Proceedings of the 5th International
IEEE EMBS Conference on Neural Engineering
Cancun, Mexico, April 27 - May 1, 2011

Towards Hierarchical BCIs for Robotic Control

Mike Chung, Willy Cheung, Reinhold Scherer, Member, IEEE, and Rajesh P. N. Rao, Member, IEEE

***Abstract*— There has been growing interest in brain-computer interfaces (BCIs) for controlling robotic devices and prosthetics directly using brain signals. Non-invasive BCIs, such as those based on electroencephalographic (EEG) signals, suffer from low signal-to-noise ratio, limiting the bandwidth of control. Invasive BCIs, on the other hand, allow fine-grained control but can leave users exhausted over long periods of time because of the amount of attention required for control on a moment-by-moment basis. In this paper, we address these problems using a new adaptive and hierarchical approach to brain-computer interfacing. The approach allows a user to teach the BCI system new skills on-the-fly; these learned skills are later invoked directly as high-level commands, relieving the user of tedious lower-level control. We demonstrate the approach using a hierarchical EEG-based BCI for controlling a humanoid robot. In a study involving four human subjects controlling the robot in a simulated home environment, each subject successfully used the BCI to teach the robot a new navigational task. They later were able to execute the same task by selecting the newly learned command from the BCI's adaptive menu, avoiding the need for low-level control. A comparison of the performance of the system under low-level and hierarchical control revealed that hierarchical control is both faster and more accurate. Our results suggest that hierarchical BCIs can provide a flexible and robust way of controlling complex robotic devices, satisfying the dual goals of decreasing the cognitive load on the user while maintaining the ability to adapt to the user's needs.**

I. INTRODUCTION

Brain-computer interfaces (BCIs) have received considerable attention in recent years due to their novel hands-free mode of interaction with the environment [1], [2], [3]. In particular, the field has seen rapid growth due to its potential for offering a new means of control for devices tailored to severely disabled and paralyzed people: examples include directing the motion of a motorized wheelchair, controlling a semiautonomous helper robot, or using a neuroprosthesis [4], [5], [6].

The most commonly used brain signal source for non-invasive BCIs in humans is the electroencephalogram (EEG). Due to its non-stationarity, inherent variability, and low signal-to-noise ratio, a reliable translation of EEG into appropriate control messages for devices can be difficult and slow.

This work was supported by the National Science Foundation (0622252 & 0930908), the Packard Foundation, and the Office of Naval Research (ONR), and the ICT Collaborative Project BrainAble (247447). We thank Rawichote Chalodhorn for helping with the HOAP-2 robot and Webots programming aspects of the project and Josef Faller for helping with the implementation of the software SSVEP stimuli.

M. Chung, W. Cheung, R. P. N. Rao are with the Neural Systems Laboratory, Computer Science & Engineering, University of Washington, Seattle, USA, {mjyc,wllychng,rao}@cs.washington.edu

R. Scherer is with the Institute for Knowledge Discovery, BCI-Lab, Graz University of Technology, Krenngasse 37, 8010 Graz, Austria reinhold.scherer@tugraz.at

Therefore, EEG signals have often been used to select a task that can be semiautonomously performed by an application (e.g., control of a humanoid robot in [5]). Invasive BCIs offer higher bandwidth and allow fine-grained control of robotic devices (e.g., [7]) but such moment-by-moment control over long periods of time can be a significant cognitive load and can exhaust the user.

In this paper, we introduce a hierarchical and adaptive approach to BCIs that combines the flexibility of fine-grained control with the lower cognitive load of coarse-grained menu-driven systems. The proposed approach allows the user to teach the system new and useful tasks on an ongoing basis. This leads to a scalable hierarchical BCI system wherein lower-level actions are first learned and later semi-autonomously executed using a higher-level command, freeing the user from having to engage in tedious moment-by-moment control. We explore the efficacy of such a BCI system in the context of controlling a humanoid helper robot.

We report results from a study in which the user's task is to navigate the robot to a desired room in a simulated home environment. In the training phase, the user guides the walking robot using three fine-grained control commands: turn left, turn right and stop. After repeated trials, the robot learns the navigational task and is consequently able to autonomously navigate to the desired room. In subsequent trials, the user has only to select the newly learned skill ("Go to room X") as a higher-level command in an adaptive menu to get the robot to navigate to the desired location. Four human subjects were able to successfully teach the simulated robot a new navigational task and later repeat the assigned task by selecting the newly-learned command. Our results provide a proof-of-concept demonstration that hierarchical BCIs may offer a flexible and robust approach to controlling complex robotic devices while minimizing the cognitive load on the human user.

