Proceedings of the 2010 3rd IEEE RAS & EMBS
International Conference on Biomedical Robotics and Biomechatronics,
The University of Tokyo, Tokyo, Japan, September 26-29, 2010

Extending the Discrete Selection Capabilities of the P300 Speller to Goal-Oriented Robotic Arm Control

Garett D. Johnson, Nicholas R. Waytowich, Daniel J. Cox, and Dean J. Krusienski

***Abstract*— A brain-computer interface (BCI), a system that translates a user's brain activity into device commands, can provide a non-muscular means for disabled individuals to interact with their environment. The P300 event-related potential, a transient brain response to a sensory stimulus, has been demonstrated to be a reliable brain signal for controlling a BCI. Traditionally, P300-based BCIs have been used for simple typing tasks using a P300 Speller application, which mimics the functionality of a computer keyboard. Here we extend the discrete selection capabilities of the P300 Speller to achieve high-level control of a 6 degree-of-freedom robotic arm. This study aims to determine if a user's performance, measured in accuracy and communication rate, is affected when a P300 Speller is used to control a robotic arm compared to simple typing. The results indicate that a user's performance is not significantly affected whether typing or controlling a robotic arm.**

I. INTRODUCTION

A Brain-Computer Interface (BCI) can provide a non-muscular method to facilitate communication and device control for a person whose motor abilities may be impaired [1]. This is often beneficial for those afflicted with such disabilities as amyotrophic lateral sclerosis, spinal cord injuries, or brain-stem stroke. Advanced instances of these debilitating conditions can render patients "locked in", unable to interact with the outside world and completely dependent upon caregivers, even though normal cognitive capabilities often exist. An effective BCI can return a level of autonomy to these disabled persons by translating brain signals, in this case recorded from the scalp using electroencephalography (EEG), into computer and device commands.

The P300 Speller, originally described by Farwell and Donchin [2], is a commonly used BCI paradigm that has been extensively examined for communication purposes. Studies on healthy individuals [3], and initial studies with disabled individuals [4], [5], [6], show that the P300 Speller has potential to become a viable method for communication. This paradigm allows users to make discrete selections based on the presence of a P300 event-related potential (ERP). The P300 ERP is a positive deflection in the EEG occurring roughly 300ms after an unpredictable or novel stimulus that occurs infrequently amongst other stimuli. When using the P300 Speller, the user focuses attention on one of the symbols presented in a matrix format while the rows and columns of the matrix flash in a random order. A P300 response is elicited when the attended symbol flashes. After multiple stimulus presentations and sufficient response averaging to improve the signal-to-noise ratio, a classifier predicts which row and column contains the desired symbol. The speller then outputs the symbol at the intersection of the predicted row and column, effectively creating a keyboard for the user. The matrix used in this study is shown in Fig. 1.

Utilization of this discrete selection capability can be directly applied to goal-oriented control of a robotic arm. Discrete BCI control of an electrical prosthesis has been previously explored using steady-state visual evoked potentials (SSVEP), where online testing accuracies only ranged between 44% and 88% [7]. Additionally, the prosthesis was limited to four controlled motions. By comparison, a matrix with 36 possible selections can be consistently classified at accuracies above 90% by using the P300 Speller paradigm [3]. As such, the use of a P300 paradigm can potentially provide more control options at a comparatively high accuracy when controlling a robotic arm. In this work, a precision Staubli TX40 robotic arm with 6 degrees of freedom is controlled using a P300-based BCI. The aim of the study is to show that the P300 Speller can be used to achieve reliable, goal-oriented control of a robotic arm, and to determine if the inclusion of a robotic arm as the control device has any adverse effect on an individual's BCI performance compared to the traditional typed feedback of a spelling application. Preliminary results obtained from five able-bodied individuals are presented.

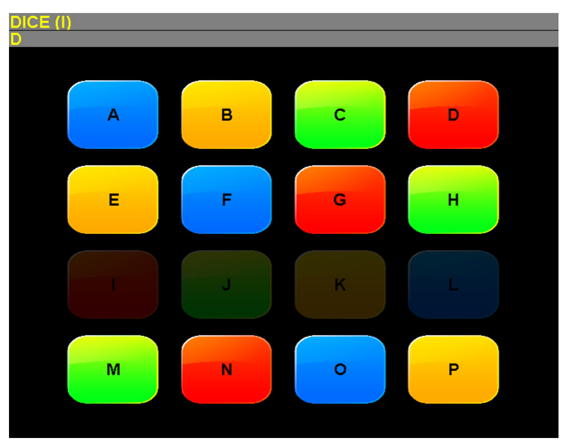

Fig. 1. The P300 Speller in Copy-Spell mode, with on-screen feedback. This mode prompts the user to spell a predefined word, in this case "DICE". The user has already correctly typed D, and should now be focused on the letter I. Since the third row is dimmed, this will likely produce the characteristic P300 response. The colored pattern is used to orient the matrix with the robotic workspace.

