Multitask Humanoid Control with a Brain-Computer Interface: user experiment with HRP-2

Pierre Gergondet, Abderrahmane Kheddar, Christoph Hintermüller,
Christoph Guger, Mel Slater

HAL Id: lirmm-00781275
https://hal-lirmm.ccsd.cnrs.fr/lirmm-00781275v2

Submitted on 3 Sep 2014

Multitask Humanoid Control with a Brain-Computer Interface: user experiment with HRP-2

Pierre Gergondet[1,2], Abderrahmane Kheddar[1,2], Christoph Hintermüller[3], Christoph Guger[3], and Mel Slater[4,5]

[1] UM2-CNRS LIRMM UMR5506, Montpellier, France
[2] CNRS-AIST Joint Robotics Laboratory (JRL), UMI3218/CRT, Tsukuba, Japan
[3] g.tec Guger Technologies OG, Austria
[4] ICREA - University of Barcelona, Spain
[5] University College London, United Kingdom
   `pierre(dot)gergondet(at)aist.go.jp`

Summary. In this paper, we present our approach to design a brain-computer interface (BCI) that allows the user to perform multitask humanoid control. We efficiently integrate techniques from computer vision and the task-function based control together with the brain-computer interface into an immersive and intuitive control application despite the well-known shortcomings of BCI. This approach is assessed in a user experiment involving 4 subjects who successfully controlled the HRP-2 humanoid robot in a scenario involving both grasping tasks and steering. The user experiences and the interface performances are presented and give a rich insight into future research that can be made to improve and extend such interface.

1 Introduction

Brain-computer interfaces (BCI) [1] allow bypassing the usual communication channels between a human and a computer such as hand or voice input interfaces. Instead, they allow the users to communicate his *intentions* to the computer. In return, the user is able to control different application (software or device) systems connected to the BCI. Recent work has already demonstrated impressive capability for controlling mobile robots, virtual avatars or humanoid robots [2].

However, the use of BCI in those previous works has been limited to the accomplishment of a single task, for example: steering a robot. Our work attempts to make more tasks available to the user. It originates from the VERE project, which aims at embodying the user's conscience into a virtual avatar or a physical robot. Therefore, we aim at allowing the user to perform a wide range of actions with an emphasis on liberty and reactivity. The scenario

that is demonstrated in this paper illustrates the approach we have taken and combines both locomotion and manipulation tasks, that is whole-body motion.

Using the well-known brain pattern: steady-state visually evoked potentials (SSVEP), we allow the user to perform humanoid whole-body control. We efficiently integrate techniques from computer vision and the task-function based control. We propose to use task-function primitives as affordances on objects of interest detected through the robot's embedded cameras. The user is fed with direct vision feedback from those cameras. Our main contributions and novelties are the following:

- By using well-known techniques from image processing, objects of interest are detected within the scene and automatically blinked at different frequencies. SSVEP allows identifying which object is of the user's interest.
- Integrating BCI and task-based control allowing instant and smooth task integration in the controller.
- A transition state machine proposes switching between whole body manipulation and locomotion tasks. During locomotion, SSVEP is also used to choose direction and speed of locomotion during which visual feedback is continuously displayed to the user.
- Our approach is assessed in real experiments using the HRP-2 robot controlled from an electroencephalography (EEG) cap and g.BCIsys (g.tec medical engineering GmbH, Austria). The scenario of these experiments allows the user to achieve multiple tasks.

2 Technical approach

This section introduces three major components of our system: (i) the brain-computer interface, (ii) the stack-of-tasks (SoT) controller for the robot, and (iii) the robot visual perception system. The integration of these components is also a key to the extended capacities of our system. This integration occurs at two levels of interaction: between the visual system and the BCI on the one hand, and between the BCI and the SoT controller on the other hand.

2.1 Brain-Computer Interface

In recent years, several frameworks such as OpenViBE or BCI2000 have introduced a similar three-layer model to produce BCI application as shown in Figure 1.