II. METHODS

The hierarchical BCI proposed in this paper is composed of three main components: (A) a steady state visual evoked potential (SSVEP) based BCI: although other EEG responses such as P300 or mental imagery could also be used, we used SSVEPs in the present study because they offer relatively high information transfer rates (ITR) with minimal user training; (B) a hierarchical menu and learning system that allows the user to teach the system new skills, and (C) the application, which, in the present case, is a simulation of a humanoid robot in a home environment that mimics the physics of the real world (Figs. 1A and 1B). The three components interact closely to make the system work. In

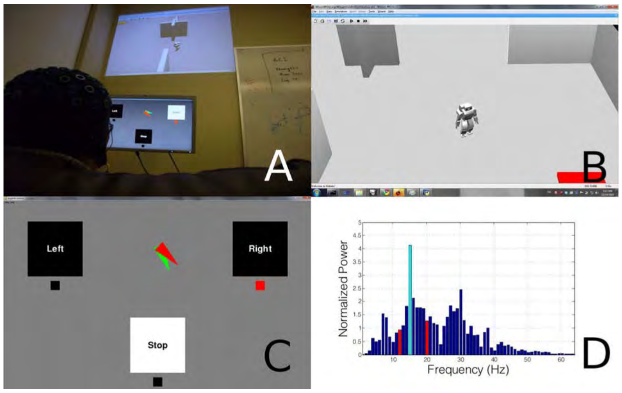

Fig. 1. **A hierarchical BCI system.** A. Experimental setup: User selects from a menu shown on a monitor while a view of the robot in its environment is shown in a larger immersive setting above, B. Simulated robot in its environment: The robot is a Fujitsu HOAP-2 humanoid simulated using the Webots software, C. A screen shot of the menu and SSVEP stimulation, D. Frequency domain representation of a subject's EEG signal illustrating a high SSVEP response to 15Hz stimulation.

particular, the hierarchical adaptive menu system displays available commands as flashing stimuli for the user to choose using the SSVEP-based BCI. The user makes the desired selection by focusing on the desired command in the menu (Fig. 1C). The BCI detects the option the user is focusing on and sends its classification output to the hierarchical menu system which sends a command to the robot and switches to the next appropriate menu. The robot executes the command it receives, which can be either a lower-level command such as turn left/right or a higher-level learned command. Finally, the user closes the control loop by observing the simulated robot's action and makes the next desired selection based on the updated menu.

We describe each of the components of the hierarchical BCI system in more detail below.

A. SSVEP-based BCI

Flickering stimuli used to elicit SSVEPs were presented on a TFT computer screen with a refresh rate of 60 Hz. Up to three different options (12 Hz, 15 Hz, and 20 Hz) could be presented to the user in any given menu.

Continuous EEG was recorded bipolarly from gold electrodes placed at electrode positions Oz and Cz (ground was linked to Cz), notch filtered at 60 Hz and digitized at 256 Hz (gUSBamp, Guger Technologies, Graz, Austria).

To detect the flashing stimulus the user was focusing on, the power spectrum was estimated using the Fast Fourier Transform (FFT). FFT was applied to 1s segments of EEG data (Hamming window) every 0.5s and the power for each frequency was then calculated using squared values. The data used for final classification was a 4-second average of these power values (calculated from 8 FFT values). The frequency with the highest power among the three target frequencies of 12, 15, and 20Hz was classified as the user's choice for that decision period (see Fig. 1D for an example).

The BCI menu on the computer monitor and a video projection of the robot simulator were placed one above the other (Fig. 1A). When the user desired BCI control, they focused on the monitor, while at other times, they watched the robot move in its environment. When the user was not focusing on the BCI menu, the power in the recorded EEG channel was markedly different, allowing a simple threshold-based detector to self-initiate the SSVEP-BCI whenever the user required control.

B. Hierarchical Adaptive Menu

The hierarchical menu (Fig. 2) is the interface the subject uses to interact with the hierarchical learning system. It displays the available commands for the hierarchical learning system, which are selected using SSVEP. The top-level menu presents two options: 'Train' and 'Test'.