This work was supported by the National Institutes of Health (NIBIB/NINDS EB00856) and the National Science Foundation (0905468)

Authors are with the School of Engineering at the University of North Florida, Jacksonville, FL 32224 (corresponding author e-mail: deankrusienski@ieee.org)

978-1-4244-7709-8/10/$26.00 ©2010 IEEE

II. METHODOLOGY

A. Participants

Five able-bodied individuals (four males, one female, ages 21 to 25) participated in this study. The participants varied in their previous BCI experience. Four had little or no experience, while one had approximately fifteen previous P300 sessions. The study was approved by University of North Florida's Institutional Review Board, and each user gave informed consent.

B. Data Acquisition

The EEG was recorded using an ElectroCap International cap with 16 electrodes distributed over the scalp, based on the International 10-20 system (see Fig. 2). The EEG was amplified, bandpass filtered 0.1-60 Hz, and digitized at 256 Hz by a 16-channel g.tec biosignal amplifier. All aspects of data collection were managed by BCI2000, a general purpose system commonly used in BCI research [8]. Five sessions were collected from each subject over a three week period, the first of which was a calibration session while the subsequent four were test sessions.

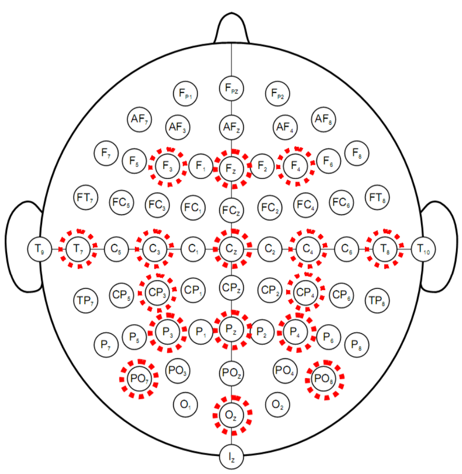

Fig. 2. The electrode montage used in the current study. The 16 electrodes used are encircled.

C. Robotic Arm Hardware and Control

1) Robotic Arm and Manipulator: The robotic arm is a six-degree of freedom, Staubli TX40 robotic manipulator arm used for high precision, industrial applications. The TX40 has a reach capacity of 0.5m and a repeatability of 0.02mm. The maximum linear movement speed of the manipulator arm is 1000mm/s, as such; the movement speed that was utilized in this study was 250mm/s. A CS8C controller, which is governed by VAL3 software, controls the robot arm. It provides kinematic and dynamic control of the robot, giving it the ability to link the tool frames to the base frame and allowing precision joint-by-joint control of the robot. The controller not only supplies power to the robot, but also loads robotic applications, executes programs, and performs other various internal system monitoring tasks. The robotic end-effector is a Schunk gripper. This gripper is comprised of a stepper motor that is powered separately from the robot through its own unique control unit. The digital signals that control the gripper are generated from digital input-output ports from the CS8C controller. The grippers are tailored to pick up a single 1-inch cylindrical plastic block, as shown in Fig. 3.

2) Communication with BCI2000: Communication between the CS8C controller and BCI2000, which is general-purpose BCI software used for stimulus presentation, data collection, and character selection, is accomplished over an Ethernet connection. Along the data-path is a custom middle-ware program written in C++, which translates the character selected by BCI2000 into ASCII code that VAL3 can interpret. Additionally, this program buffers the selected characters until the robotic arm requests the next command. The middle-ware transmits the data to the controller through an Ethernet socket. Once the controller identifies the input, it calls the corresponding subroutine and outputs the position and movement data to the robot. The CS8C controller also has internal buffering system that stacks movement commands in a command queue until the robotic arm can complete them or the program is terminated.

