Celestijnenlaan 300B, bus 2420, 3001 Heverlee - Belgium

2- École Polytechnique Fédérale de Lausanne
CE 3 316 (Centre Est), Station 1, 1015 Lausanne - Switzerland

Abstract. For severely disabled people, Brain-Computer Interfaces (BCIs) may provide the means to regain mobility and manipulation capabilities. However, information obtained from current BCIs is uncertain and of limited bandwidth and resolution. This paper presents a Bayesian framework that estimates from uncertain BCI signals a richer representation of the task a robotic mobility or manipulation device should execute, such that these devices can be operated more safely, accurately and efficiently. The framework has been evaluated on a simulated robotic wheelchair.

1 Introduction

People suffering from quadriplegia such as those with a locked-in syndrome cannot control most assistive devices with conventional interfaces. A promising solution for these people are Brain-Computer Interfaces (BCIs). Unfortunately, information from BCIs arrives at a relatively slow pace, and it is usually noisy and of limited resolution. This may prohibit the efficient and safe direct control of robotic mobility and manipulation devices. This paper focuses on the interpretation of noisy BCI signals for continuous control of assistive devices. From the noisy BCI signals and an observation of the user's environment, we estimate a richer representation of the task that the user has in mind for the assistive device. This is the problem of *plan recognition* or *intent estimation* [1]. It is hoped that based on this richer task representation, the user can be helped in a safer and a more accurate and efficient way as compared to a direct execution of the noisy BCI commands.

Section 2 presents our Bayesian plan recognition framework for BCIs. In Section 3 this plan recognition framework is applied to assistive wheelchairs. Finally, Section 4 presents experimental results regarding BCI control of a robotic wheelchair in simulation.

*This work was supported by the European IST Programme FET Project Maia (FP6-003758).

ESANN'2009 proceedings, European Symposium on Artificial Neural Networks - Advances in Computational Intelligence and Learning. Bruges (Belgium), 22-24 April 2009, d-side publi., ISBN 2-930307-09-9.

2 Bayesian plan recognition using BCIs

Generally speaking, users want an assistive robot to reach a certain goal configuration $\boldsymbol{c_{goal}}$ with a certain goal velocity $\boldsymbol{v_{goal}}$. For example, $\boldsymbol{c_{goal}}$ may correspond to a desired cursor position in the case of mouse control in multimedia applications, to a desired goal position for a wheelchair, or to a desired end effector configuration for a robotic manipulator. A configuration $\boldsymbol{c}$ and velocity $\boldsymbol{v}$ will be represented jointly as the robot state $\boldsymbol{x}$. A user plan $\boldsymbol{i}_k$ at time k can then be generically described as a trajectory $\boldsymbol{i}_k = \{\boldsymbol{x}_{current}, \cdots, \boldsymbol{x}_{goal}\}$, which the user has in mind to achieve the goal state $\boldsymbol{x}_{goal}$ from the current robot state $\boldsymbol{x}_{current}$.

In each time step k, hypotheses regarding user plans $\boldsymbol{i}_k$ are generated. In this paper, first all plausible goal state candidates $\boldsymbol{x}_{goal}$ are generated, and in a second phase all trajectories $\boldsymbol{i}_k$ to these goal states. A probability distribution is maintained over the generated user plan hypotheses $\boldsymbol{i}_k$. Typically, at start-up, the probability function over user plans is modelled to be a uniform distribution, since at that time little is known regarding the desired robot's goal state $\boldsymbol{x}_{goal}$.

We assume that information from at most m past time steps influences the user plan and user signal at time k. At each time instant k, the BCI user performs a mental task corresponding to a discrete robot command $\boldsymbol{u}_k$, and the algorithm that processes the EEG signals $\boldsymbol{g}_{k-m:k}$ resulting from the user's thoughts yields a likelihood function over the set of possible commands $\boldsymbol{u}_k$, i.e. $p\left(\boldsymbol{u}_{k-m:k}|\boldsymbol{g}_{k-m:k}\right)$. The probability distribution over user plan hypotheses is then determined as follows:

$$
\begin{aligned}
& p\left(\boldsymbol{i}_{k-m:k}|\boldsymbol{g}_{k-m:k}\right) \\
& \overset{\text{total probability}}{=} \sum_{\boldsymbol{u}_{k-m:k}} p\left(\boldsymbol{i}_{k-m:k}, \boldsymbol{u}_{k-m:k}|\boldsymbol{g}_{k-m:k}\right) \\
& \overset{\text{product rule}}{=} \sum_{\boldsymbol{u}_{k-m:k}} p\left(\boldsymbol{i}_{k-m:k}|\boldsymbol{u}_{k-m:k}, \boldsymbol{g}_{k-m:k}\right) \cdot p\left(\boldsymbol{u}_{k-m:k}|\boldsymbol{g}_{k-m:k}\right) \\
& \overset{\text{see text}}{=} \sum_{\boldsymbol{u}_{k-m:k}} p\left(\boldsymbol{i}_{k-m:k}|\boldsymbol{u}_{k-m:k}\right) p\left(\boldsymbol{u}_{k-m:k}|\boldsymbol{g}_{k-m:k}\right).
\end{aligned}
\tag{1}
$$

