2011 11th IEEE-RAS International Conference on Humanoid Robots
Bled, Slovenia, October 26-28, 2011

An Adaptive Brain-Computer Interface for Humanoid Robot Control

Matthew Bryan*, Joshua Green*, Mike Chung*, Lillian Chang*,‡,
Reinhold Scherer†, Joshua Smith*, Rajesh P. N. Rao*
*Computer Science & Engineering, University of Washington, Seattle, USA
†Institute for Knowledge Discovery, TU Graz, Graz, Austria
‡Intel Corporation, UW Intel Science and Technology Center, Seattle, USA
Email: {mmattb,greenj5,mjyc,lchang,jrs,rao}@cs.washington.edu, Reinhold.Scherer@tugraz.at

Abstract—**Recent advances in neuroscience and humanoid robotics have allowed initial demonstrations of brain-computer interfaces (BCIs) for controlling humanoid robots. However, previous BCIs have relied on higher-level control based on fixed pre-wired behaviors. On the other hand, low-level control can be tedious, imposing a high cognitive load on the BCI user. To address these problems, we previously proposed an adaptive hierarchical approach to brain-computer interfacing: users teach the BCI system new skills on-the-fly; these skills can later be invoked directly as high-level commands, relieving the user of tedious control. In this paper, we explore the application of hierarchical BCIs to the task of controlling a PR2 humanoid robot and teaching it new skills. We further explore the use of explicitly-defined sequences of commands as a way for the user to define a more complex task involving multiple state spaces. We report results from three subjects who used a hierarchical electroencephalogram (EEG)-based BCI to successfully train and control the PR2 humanoid robot in a simulated household task maneuvering the robot's arm to pour milk over a bowl of cereal. We present the first demonstration of training a hierarchical BCI for a non-navigational task. This is also the first demonstration of using one to train a more complex task involving multiple state spaces.**

I. Introduction

Using humanoid robots to perform remote tasks with human supervision has been a subject of considerable interest in the robotics community [1], [2], [3], [4]. Humanoid robots are often considered proxies or assistants to humans, performing tasks in both real-world environments designed for humans and in environments considered too dangerous for humans [1], [2]. Additionally, humanoid robots have also been proposed as a telecommunications medium [4].

One interesting approach to controlling a humanoid robot is direct brain-based control using a brain-computer interface (BCI) [5], [6], [7]. To achieve such control, one must confront the high degrees-of-freedom inherent in humanoid robots. This problem is common to other, more standard ways of controlling a humanoid, such as using a joystick, speech recognition, and visual feedback systems [8], [9], [10]. This problem is made worse when we couple a humanoid with BCIs, which tend to have low throughput. Low throughput implies the BCI system contrains the number and resolution of commands the user can discriminate between in any given period of time.

Besides low throughput, there are problems inherent in using BCIs as robotic controllers. Non-invasive BCIs, e.g., those based on EEG signals recorded from the scalp, suffer from low signal-to-noise ratio, which limits the bandwidth of control. Invasive BCIs that tap directly into neurons in the brain allow fine-grained control, but such an approach can leave users exhausted since control is typically exerted on a moment-by-moment basis [11].

To overcome these problems, we previously proposed an adaptive hierarchical BCI (HBCI) architecture which allows the user to teach the system new and useful tasks on an ongoing basis [12]. First, the user demonstrates a higher-level skill to the robot using only lower-level commands (e.g. turn left). Later, the user executes the high-level command (e.g., the command "go to point A on the map"), which is then carried out semi-autonomously by the robot. Such higher-level control frees the the user from having to engage in tedious moment-by-moment control once a command has been learned.