3) Workspace: A 12-inch square board positioned directly in front of the robotic arm serves as the workspace for the sessions. The board is segmented into a 4x4 grid of 16 color-coded squares as shown in Fig. 3. The robot is programmed to pick up individual cylindrical plastic blocks at a loading tray to the side of the robot. The robot then defaults to a center position above the workspace to where it waits for input from BCI2000. Once it receives the transmission denoting a target square, it places the block at the desired location and automatically picks up another block. The robot is also programmed with the capability to vertically stack the blocks. This functionality is necessary because of the potential for errors in classification. The robot itself does not currently contain any external sensors that are capable of detecting the position of the blocks. Instead, the VAL3 program was implemented with logic tracking to recognize if a position contains a block, and adjust movement protocol accordingly to stack the incoming block.

D. Task and Data Processing

The participants sat upright in front of both a video monitor and the Staubli TX-40 robotic arm at a comfortable viewing distance, as shown in Fig. 3. The task was to focus attention on a specified target letter of the matrix on the monitor, and silently count the number of times the target letter flashed (in this case the target icon intensity actually dimmed for a stimulus event, which will be referred to as a "flash" herein). The data in all sessions were collected in the copy speller mode, where for each character epoch the user was prompted to attend to a letter specified by the researcher and indicated in parenthesis on the top line of the P300 Speller (see Fig. 1). One character epoch consisted of 15 flashes of each row and column. The rows and columns were flashed in random order, with a single flash having a duration of 100ms, and with 75ms before the next row or column was flashed. A 5 second pause was given between each character epoch.

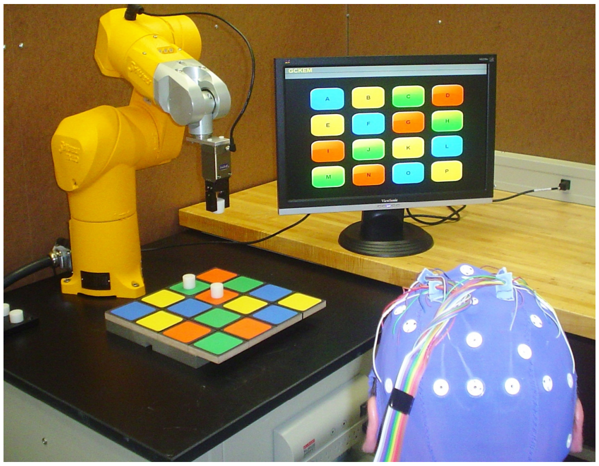

Fig. 3. The Staubli robotic arm and display setup. Note the board colors spatially correspond to the display, allowing the subject to quickly ascertain if the robot placed the block where they intended.

During the initial calibration session, the users performed 8 sets of 5 random character selections for a total of 40 character trials. No feedback was provided in this calibration session. After the users' calibration session, a subject specific classifier was constructed using Fisher's linear discriminant (FLD) [9]. For each of the 16 channels, an 800ms segment (198 samples) of data immediately following each flash was extracted. These segments were smoothed and decimated to 15Hz. The resulting data segments were then concatenated by channel, creating a feature vector corresponding to each stimulus. This resulted in a feature vector of length 176 (198/18 samples * 16 channels) for each of the 4,800 stimuli. Of these stimuli, 1,200 were target flashes and 3,600 were non-targets. The target stimuli were assigned a class label of +1, and non-target stimuli were labeled as -1, creating a binary classification problem. The weights generated by the FLD solution, which are the same as the ordinary least-squares regression for a binary case, are computed as:

$$w = (X^T X)^{-1} X^T y \tag{1}$$

where X is the matrix containing the feature vectors for all 4,800 stimuli, and y is the vector of class labels. By taking the scalar product of these weights and the feature vectors, and selecting the resulting row and column stimulus that has the largest scalar product across each character epoch, we can predict which character the user is attending. Thus, the predicted row and column are selected as:

$$predicted\,row = \max_{rows}\Big(\sum_{i_{row}} w \cdot x_{i_{row}}\Big) \tag{2}$$

$$predicted\,column = \max_{cols}\Big(\sum_{i_{col}} w \cdot x_{i_{col}}\Big). \tag{3}$$

The weights generated from the calibration session were used to provide feedback for the four remaining sessions. The four subsequent test sessions also consisted of 40 character trials apiece. Each session could be considered to consist of 2 half-sessions, where in one half the user performed copy spelling with just the matrix on the video screen. The robotic arm was not activated for these runs, and the feedback in the form of the character predicted by the classifier was provided to the user on the monitor in the space provided immediately under the text to spell (see Fig. 1).