The signal acquisition layer monitors the physiological signals from the brain through one or several physical devices and digitizes these signals to pass them onto the signal-processing unit. The signal-processing unit is in charge of extracting features —e.g. power spectrum, signal energy— from the raw signals, and pass them onto a classification algorithm to distinguish the *intentions* of the user. Finally, these decoded intentions are passed onto the user application.
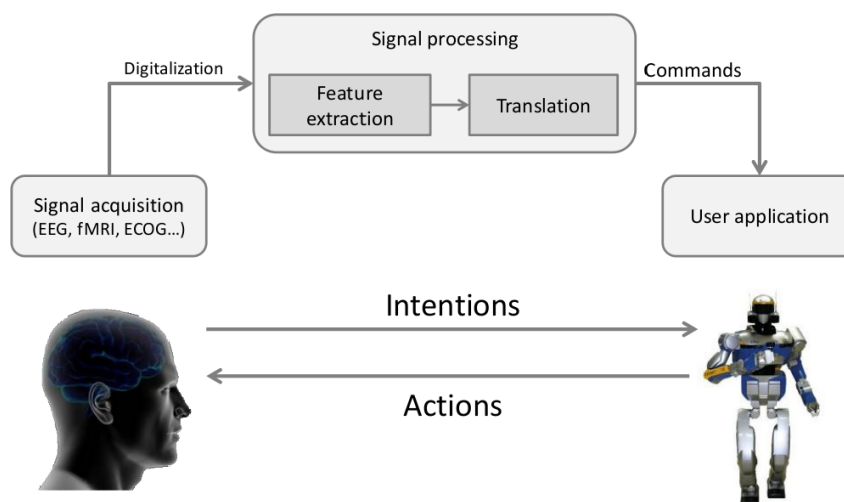
Fig. 1. General design of a BCI system

In this work, we employ the widely used electroencephalography (EEG) technique for signal acquisition because it is non-invasive, cheap and allows for real-time acquisition even though it suffers from poor spatial localization accuracy and a poor signal to noise ratio.

The brain features we decided to exploit are the steady-state visually evoked potentials (SSVEP). The SSVEP describe the activities that the brain generates when the user observes a flickering stimulus. The method relies uniquely on the user's attention to the stimulus. It also allows detecting that the user is maintaining his attention on a given stimulus and to detect a shift of attention in a few seconds. The process we used to extract the SSVEP is based upon the minimum energy classifier approach introduced in [3]. It provides a zero-class implementation that allows to detect that the user is not interested in interacting with the system. After a short training, about 6 minutes, it is able to operate at an 80% recognition rate [4] for 4 classes and can provide a new command every 200ms. This is a satisfactory performance for an SSVEP-based BCI system [5].

Recursive and Enforced SSVEP Selection

As mentioned previously, our SSVEP extraction process can reach an 80% successful recognition rate with a short training phase. However, given the nature of the SSVEP stimuli, errors are bound to happen over the course of the experiment due to distraction or fatigue [6]. Therefore, the 100% mark would be difficult to reach. This unreliability of the decision outcome becomes an important problem when using the BCI to control an avatar, especially if

the decision that was taken cannot be reversed. Moreover we aim at minimizing the number of frequencies that we can detect to ensure high accuracy while keeping the training time as short as possible. Therefore we devised an SSVEP-based selection paradigm that allows for a large number of commands and puts an emphasis on accuracy.

To increase the number of commands we used a simple recursive selection algorithm. For example, if we have sixteen commands available but only trained the system to detect four different frequencies we split each command into four groups. The user then selects one of these groups and finally selects a command among the four commands in this group.

To enforce the selection we ask the user to maintain his attention on the command he wants to select for a certain time. We consider this command as the actual intention of the user only if he was able to maintain his attention "long enough', which in our case means three seconds, i.e. fifteen successive classifications.

User Interface

We also developed an application framework for the user interface that allows practical switching between different BCI paradigms, thus allowing multiple tasks control with our BCI system.