These previous HBCIs applied only to navigational tasks, limiting their domain of use. In this paper, we investigate whether an HBCI can be used in a close-range space, such as the space of all points where a humanoid robot manipulator can reach. We demonstrate the learning of arm trajectories using the HBCI. We give the user the ability to train the robot to maneuver its grippers in a close-quarters environment. Second, we propose a simple sequence memorization scheme where a user is able to define a new higher-level skill by concatenating several mid-level skills and lower-level motor primitives such as opening/closing and rotating the gripper. The user can subsequently use this sequence command to solve the more complex task. Since sequencing in this case involves individual commands at any level of the control hierarchy, it is not tied to any particular robot state space, allowing for the

This work was supported by the National Science Foundation (0622252 & 0930908), the Office of Naval Research (ONR) Cognitive Science Program, and the ICT Collaborative Project BrainAble (247447). This material is based upon work supported by the National Science Foundation under Grant # 1019343 to the Computing Research Association for the CIFellows Project. We thank UW Sensor System Lab members Vibinash Thomas and Harnoor Singh for assisting with the PR2 programming aspects of the project. We also thank Joseph Godlewski for his considerable assistance in setting up and conducting experiments.

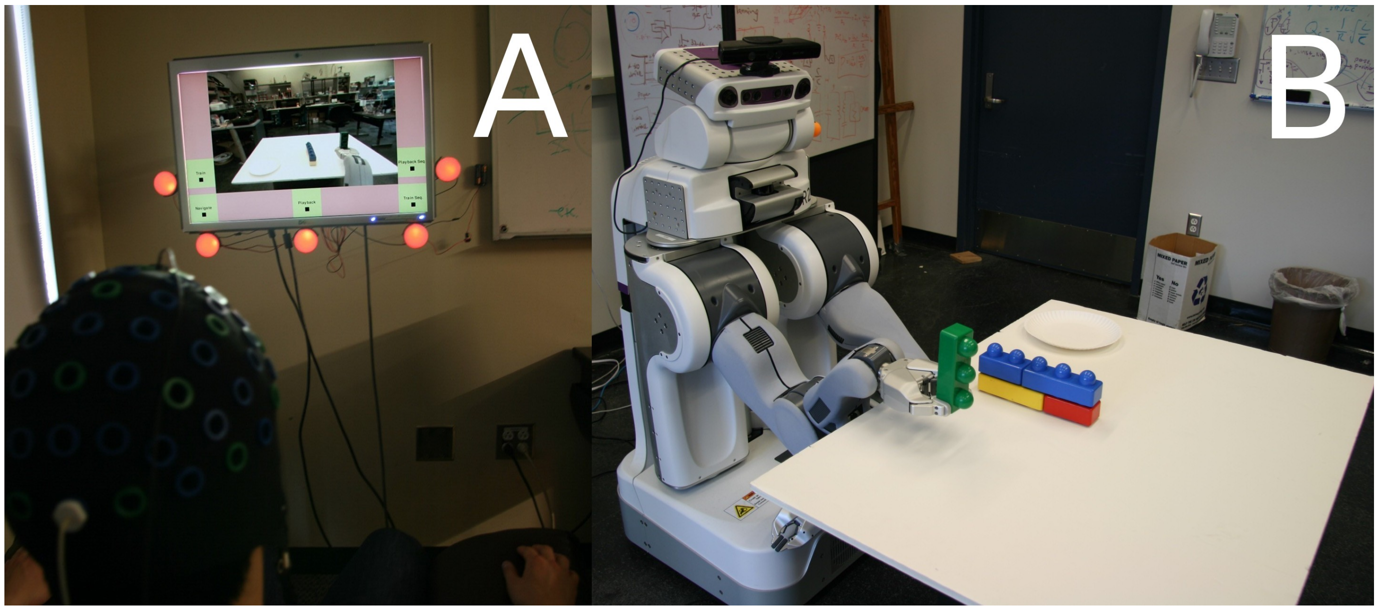

Fig. 1. **Experimental set-up**. A. The user selects from a menu shown on a monitor by focusing on one of five LED lights while visual feedback from the PR2 robot's head camera is provided on the main portion of the screen. B. Semi-humanoid PR2 robot in remote location. The robot was used in this paper for performing in-range manipulation tasks while remaining in a fixed position.

training of more interesting tasks.