Selecting 'Train' allows the user to either teach the system a new task ('new' option) or update an existing one ('existing' option). If 'new' is selected, the next menu presented is the robot navigation menu. If 'existing' is selected, the user must choose a task to update before the navigation menu is displayed (see Fig. 2). In navigating the robot, the user has three choices: left, right, and a stop option indicating the user is done with the task. To continue moving forward in the current direction, the user need not make a choice. When 'stop' is selected, a menu offers the user the option of saving the task for inclusion in the training dataset for training the robot. In order to mitigate the effects of erroneous classifications, the system includes various confirmation menus, giving users the ability to verify or correct their last choice.

Selecting 'Test' allows the user to select a task that was previously learned by the system. After the user has demonstrated and saved examples of a task, the robot learns the task (see next section) and the system incorporates this task into the hierarchical menu as a new option in the 'Test' menu. The user can now simply select the task as a high-level command, and be at ease while the robot autonomously performs the task.

C. Robot Application

Our previous work demonstrated a BCI for high-level control of a Fujitsu HOAP-2 humanoid robot [5]. For the present study, we used the Webots simulator [8] to simulate the HOAP-2 robot rather than use the actual robot since we wanted to focus on the hierarchical learning system. Note that rather than representing an animation of the robot, the Webots software simulates the physical dynamics of the Fujitsu HOAP-2 robot and its environment; this facilitates the transition of the results to real-world scenarios. In the experiments, the robot was initially located in the lower left corner of a simulated home environment. The environment was divided into four rooms by four walls, each with a door allowing entry into a room from an adjacent room (see Fig. 3 for overhead view). Each room had a distinctively colored box for identification purposes.

The simulated robot was preprogrammed with routines to walk forward, turn right, turn left, and make smooth

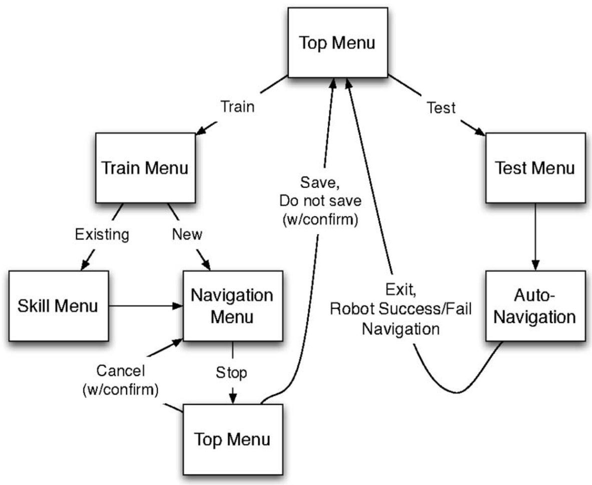

Fig. 2. **Overview of control flow in the hierarchical menu system.**

transitions from one motion to another motion. The robot was also programmed with a simple collision avoidance behavior to keep the robot from walking into a wall or other obstacles during navigation. Given these basic navigational routines, we developed a controller for robot navigation, with the user having a birds-eye view of the robot. The robot was always in motion unless stopped by the user or the collision avoidance behavior.

The hierarchical BCI differs from traditional BCI systems in its ability to learn new behaviors from user demonstration. Learning occurs in the robotic component of the BCI system, and is then abstracted into the hierarchical menu system. In the current implementation, we utilized a simple position-based learning method for navigation based on a radial basis function (RBF) neural network [9]. Using a simulated on-board GPS sensor, the robot's position data was logged at a sampling rate of 0.5hz as the user guided the robot to a desired location. When the user subsequently commands the robot to learn the demonstrated navigation skill, the robot uses the logged position data to calculate trajectories for training the RBF network. The logged data and neural network are stored locally, and the user can update a selected skill with more demonstrations as needed, improving performance over time. This arrangement also allows training over multiple days.