In the other half of the session, the 16 color-coded squares on the monitor spatially correspond to the 16 color-coded squares in the robot's workspace. The user would not be provided feedback on the monitor, but instead the robotic arm would move a small cylindrical block from a fixed position adjacent to the workspace onto the square in the workspace corresponding to the square on the monitor predicted by the classifier. Although all specified target locations were unique for each run, the robot was programmed with the capability to vertically stack blocks if a location was erroneously selected multiple times within a run. With 5 seconds between character trials, the participants had ample time to ascertain if the location of the placed block was correct. The ordering of the half-sessions was counterbalanced across sessions to mitigate effects of order on performance.

III. RESULTS

The online prediction accuracy averaged over the four test sessions for each subject and feedback method is shown in Table 1. Although the final accuracy when using the robotic arm is 3.5% lower than the standard method, where feedback is displayed on a video monitor, an analysis of variance on the online accuracies did not reveal any statistically significant difference between the two methods ($p = 0.185$). Fig. 4 shows the percentage of correctly classified targets, averaged across all users and test sessions, for each feedback method as a function of the number of flashes. This offline analysis suggests that the performance difference also is consistent for fewer flashes.

TABLE I

ONLINE AND OFFLINE PERFORMANCE FOR EACH SUBJECT

	Standard Feedback			Robotic Arm Feedback		
Subject	Accuracy	Max Bitrate	Flashes	Accuracy	Max Bitrate	Flashes
A	85.0	8.2	7	80.0	7.1	11
B	98.8	14.6	4	97.5	15.4	4
C	100	13.1	7	96.3	13.1	7
D	90.0	10.1	5	81.3	7.7	10
E	58.8	3.9	8	60.0	3.7	9
AVG*	**86.5**	**9.2**	**7**	**83.0**	**8.1**	**7**

* Maximum bitrate and the corresponding number of flashes are based on the average accuracy accross users.

Another common metric used to evaluate BCI performance is the bitrate, which accounts for speed as well as accuracy. During actual use, although the highest accuracy may be achieved after 15 flashes, this number of flashes could be suboptimal in terms of bitrate. This can be remedied by adjusting the system to make a prediction after fewer flashes, when the bitrate is highest for a given individual based on offline analysis. The formula for computing the number of bits communicated per character epoch is:

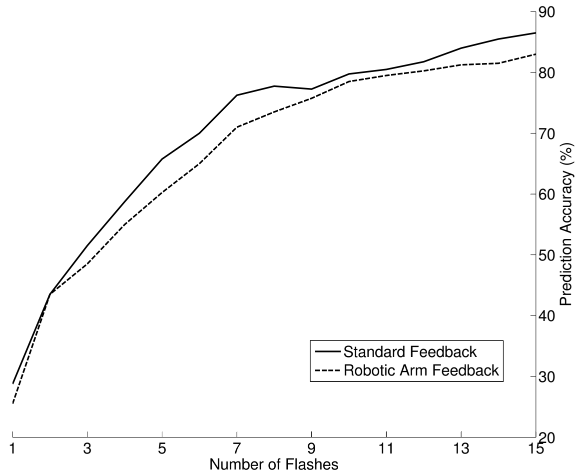

Fig. 4. The symbol prediction accuracy averaged across all 5 subjects and test sessions. On average, the accuracy when using the robotic arm is marginally less than when using the standard video feedback through most of any given character trial.

$$B = \log_2 N + P \log_2 P + (1-P) \log_2 \left(\frac{1-P}{N-1} \right) \quad (4)$$

where N is the number of possible targets and P is the probability that the target is accurately classified [10]. The bitrate is then B divided by the character epoch duration. Fig. 5 shows the bitrates computed from the accuracies shown in Fig. 4. Table 1 provides the maximum bitrate, based on the average accuracy across test sessions, and the number of flashes required to obtain this bitrate for each user. As with the accuracy results, the bitrate results only indicate a marginal difference between feedback methods.