While we only allowed the user to sequence lower- and mid-level skills for the sake of this demonstration, the technique given here could easily scale to arbitrary levels of hierarchy. A sequence could hold commands from any level of the command hierarchy, opening the door to sequences of sequences representing whole lists of tasks. For instance, in this case we demonstrate the pouring of milk over cereal, but the system could just as easily extend to making breakfast, which is a whole series of complex tasks.

We report results from three subjects who used a hierarchical EEG-based BCI to successfully train and control a PR2 humanoid robot. The task involved teaching the PR2 to perform actions that simulate pouring milk over a bowl of cereal. Our results suggest that HBCIs could offer an efficient and adaptive approach to using brain signals to teach and control humanoid robots the solutions to complex tasks.

II. System Architecture

The hierarchical architecture is a modular system with three elements:

1) A control interface which receives noisy input from the brain, in this case a BCI based on Steady State Visually-Evoked Potentials (SSVEPs). The BCI operates by exposing the user to oscillating visual stimuli. Electrical activity corresponding to the frequency of this oscillation (and its multiples) can be measured from the occiptial lobe of the brain located at the back of the skull. The user issues a command by choosing a stimulus (and therefore a frequency) to pay attention to. The BCI measures the corresponding EEG activity and attempts to infer from it the command the user has chosen for execution [13] (Fig. 1.A). Note, though, that other BCI paradigms allowing discrete classification could be used as well;
2) A hierarchical and adaptive menu;
3) A humanoid robot; in this case we used the Willow Garage PR2 semi-humanoid robot which has a head with 2 dofs and two arms with 7-8 dofs each (Fig. 1.B), but other humanoid robots could also be used.

We included several enhancements relative to the system proposed in [12] to scale up the interface to control a high degrees-of-freedom semi-humanoid robot. We used ROS [14] to handle the communication between components. We describe each component in detail below.

A. *SSVEP-based BCI*

LEDs were used as stimuli for generating SSVEPs (rather than an LCD monitor as in [12]). We have found LEDs to offer a greater degree of freedom in choice of frequencies for SSVEP-based BCIs. We used five red LEDs in circular light boxes. These oscillated at frequencies of 12 Hz, 15 Hz, 17 Hz, 20 Hz, and 22 Hz. The light boxes were mounted to the screen next to their corresponding menu options as shown in Fig. 2 and laced behind a diffusive material to enhance the stimulus effect.

Continuous EEG signals were recorded bipolarly from gold electrodes placed at two standard locations on the skull - Fpz which is located on the forehead, and Oz, which is located at the center of the back of the skull. We connect ground at the Fpz position. The signal was notch filtered at 60Hz and digitized at 256Hz (gUSBamp, Guger Technologies, Graz, Austria).

We estimated the signal's power spectrum using the Fast

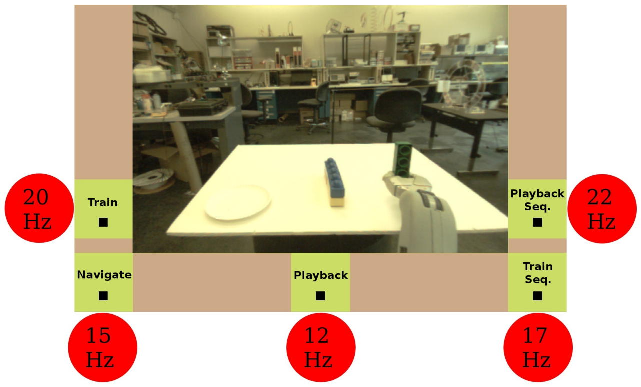

Fig. 2. **SSVEP-based Control Interface**. The interface is comprised of LED stimuli and the main screen. The main screen is composed of a video feed from the robot and current menu options. In the screenshot above, the top menu options are being displayed.