Simplification of $p\left(\boldsymbol{i}_{k-m:k}|\boldsymbol{u}_{k-m:k}, \boldsymbol{g}_{k-m:k}\right)$ to $p\left(\boldsymbol{i}_{k-m:k}|\boldsymbol{u}_{k-m:k}\right)$ in the last equation is possible because knowledge of ground-truth information regarding the actual user signal $\boldsymbol{u}_{k-m:k}$ makes knowledge of the noisy data $\boldsymbol{g}_{k-m:k}$ superfluous.

The factor $p\left(\boldsymbol{i}_{k-m:k}|\boldsymbol{u}_{k-m:k}\right)$ in this equation can be determined as follows. Based on the robot behaviour encoded as $\boldsymbol{x}_k$ and the actual interface signals $\boldsymbol{u}_k$ the user gives, the probability function over $\boldsymbol{i}_{k-m:k}$ becomes [2]:

$$
\begin{aligned}
p_k\left(\boldsymbol{i}_{k-m+1:k}|\boldsymbol{u}_{k-m+1:k}\right) = \ & \eta \cdot p_{user}\left(\boldsymbol{u}_k|\boldsymbol{i}_{k-m+1:k}, \boldsymbol{u}_{k-m+1:k-1}\right) \\
& \cdot \sum_{\boldsymbol{i}_{k-m}} \Big(p_{process}\left(\boldsymbol{i}_k|\boldsymbol{i}_{k-m:k-1}, \boldsymbol{u}_{k-m:k-1}\right) \\
& \cdot p_{k-1}\left(\boldsymbol{i}_{k-m:k-1}|\boldsymbol{u}_{k-m:k-1}\right) \Big)
\end{aligned}
\tag{2}
$$

ESANN'2009 proceedings, European Symposium on Artificial Neural Networks - Advances in Computational Intelligence and Learning. Bruges (Belgium), 22-24 April 2009, d-side publi., ISBN 2-930307-09-9.

where:

1. p_{k-1} is the a priori distribution over user plans, given previous user signals $\boldsymbol{u}_{k-m:k-1} = \{\boldsymbol{u}_{k-m}, \cdots, \boldsymbol{u}_{k-1}\}$. It reflects the belief in the different user plan hypotheses prior to the robot having moved and prior to having taken new user signals into account.

2. p_{user} is the *user model*, which expresses the likelihood that the user gives the observed interface signal $\boldsymbol{u}_k$, given that the user has had intent evolution $\boldsymbol{i}_{k-m+1:k}$, and given previous user signals $\boldsymbol{u}_{k-m+1:k-1}$.

3. $p_{process}$ is the *plan process model*, which determines both the shape and the probability of a user plan $\boldsymbol{i}_k$ at time k, given that the user has had intent evolution $\boldsymbol{i}_{k-m:k-1}$.

4. p_k is the a posteriori distribution over user intents, i.e. the probability of the different user plans after user signals and robot motion have been taken into account.

5. η is a scale factor to normalise the probability distribution.

This framework will be explained further in the next section.

3 Application to robotic wheelchairs

Various experiments have been performed with a BCI on our robotic wheelchair in the framework of the European project MAIA [4]. In this paper, we present an experiment where we first evaluate the performance of a BCI user who directly controls a simulated wheelchair. Secondly, it is verified whether the proposed Bayesian plan recognition algorithm can estimate a richer representation of the task the robot should execute.

To evaluate the control performance of the BCI user, the BCI user was asked to execute straight line paths to different goal locations with a simulated wheelchair (see Fig. 1), where the user is in *full control* of the wheelchair.

The simulator selects a goal position at random, and indicates this to the user via a line from the robot pose to the goal position. This way, ground truth regarding user plans is known. The user can give three commands with the BCI: *left* (turning 0.1 rad counterclockwise), *forward* (moving 0.2 m forward), and *right* (turning 0.1 rad clockwise). The simulator considers a goal position to be reached if the robot comes within a circle with radius $r_{goal} = 0.5$ m of the goal position, after which a new goal position is immediately selected.[1]

For this setup, a user plan can be represented as a straight-line path from the current robot location to a goal location. The set of goal positions is assumed

[1]This radius was chosen to ease navigation in simulation. For real navigation, this radius may have to be smaller, e.g. for docking at a table. Given the difficulty of achieving goal positions with the chosen radius already (cf. Section 4), this indicates how difficult real wheelchair navigation with a BCI can be.