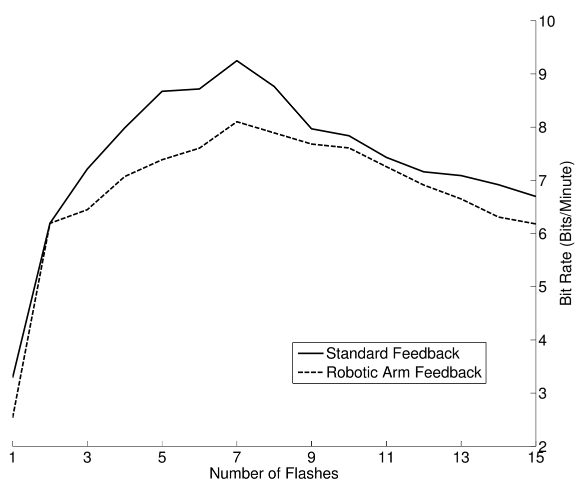

Fig. 5. The bitrate curves corresponding to the accuracies in Fig. 4.

IV. DISCUSSION

The study demonstrates an important initial step toward implementing a practical P300-based robotic arm system by examining the relative performance of using a robotic arm as the control device for the P300 Speller. The results indicate that the P300 Speller paradigm can be readily extended from traditional typed communication to high-level robotic arm control. More importantly, this can be accomplished without any significant effects on a user's performance. Although there is an overall 3.5% degradation in accuracy from the standard feedback method, neither this nor the bitrate results indicated statistically significant differences between the performance of the methods. Given that the task was identical for both feedback methods, the results suggests that robotic feedback did not create enough additional workload or distraction to adversely impact performance.

It should also be noted that when controlling a robotic arm to position physical objects in a workspace, the overall accuracy is arguably more important than a higher communication rate. When interacting with practical objects, rather than simple plastic cylinders, selection errors can lead to collisions in the workspace that can damage the objects, relocate them to undesirable positions, and/or lead to safety concerns for the user, particularly if they are disabled. Nevertheless, the system must also be fast enough for practical use. To accomplish this, various high-level macro functions, similar to the macro functions used to pick and place the blocks in this study, can be assigned to the individual P300 Speller selections. When a single P300 Speller selection is made, the robot can be employed to perform a complex interaction with specific objects and environments. This approach is substantially faster and more practical than associating the P300 Speller selections to individual robotic arm joint angles, for instance. However, strictly predefined macro functions can be limiting in terms of providing the freedom to achieve arbitrary trajectories or interact with arbitrary objects in the workspace. Therefore, a vision system will be incorporated to give the system a level of intelligence and flexibility for interacting with novel objects and environments.

REFERENCES

[1] J. R. Wolpaw, *et al.*, "Brain-computer interfaces for communication and control," *Clin Neurophysiol*, vol. 113, pp. 767-91, Jun 2002.

[2] L. A. Farwell and E. Donchin, "Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials," *Electroencephalogr Clin Neurophysiol*, vol. 70, pp. 510-23, Dec 1988.

[3] D. J. Krusienski, *et al.*, "Toward enhanced P300 speller performance," *J Neurosci Methods*, vol. 167, pp. 15-21, Jan 15 2008.

[4] F. Nijboer, *et al.*, "A P300-based brain-computer interface for people with amyotrophic lateral sclerosis," *Clin Neurophysiol*, vol. 119, pp. 1909-16, Aug 2008.

[5] E. W. Sellers and E. Donchin, "A P300-based brain-computer interface: initial tests by ALS patients," *Clin Neurophysiol*, vol. 117, pp. 538-48, Mar 2006.

[6] T. M. Vaughan, *et al.*, "The Wadsworth BCI Research and Development Program: at home with BCI," *IEEE Trans Neural Syst Rehabil Eng*, vol. 14, pp. 229-33, Jun 2006.

[7] G. R. Muller-Putz and G. Pfurtscheller, "Control of an electrical prosthesis with an SSVEP-based BCI," *IEEE Trans Biomed Eng*, vol. 55, pp. 361-4, Jan 2008.

[8] G. Schalk, *et al.*, "BCI2000: a general-purpose brain-computer interface (BCI) system," *IEEE Trans Biomed Eng*, vol. 51, pp. 1034-43, Jun 2004.

[9] D. J. Krusienski, *et al.*, "A comparison of classification techniques for the P300 Speller," *J Neural Eng*, vol. 3, pp. 299-305, Dec 2006.

[10] D. J. McFarland and J. R. Wolpaw, "EEG-based communication and control: speed-accuracy relationships," *Appl Psychophysiol Biofeedback*, vol. 28, pp. 217-31, Sep 2003.