The framework relies on a core class: BCIInterface. Its role is to create the graphical interface, handle and distribute events and get the different primitives of the library working together. As illustrated in Figure 2, the main graphical loop executes the following operations for each frame:

1. Get and handle events, pass unhandled events to DisplayObjects and CommandInterpreter
2. Receive the current command from the BCI thanks to a CommandReceiver instance
3. Give this command to a CommandInterpreter instance that will update the controlled system status and/or the DisplayObjects to give feedback to the user according to its implementation
4. Display a Background instance and then display all DisplayObjects

Additionally, the elements that require synchronous operations to work - e.g. a CommandReceiver that receives commands over the network - can implement another loop; this loop is run in a separate thread to have no impact on the display loop performance and it does not require any extra work from the developer. A paradigm switch can be triggered by the owner of the BCIInterface instance or by the CommandInterpreter.

2.2 Stack of Tasks controller

The task-function based control is a powerful control paradigm to design complex behaviors for robots without explicit trajectory planning at the joint or
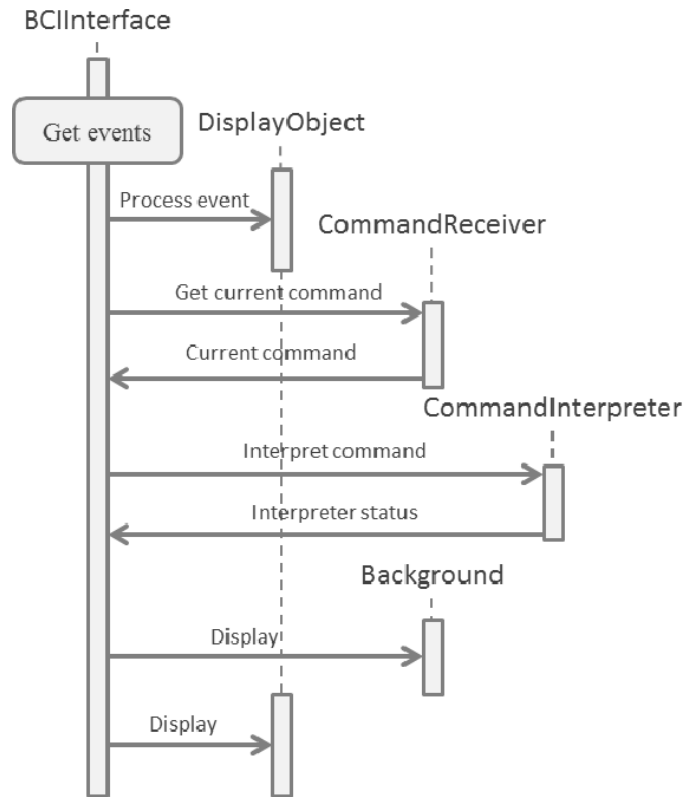
Fig. 2. A display loop iteration by the BCIInterface core

Cartesian level. A task can be seen as motion primitives or constraints that can be defined directly or indirectly in the robots sensory space. The desired task can be defined simply as a state or a state error vector in the sensory space, which is mapped into the robots motor space (joint actuators) using an appropriate projection operator, e.g. the robot's jacobian in kinematics. A complex behavior can then be planned as a succession of tasks, which can in turn be structured into a hierarchical way: a stack of tasks. This formalism proved to be particularly suited for the control of highly redundant robots; we have demonstrated several complex scenarios using this powerful control tool with our HRP-2 humanoid robot [7]. We have also extended this controller, called simply stack-of-tasks (SoT), to address issues such as tasks scheduling and the fundamental issue of control continuity under discreet tasks scheduling operations in the SoT, such as tasks insertion, removal and swapping. Our controller is able to insert or remove tasks components, on the fly, and we can adapt it to be used in a BCI control context.

2.3 Vision System

Our vision system is built around a lightweight core that operates different cameras and plugins aimed at realizing a specific task such as transmitting images over the network or recognizing objects within the scene.

Objects Recognition

The object recognition method is based on the work presented in [8] and its extension to account for color properties of objects in [9]. This method relies on the construction of a vocabulary set of texture-related features and color-related features. These features are trimmed down to a restricted set through a k-means clustering algorithm, associated to the relevant objects and organized in a kd-tree for efficient closest neighbor research needed by the recognition algorithm. This allows this method to scale very well as when the objects database grows, the vocabulary itself does not grow but evolves and enriches itself.