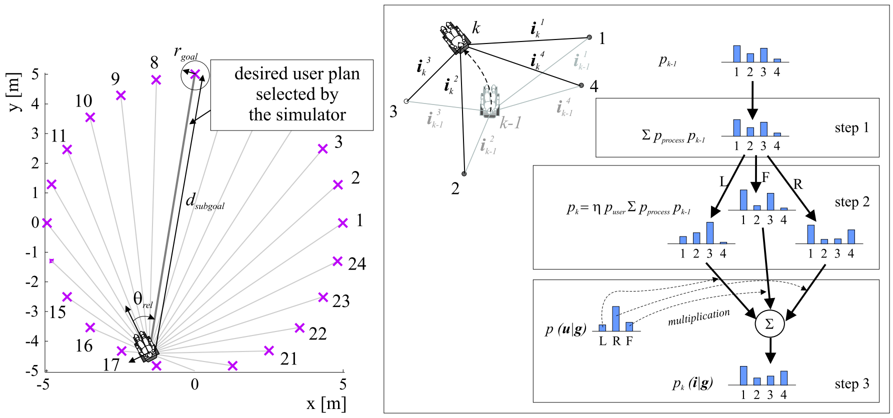

Fig. 1: The left figure shows the set of goal positions and the indication of the goal position that the user should pursue. The right figure schematically depicts the plan recognition process for a situation with 4 goal positions (explanation in the text).

to be known a priori. The 24 crosses in Fig. 1 correspond to these global goal positions.

Plan recognition then proceeds as depicted schematically in Fig. 1. At startup, the probability function over user plan hypotheses $\boldsymbol{i}_k$ is taken to be uniform. At time step k, the probability distribution $p_k\left(\boldsymbol{i}_{k-m+1:k}|\boldsymbol{u}_{k-m+1:k}\right)$ is first predicted based on the plan process function $p_{process}\left(\boldsymbol{i}_k|\boldsymbol{i}_{k-m:k-1}, \boldsymbol{u}_{k-m:k-1}\right)$ (this corresponds to the summation in Eq. 2 and to step 1 in Fig. 1). More specifically, when the robot moves from state $\boldsymbol{x}_{k-1}$ to state $\boldsymbol{x}_k$, the straight path between $\boldsymbol{x}_{k-1}$ and the j-th goal is transformed into the straight path between $\boldsymbol{x}_k$ and the j-th goal. The j-th probability is completely transferred to this new path.

Next, the user's signal is taken into account using the user model. The user model $p_{user}\left(\boldsymbol{u}_k|\boldsymbol{i}_{k-m+1:k}, \boldsymbol{u}_{k-m+1:k-1}\right)$ receives as input the plan to a goal position, i.e. a straight path from the current pose to a goal position. The user is assumed to first turn over the shortest angle in the direction of the goal location, and then to move forward. The chosen user models are shown in Fig. 2. The user model for commands *left* and *right* is a function of the previous angle tracking error $\theta_{rel,\,k-1}$ and of the current angle tracking error $\theta_{rel,\,k}$ (hence $m = 2$). The user model for command *forward* is made dependent on the distance to the subgoal $d_{subgoal}$ and the current angle tracking error $\theta_{rel,\,k}$, where its likelihood decreases as the distance to the goal decreases. The probability distribution $p_k\left(\boldsymbol{i}_{k-m+1:k}|\boldsymbol{u}_{k-m+1:k}\right)$ is calculated for each of the possible user signals $\boldsymbol{u}_k$ (this corresponds to the multiplication of user model and summation in Eq. 2 and to step 2 in Fig. 1).

In a final step, the actual BCI signals are taken into account (this corresponds to Eq. 1 and to step 3 in Fig. 1). The BCI adopted in this experiment is

ESANN'2009 proceedings, European Symposium on Artificial Neural Networks - Advances in Computational Intelligence and Learning. Bruges (Belgium), 22-24 April 2009, d-side publi., ISBN 2-930307-09-9.