D. Experimental Procedure

Four able-bodied subjects participated in the experiments (all male, 25 $\pm$3.2 yr). To calibrate SSVEP response, subjects were given instructions to attend one out of three flashing stimuli (cue-guided 4-s randomized trials, inter-trial period 3$\pm$0.5 s, 5 trials per stimulation frequency) presented on the screen (see Fig.1.C). Each subject ran two or three of these SSVEP-only sessions (about 10 minutes). After the calibration, subjects had 20 minutes of free training to get familiar with the hierarchical BCI before the experiment started. For the experiment, subjects were first given the task

TABLE I

Performance Comparison

	Low-level BCI	Hierarchical BCI
Mean among four subjects (std)		
Number of selections made	20 (7)	5 (2)
Task completion time (s)	220 (67)	112 (25)
Navigation only time (s)	124 (37)	73 (19)
Mean of three trials from best subject (std)		
Number of selections made	15 (5)	4 (1)
Task completion time (s)	141 (42)	85 (4)
Navigation only time (s)	99 (30)	74 (9)
Minimum		
Number of selections made	8	4
Task completion time (s)	91	75
Navigation only time (s)	59	59

of manually navigating the robot from the initial position (lower-left corner) to an assigned goal position (lower-right corner) using low-level commands (left/right/stop). From these user-guided trajectories, the hierarchical system learned the task, and subjects were subsequently asked to reproduce the same task using the high-level command learned by the hierarchical system. This procedure was done once, and the duration was 10 to 15 minutes for each subject.

We additionally conducted a more extensive experimental session with the best subject from our first set of experiments, where he was asked to perform the navigation task three times using low-level controls and three times using the high-level command.

To compare the performance of the hierarchical to the low-level BCI, we employed three metrics (Table I): cognitive load, measured by the number of commands the user had to issue to achieve a given task ('Number of selections made'); the time taken to complete the task ('Task completion time'); and the time spent only on controlling the robot ('Navigation only time').

III. RESULTS

All four subjects were able to use the hierarchical BCI to complete the assigned tasks. The average($\pm$std) SSVEP-based 3-class accuracy for the four subjects from the calibration was 77.5%$\pm$13.8. Although somewhat lower than other SSVEP rates reported in the literature, all four subjects were able to successfully complete the requested tasks using the entire system with closed-loop feedback.

Results obtained for the three different performance metrics are shown in Table I. In the table, we also include for comparison purposes the minimum values for these metrics, assuming 100% SSVEP accuracy.

The results indicate that for all three metrics, subjects demonstrate improved performance using the hierarchical BCI: both the mean and variance for all three performance metrics are lower when using the hierarchical BCI compared to the low-level BCI.

Results from the best performing subject provide interesting insights regarding the use of high-level commands in a hierarchical BCI. Due to the high SSVEP accuracy of this subject (90%), the difference in the mean values between

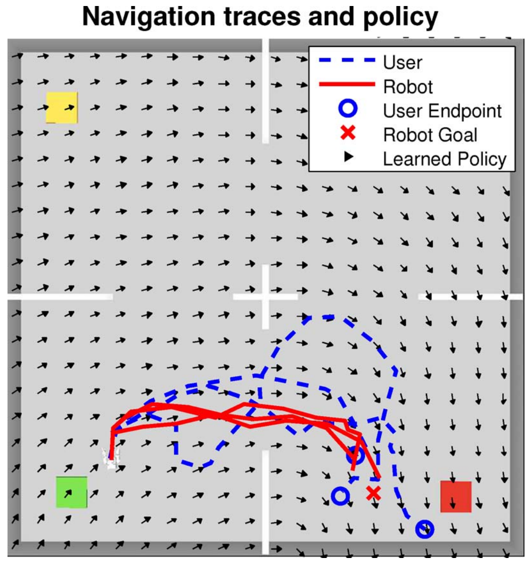

Fig. 3. **Example robot trajectories from user-demonstrated low-level control and hierarchical control**. The blue trajectories represent low-level navigational control by the user. These trajectories were used to train the RBF neural network. The red trajectories represent autonomous navigation by the robot using the RBF network after selection of the corresponding high-level command by the user. The small arrows indicate the vector field learned by the RBF network ('Learned Policy') based on the user's demonstrated trajectories.

low-level and hierarchical modes of control was smaller, but the variance for low-level control was significantly greater than for higher-level control (Table I). This is corroborated by the navigational traces in Figure 3, where we see that trajectories from the hierarchical BCI tend to follow the minimal path to the goal location based on the learned representation in the neural network. This result confirms the expectation that the network learns an interpolated trajectory that minimizes the variances inherent in the training trajectories, with more training data leading to better performance.