The recognition algorithm then consists in (i) extracting interest points in the scene, (ii) computing color and texture features at those interest points, (iii) match those features with the ones from the vocabulary, (iv) each feature from the vocabulary will then cast a vote for the relevant object - this voting mechanism is further explained afterward - (v) the object presence is decided based upon its best score value. The score each object will give is determined from a training set where the algorithm knows the objects present in the scene. The votes are computed so that the more specific to an object a feature is the bigger vote it will cast. For example, if the same feature is extracted from 10 images with 10 different objects in the scene it will contribute a 0.1 vote to each object. However, among the same set, if a feature is found only twice, it will cast a 0.5 vote for the two objects involved in these two scenes. Finally, a second pass over the training set allows us to define threshold score above which the object presence is assumed. Thanks to the sharing of features among different objects, the recognition algorithm can operate very efficiently, above 15 Hz even in rich scenes, and the features selected for classification permits a consistent detection of the objects.

Shape Extraction

In order to allow a smooth integration of the detected objects within the user interface, we extended this method to detect the objects' shapes.

To do so, we need to collect more data when we build the vocabulary as well as during the course of the algorithm. During the training run, when we create a link between a feature and the object, we register two information from the training image: the shape of the object and the position of the feature point. During the recognition process, we maintain a list of all voters for each

object. For each voter, we register the point of interest position in the image being processed, the link it is associated with and where the vote is casted.

Once an object and its center have been detected, we extract the voters that voted for the object around the object's center. Each voter is ultimately linked to an object's mask from the vocabulary set through the link it holds. For each of these masks, we compute an homography from the points in the vocabulary set to the matched points in the processed image. The points are filtered before the homography computation. If multiple points in the processed image match the same point in the vocabulary set, we keep the one for which the distance between the point and the center of the object is closest to the same distance for the vocabulary point. The same selection criterion is applied when the same point in the processed image matches different points in the vocabulary set, which can happen when the texture match and the color match are different points from the same image of the vocabulary set. The vocabulary object that retains the most matching points in the end is selected and its shape is deformed thanks to the computed homography which allows us to match the shape of the object in the processed image.

2.4 BCI and Visual System Integration

When the user is interacting with the world through the robot he needs to be informed about the capacities of the robot and the interaction it can operate with the world. To do so we integrate visual stimuli on top of the video feedback fed to the user. We can distinguish two kinds of integrated stimuli: static ones and dynamic ones. On the one hand, the static stimuli are relative to the current operation mode of the robot. In visual exploration mode they will consist in flickering arrows to control the robot's gaze, while in steering mode a similar set of arrows will control the robot's speed. On the other hand, the dynamic stimuli are controlled by the visual system. This allows presenting the user with the objects he will be able to interact with. A flickering stimulus is shown at the detected center of the object, communicated by the vision system. Their color is chosen to fit the recognized object. The result can be observed in Figure 3 and in the experiment video linked below.