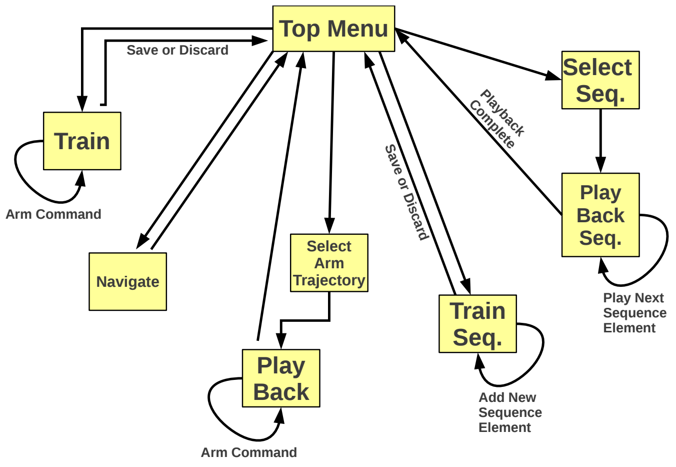

Fig. 3. **Overview of control flow in the hierarchical menu system.**

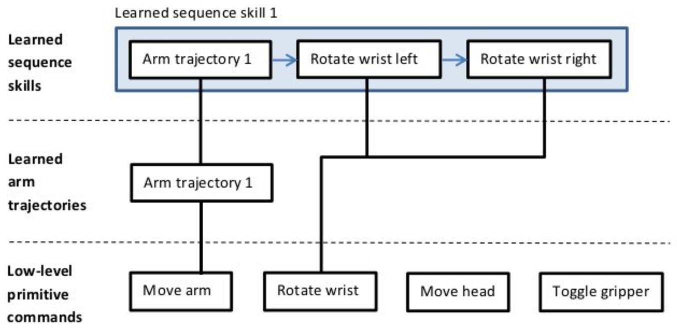

Fig. 4. **Hierarchical BCI Architecture**. The user can train arm trajectories and define higher-level command sequences consisting of lower-level primitives and learned trajectories.

Fourier Transform (FFT), which was used to classify the user's input. The FFT was applied to 1.0s intervals of the EEG signal every 0.5s and the power for each frequency was then computed using squared values. The data used for the classification was a 4s average of the power values. The frequency with the highest average was classified as the user's choice for that time period and the appropriate commands were sent to the robot and to the menu system.

Note that this hierarchical system allows for the use of many types of BCI paradigms other than SSVEP. Any BCI allowing discrete classification could make use of such a hierarchical command structure. As such, the use of a hierarchical command structure is not limited to SSVEP alone.

B. Hierarchical Adaptive Menu

The hierarchical menu is the user interface to the entire system. The menu allows the user to control the robot directly as well as interact with the learning system to teach the robot new skills. The menu display consists of five menu choices around the border of the screen. The middle of the screen displays a live video feed from a camera on the robot's head. When the experiment first begins, the menu presents 5 options: Train, Train Sequence, Navigate, Playback, and Playback Sequence (Fig. 2).

Selecting the command "Train" allows the user to teach the system a new arm trajectory skill. This menu gives the user the option of moving the arm forward, backwards, left, or right, or finish training the new skill. When the user selects "Stop", the user is presented with a save menu which gives the user the option of saving the skill or discarding the collected data. To enhance the robustness of selection, we present the user with several confirmation menus to safeguard against misclassification. After the user presents an example trajectory of a skill to learn, the learning system generates a policy mapping robot states to actions based on the recorded data.

We create the policy using Gaussian Processes as in [12]. This policy interpolates for areas away from the original training path, enabling it to be somewhat flexible with respect to the starting position of playback. The user can then test the learned skill using the Playback menu or use it as part of a a higher-level command sequence.

Note that while our screen to playback sequences in this case allows for only four choices of sequences, this is easily scalable to more. We could allow for an arbitrary number of menu screens, enough to hold the skills needed. Also, to prevent the user from needing to search through a large number of screens, the HBCI system could filter the current sequences based on relevance to the robot's present state (e.g. to not show sequences involving dropping an object if the gripper currently holds nothing and the sequence does not involve grasping).