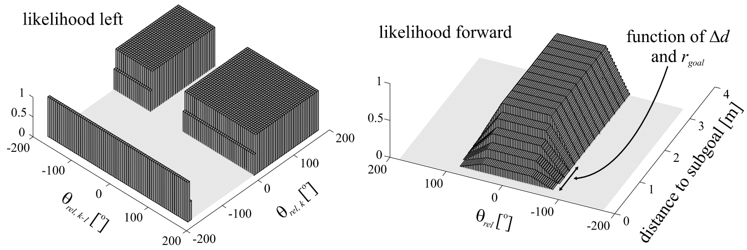

Fig. 2: This figure shows a user model for *left* and *forward* buttons. The likelihood function for *left* signals is modelled to be a function of the previous and current tracking error $\theta_{rel,\,k-1}$ and $\theta_{rel,\,k}$, and the *forward* likelihood function depends on the distance to the goal and on the current angle tracking error $\theta_{rel,\,k}$.

noninvasive, and consists of a cap with 64 electrodes measuring the EEG signals $\boldsymbol{g}$ of the user. The BCI user generates different EEG patterns by performing different mental tasks, such as the imaginary preparation to move the left hand or performing non-trivial arithmetics. At this moment, three different discrete commands can be discerned this way (corresponding to commands *left*, *right* and *forward*). This classification occurs at a constant rate of 2 Hz, and yields a probability distribution $p\left(\boldsymbol{u}_k|\boldsymbol{g}_{k-m:k}\right)$ over the discrete set of commands $\boldsymbol{u}$.

In order to navigate, the interface class that has the maximum probability is chosen to be sent as a control command to the robot. However, the plan recognition algorithm adopts the full distribution $p\left(\boldsymbol{u}_k|\boldsymbol{g}_{k-m:k}\right)$ to maintain a probability distribution over user plans.

4 Experimental results with BCI

Fig. 3 shows several snapshots of trajectories executed by the BCI user. The performance of the plan recognition algorithm is acceptable in that the actual user plan is estimated rather well when the robot is near the goal. Even if the robot is far from the goal, either the true goal position or some goal positions in its neighbourhood are probable, and therefore decisions based on these probabilities can be expected to yield correct assistive behaviour.

These results show that BCI users can manage to drive to any goal location themselves. However, their path is not smooth and hence navigation assistance can be beneficial. The plan recognition algorithm proposed in this paper seems a promising way to take assistive actions. As shown in [2] for button interfaces, the efficiency and accuracy of robot navigation can be increased this way.

5 Conclusions and future work

This paper presented a Bayesian plan recognition framework that can be adopted with BCIs to more accurately estimate the task that assistive devices should

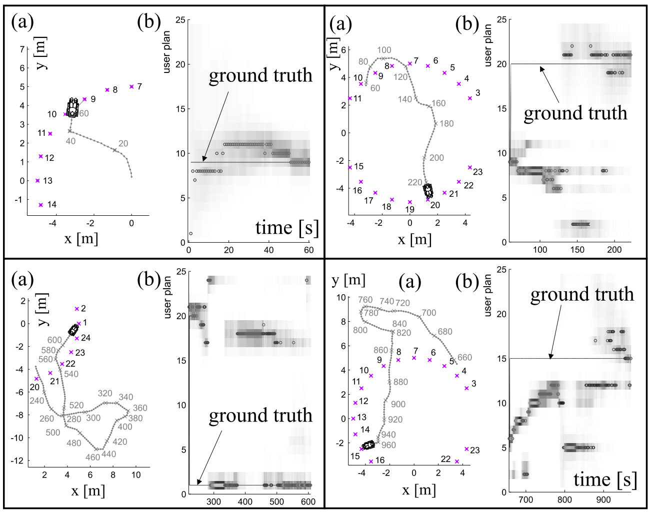

Fig. 3: These 4 figures show snapshots of experiments with the wheelchair simulator and a real BCI. The left figures (a) show the trajectory followed by the user. Time steps are plotted along the followed trajectory at regular intervals. The right figures (b) show the evolution of the probability function over time. The horizontal line in the right figures represents the ground truth user plan. The grey values in the right figures correspond to the probability of a user plan: the darker a cell, the more probable the corresponding user plan is at that moment.

execute. When evaluating the performance of BCI users who directly control a simulated wheelchair, it was found that BCI users can manage to reach any position desired, but robotic assistance based on Bayesian plan recognition may make navigation more accurate and efficient.

References

[1] Sandra Carberry, Techniques for Plan Recognition, *User Modeling and User-Adapted Interaction*, vol. 11, no. 1–2, 2001, pp. 31–48.

[2] E. Demeester, A. Hüntemann, D. Vanhooydonck, G. Vanacker, H. Van Brussel, M. Nuttin, User-adapted plan recognition and user-adapted shared control: A Bayesian approach to semi-autonomous wheelchair driving, *Autonomous Robots*, vol. 24, no. 2, February 2008, pp. 193 - 211.

[3] J. Del R. Millán, P. W. Ferrez, F. Gallán, E. Lew and R. Chavarriaga, Non-invasive brain-machine interaction, *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 22, no. 5, 2008, pp. 959-972.

[4] Home page of the European project MAIA, http://www.maia-project.org/.