2.5 Tasks selections with a BCI

Contrary to other interfaces used to control a robot, EEG-based BCI is a rather unreliable and very slow interface. In other words, the user will not be able to guide efficiently an arm to grasp an object in a certain position and in a particular way. What s/he will be able to request however is an action that results from an association of the recognized object of intention and the afforded task (or tasks). For example, "grasp object A, given that this "object A has been detected and localized in the real world via the visual system and presented to the user as explained previously. This naturally translates into considering tasks as affordances on the environment objects.
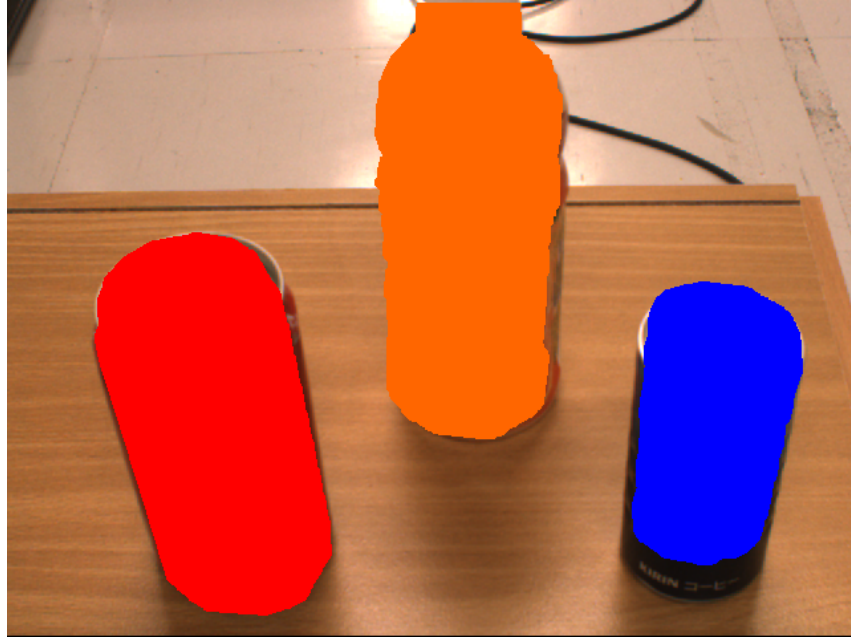
Fig. 3. Interface for object grasping: images are streamed from the vision server, object-related stimuli are positioned thanks to the objects detection plugin

This approach is reminiscent of the shared-control approach that has been often used in BCI-based control since it allows the user to perform the same task better with less input commands. However, the system does not only rely on its own intelligence to perform the task based on a limited input, it also shares its knowledge of the environment with the user to present affordable tasks to him, that is what we refer to as shared intelligence.

3 Experiment Scenario

The scenario is designed to illustrate multitask control, i.e. locomotion and manipulation, via BCI.

At first, the user is presented, through the robot's eyes, with multiple objects - known from the recognition system - put on a cupboard. In this phase, the user selects the object he wishes to interact with. The selection relies on an SSVEP paradigm: the objects are blinked as discussed in the previous section. Once this selection happens, the robot grasps the object and the second phase of the experiment begins.

In the second phase, the user steers the robot freely in its environment to a location of his choice. The steering is done through SSVEP. Three stimuli allow the user to control the robot orientation and make it move forward, a

fourth one allows him to stop the robot. Once the robot is stopped, the final phase begins.

In the final phase, the user recursively selects a position within the visual field of the robot. The selection of a position makes the robot drop the object above this position thus achieving the experiment.

To confirm the usability of the interface, the user is given two instructions that define his mission: (a) which object to pick up from the cupboard and (b) which sixteenth of the screen to select at the end. To evaluate the objective performances of the interface we measure the time taken to achieve each phase and the success rate of the missions.

4 Experiments

4.1 Material and System Setup

We use a g.USBamp (24 Bit biosignal amplification unit, g.tec Medical Engineering GmbH, Austria) to acquire the EEG data from the user's brain at a sampling frequency of 256 Hz, bandpass filtered between 0.5 and 30 Hz with a notch filter at 50 Hz to get rid of the power line noise. The electrodes positioning is shown in Figure 4. We use 8 Ag/AgCl active electrodes. The electrodes are placed on the POz, PO3, PO4, PO7, PO8, O1, O2 and Oz positions of the international 10-20 system [10], Fpz is used as the ground electrode and the earlobe as a reference.
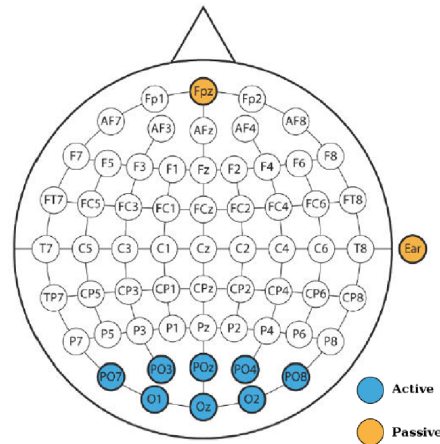

Fig. 4. Electrodes positions for the experiment

The experiment was carried out using the HRP-2 humanoid robot. The subject, equipped with an EEG cap, is comfortably seated in an armchair, about 1 meter away from of a 17" LCD screen. In such setup the accurate display of SSVEP stimuli is ensured thanks to the method proposed in [11]. The SSVEP stimuli frequencies that were used in this work are: 6, 8, 9 and 10 Hz. Those were carefully selected to have neither common first or second harmonics and are below 20 Hz to minimize the risk of eliciting an epileptic crisis in healthy subjects as advised in [12].