Selecting the "Train Sequence" command gives the user the ability to create a higher-level command sequence, which can consist of previously trained arm trajectory commands as well as lower-level primitives such as rotating the wrist left or right, and opening or closing the gripper. When a user selects a skill to add to the new command sequence, a confirmation screen is displayed to verify the user's intentions. Any number of skills can be added to the sequence. Once the user selects "Exit" and saves the new sequence, it can then be selected from the Sequence Playback menu. When a user executes a high-level command sequence, the list of commands that make up this sequence will execute in order, one after the other. While the robot executes these commands, the user has the ability to move the head of the robot up, down, left, and right in order to monitor the robot's motion (see Fig. 4).

C. Robot and Robot Software

The experiments in this paper were carried out on a Willow Garage PR2 semi-humanoid robot. We presented the user with basic lower-level motor primitives using pre-programmed library functions provided by ROS [14]. These include basic object manipulation primitives such as toggling the right gripper open or closed and rotating the right wrist link. The library also includes head joint manipulation, allowing the user to swivel the head-mounted camera. Finally, it includes moving the gripper forward, backward, left, and right relative to the torso.

The current system focuses on learning arm movements using the position-based trajectory learning framework proposed in [12]. However, this approach could be extended to more complex learning schemes including control of multiple joints simultaneously or separately. We intend to explore such schemes in the near future.

Video feed from a camera on the robot's head was continuously fed to the user for visual feedback on the robot's actions (Fig. 2).

III. Experimental procedure

Three able-bodied male subjects (ages: 21, 27, 20) participated in the study, which was approved by the University of Washington Institutional Review Board. All subjects gave written informed consent.

We instructed the subjects to teach the humanoid a command sequence representing a milk-pouring task. The experiment assumes that an open container containing milk (represented by a lego block as shown in Fig. 1.B) is in the robot's right gripper. The user must maneuver the container over a receiving receptacle on the other side of the table and rotate the wrist to pour the milk. To make the task more challenging, we placed an obstacle (represented by the blue block) between the right gripper's starting position and the goal position (see Fig. 1).

Over the course of a single session, the subject was instructed to

1) Train an arm trajectory which avoids the obstacle and arrives at the goal position;
2) Create a command sequence representing the entire complex skill;
3) Play back the learned sequence from three different starting positions.

To demonstrate the robustness of the position-based method for learning arm trajectories, we reset the robot three times and instructed the user to execute the learned skill. With each reset, the robot's starting state was set to a different initial position. The initial positions were approximately the same for each user. To measure success, we observed whether the gripper ended over the bowl of cereal given the starting position.

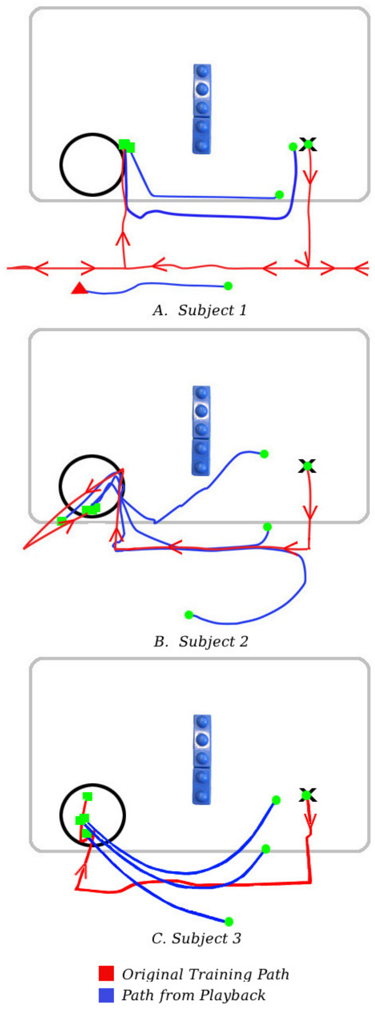

Fig. 5. **Subject Arm Trajectories** Red indicates an original training trajectory. Blue indicates trajectories that result from executing the learned policy. Circles indicate starting states. Squares indicate successful ending states. Triangles indicate unsuccessful ending states. Black X indicates starting state for original training.