IV. Discussion

1) Combination of Scalability and Efficiency: As noted in the introduction, current BCIs for robotic control have a trade-off between cognitive load and scalability. More robotic autonomy [5] implies coarse-grained control and less flexibility, while fine-grained control provides greater flexibility but higher cognitive load. Our approach attempts to combine the advantages of these two approaches using a hierarchical learning system. We provide preliminary results in this paper but future work could incorporate more powerful machine learning techniques such as Gaussian processes for supervised learning from user demonstrations and probabilistic reasoning for making decisions during control and menu-based selection.

2) Multi-tasking for Increasing Bandwidth: During execution of high-level commands, users were not required to be engaged with the BCI, and typically waited while the robot finished the assigned task. During this waiting period, the user could potentially control a *second* device using another BCI menu, allowing multi-tasking. We have tested this idea in preliminary experiments where the user has the option of controlling the brightness of lights in the rooms while the robot is executing the high-level command. Our best subject demonstrated the ability to control brightness as instructed. These results suggest a new way of increasing the BCI bandwidth through multi-tasking.

3) Longterm Usability: Current EEG BCIs suffer from the problem of inconsistent performance over multiple days. Our approach provides a way of mitigating this problem by storing user-taught skills for long-term use, allowing a learned skill to be selected as a high-level command and executed consistently from day to day. The user can teach the system new skills on "high-performance" days, and execute these skills reliably with much less effort on subsequent days.

V. Conclusion

This paper proposes a new approach to scalable and adaptive control of robotic devices using hierarchical BCIs. To our knowledge, the results presented here are the first demonstrations of hierarchical and adaptive robotic control using non-invasive brain signals. The main aim of this study was to provide a proof of concept of the hierarchical BCI system. Therefore, we used a straightforward SSVEP-based approach and a simple GUI. Our next set of studies will target actual robotic devices, including a mobile robot and a robotic arm-hand system, as well as explore the use of other types of brain responses (P300 and imagery). In the long-term, we believe that a hierarchical adaptive BCI system that combines low-level control with high-level commands as suggested here could significantly enhance human-machine interaction.

References

[1] R. Rao and R. Scherer, "Brain-computer interfacing [in the spotlight]," *Signal Processing Magazine, IEEE*, vol. 27, no. 4, pp. 152 –150, 2010.

[2] R. Scherer, F. Lee, A. Schlogl, R. Leeb, H. Bischof, and G. Pfurtscheller, "Toward self-paced brain–computer communication: Navigation through virtual worlds," *Biomedical Engineering, IEEE Transactions on*, vol. 55, no. 2, pp. 675–682, 2008.

[3] J. Faller, G. Müller-Putz, D. Schmalstieg, and G. Pfurtscheller, "An Application Framework for Controlling an Avatar in a Desktop-Based Virtual Environment via a Software SSVEP Brain-Computer Interface," *Presence: Teleoperators and Virtual Environments*, vol. 19, no. 1, pp. 25–34, 2010.

[4] F. Galán, M. Nuttin, E. Lew, P. Ferrez, G. Vanacker, J. Philips, and J. Millán, "A Brain-Actuated Wheelchair: Asynchronous and Non-Invasive Brain-Computer Interfaces for Continuous Control of Robots," *Clinical Neurophysiology*, vol. 119, no. 9, pp. 2159–2169, 2008.

[5] C. Bell, P. Shenoy, R. Chalodhorn, and R. Rao, "Control of a humanoid robot by a noninvasive brain–computer interface in humans," *of Neural Engineering*, vol. 5, p. 214, 2008.

[6] G. R. Müller-Putz, R. Scherer, G. Pfurtscheller, and R. Rupp, "EEG-based neuroprosthesis control: a step towards clinical practice," *science Letters*, vol. 382, pp. 169–174, 2005.

[7] M. Velliste, S. Perel, M. Spalding, A. Whitford, and A. Schwartz, "Cortical control of a prosthetic arm for self-feeding," no. 7198, pp. 1098–1101, 2008.

[8] "Webots," http://www.cyberbotics.com/, 2010, [Online; accessed 12-13-2010].

[9] "MATLAB newgrnn," newgrnn.html, 2010, [Online; accessed 12-13-2010].