IV. Results

All subjects were able to use the HBCI to teach the robot an arm trajectory and subsequently define a higher-level command sequence. In all cases the command sequence enabled the user to perform the more complex task with a minimal number of commands. We compared this directly to the number of commands used in defining the sequence in the first place, since this is equivalent to the number of commands necessary to complete the task using a combination of lower- and mid-level skills. In all cases the use of the sequence skill drastically reduced the number of commands necessary to accomplish the task (Fig. 6.)

We measured the number of commands made by the user for each phase of the experiment. The number of commands made by the user has a direct relationship to the cognitive load encountered.

The arm trajectory plots, shown in Fig. 5, provide several interesting insights. First, for all users, the trained sequence

TABLE I
SUMMARY OF SEQUENCE EXECUTION PERFORMANCE

Starting state	Subject 1	Subject 2	Subject 3
1	X	X	X
2	X	X	X
3	–	X	X

X indicates goal state reached, **–** indicates goal state not reached.

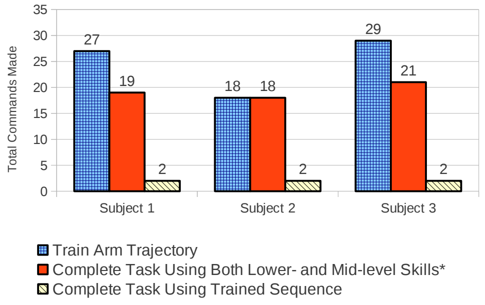

Fig. 6. **User Command Counts vs. Task** *This category is equivalent to the number of commands used to define the sequence skill.

was successfully executed regardless of starting state, with one exception. The learned policy for arm trajectories thus appears to be reasonably robust with respect to starting state.

Second, the robustness of any given arm trajectory skill appears to be a function of the amount of noise inherent in a user's original training data. For example, Subject 1 had trouble in some cases differentiating between moving the arm left versus moving it right. This directly contributed to the robot's inability to complete the assigned task when beginning in the lower starting state since it affected the shape of the learned policy in that area. Subject 2 had noisy training data near the end of the trajectory, which resulted in the robot over-shooting the target location but the learned policy was robust enough to allow the robot to autonomously accomplish the assigned task (in one case the target was slightly overshot). This suggests that the position-based trajectory learning method smoothes out some of the noise in the users' original training data. Subject 3 had near perfect control of the robot and the played-back paths of this subject's arm trajectory were the smoothest. These results imply that some inefficiencies in a user's original training example may be automatically removed through the trajectory learning method.

Fig. 6 show the results of monitoring user commands in the BCI system. Training a sequence allows the user to define the skill once but then execute it as many times as desired. First, as expected, relative to executing or training a sequence, selecting a learned command sequence requires a minimal amount of cognitive effort (2 commands in the hierarchical menu system). In comparison, when the users specified the task by selecting the arm trajectory and wrist rotation primitives individually, they used approximately 20 commands. This reduction implies a considerable savings in the number of commands necessary to accomplish a task. This directly improves the throughput of the BCI and lowers the demand on the user.

V. Summary and Conclusion

BCIs have only recently been suggested as control methods for humanoid robots. We have proposed hierarchical BCIs as a new approach to controlling humanoid robots that combines the flexibility of low-level control with the lower cognitive load of high-level control. Such an approach allows humanoid control to be tailored to the needs and the enivironment of the user on an as-needed basis.

Our results from three subjects provide a proof-of-concept demonstration that:

1) Hierarchical BCIs can be used to teach a humanoid robot new skills at multiple hierarchical levels;
2) Adding an additional level to the command hierarchy can result in a significant reduction in cognitive load;
3) Sequencing over both primitive motor commands (low-level control) and learned high-level skills can allow a user to solve complex tasks directly using brain signals.

Our ongoing efforts are focused on:

1) Extending the BCI architecture to allow learning over multiple state spaces, for instance, learning both navigational commands as well as arm/hand movements;
2) Incorporating pre-programmed robotic skills such as auto-grasping or auto-navigation to further reduce user training time and allow the user to focus on learning higher-level command sequences;
3) Utilizing more sophisticated machine learning and probabilistic reasoning algorithms to handle the uncertainty and noise inherent in brain-based robotic control;
4) Augmenting brain signals with other signals such as eye movement, voice, and muscle-based commands to explore the full-range of human biological control of humanoid robots.

References

[1] T. Nishiyama, H. Hoshino, K. Sawada, Y. Tokunaga, H. Shinomiya, M. Yoneda, I. Takeuchi, Y. Ichige, S. Hattori, and A. Takanishi, "Development of user interface for humanoid service robot system," in *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, vol. 3. IEEE, 2003, pp. 2979–2984.

[2] R. Ambrose, H. Aldridge, R. Askew, R. Burridge, W. Bluethmann, M. Diftler, C. Lovchik, D. Magruder, and F. Rehnmark, "Robonaut: Nasa's space humanoid," *Intelligent Systems and their Applications, IEEE*, vol. 15, no. 4, pp. 57–63, 2000.

[3] A. Meltzoff, R. Brooks, A. Shon, and R. Rao, "socialrobots are psychological agents for infants: A test of gaze following," *Neural Networks*, vol. 23, no. 8-9, pp. 966–972, 2010.

[4] D. Sakamoto, T. Kanda, T. Ono, H. Ishiguro, and N. Hagita, "Android as a telecommunication medium with a human-like presence," in *Proceedings of the ACM/IEEE international conference on Human-robot interaction*. ACM, 2007, pp. 193–200.

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

[6] Y. Chae, S. Jo, and J. Jeong, "Brain-actuated humanoid robot navigation control using asynchronous brain-computer interface," in *Neural Engineering, 2011. NER'11. 5th International IEEE/EMBS Conference on (to appear)*. IEEE, 2011.

[7] J. Millan, F. Renkens, J. Mouriño, and W. Gerstner, "Noninvasive brain-actuated control of a mobile robot by human eeg," *Biomedical Engineering, IEEE Transactions on*, vol. 51, no. 6, pp. 1026–1033, 2004.

[8] S. Kagami, J. Kuffner Jr, K. Nishiwaki, T. Sugihara, T. Michikata, T. Aoyama, M. Inaha, and H. Inoue, “Design and implementation of remotely operation interface for humanoid robot,” in *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, vol. 1. IEEE, 2001, pp. 401–406.

[9] N. Sian, K. Yokoi, S. Kajita, F. Kanehiro, and K. Tanie, “Whole body teleoperation of a humanoid robot-development of a simple master device using joysticks,” in *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, vol. 3. IEEE, 2002, pp. 2569–2574.

[10] J. Chestnutt, P. Michel, K. Nishiwaki, J. Kuffner, and S. Kagami, “An intelligent joystick for biped control,” in *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*. IEEE, 2006, pp. 860–865.

[11] M. Velliste, S. Perel, M. Spalding, A. Whitford, and A. Schwartz, “Cortical control of a prosthetic arm for self-feeding,” *Nature*, vol. 453, no. 7198, pp. 1098–1101, 2008.

[12] M. Chung, W. Cheung, R. Scherer, and R. Rao, “A hierarchical architecture for adaptive brain-computer interfacing,” in *Twenty-Second International Joint Conference on Artificial intelligence (IJCAI11) (to appear)*, Barcelona, Spain, July 2011.

[13] G. R. Müller-Putz and G. Pfurtscheller, “Control of an electrical prosthesis with an ssvep-based bci,” *Biomedical Engineering, IEEE Transactions on*, vol. 55, no. 1, pp. 361–364, 2007.

[14] M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, and A. Ng, “Ros: an open-source robot operating system,” in *ICRA Workshop on Open Source Software*, 2009.