4.2 Results

A video showing the interface in action as well as the robotic counterpart can be retrieved at the following URL: http://pbs.dinauz.org/Videos/ISER-2012.avi.

Four users performed the scenario that we described. Each of them performed the scenario five times with different conditions, i.e. different objects and different drop locations. Over the 20 trials, the users' selections were consistently accurate thanks to the enforcing of SSVEP selection we setup. In Figure 5 we report their average performance over the multiple trials.
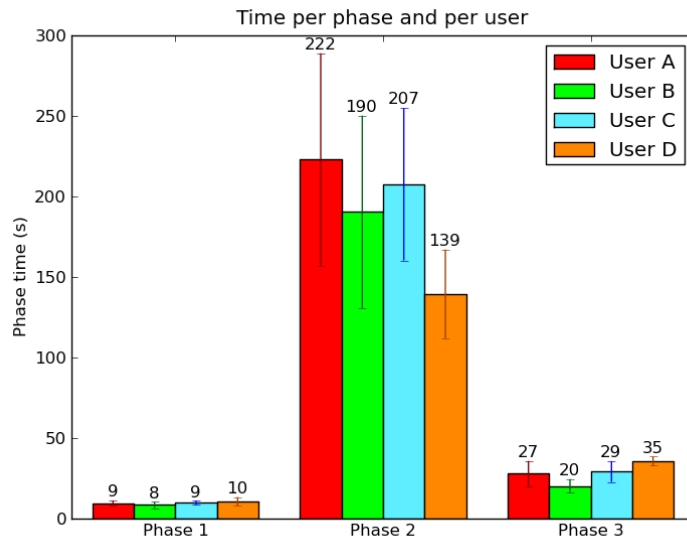

Fig. 5. Average performance of each user over the experiment trials

The phase 1 and phase 3 times are consistent with the performance of our SSVEP classification method and the adoption of the enforced SSVEP selection process. The system operates on a 3 seconds window of EEG data and we require the user to maintain his attention on the stimulus for 3 seconds before we make a conclusion about his intention. The phase 3 involves the recursive SSVEP paradigm described earlier and thus the time needed to reach a conclusion in phase 3 is about twice the time needed in phase 1.

The enforced SSVEP selection paradigm also proved its usefulness during the trials as no misinterpretation of the user's intentions occurred during these phases for all subjects across all trials. However, we chose to enforce the selection a 3 seconds period which covers 15 decisions by the SSVEP extraction process. To achieve a more reactive experience, this activation time could be tuned down according to the classifier performance with the user. This would allow the system to reach a conclusion more rapidly while keeping a very high-level of accuracy.

Finally, the phases 2 times illustrate interesting effects of training and motivation on the scenario performance. The performance of each subject over each trial can be seen in Figure 6. It shows that the user's performance in the navigation phase improve after each trial. Two factors, reported by the users, can explain this phenomenon. On the one hand, the users felt more and more comfortable with the interface. They progressively learned how to compensate for the lag of the SSVEP extraction process and they also acquainted themselves with the environment and the way it is perceived through the robot's camera. Indeed, this camera has a rather constrained field of view which is well suited for the stereo application it is used for but makes it particularly difficult for the user to understand the robot's location in space. On the other hand, the user also used more 'aggressive' strategy to reach their destination because they felt more comfortable steering the robot but most importantly because they wished to 'beat' their previous time which illustrates the importance of motivation in the context of BCI applications [13].

5 Future Works

The success of the experiment and positive feedback from the users confirm the viability of the concepts presented throughout this paper. This section discusses possible improvements and issues left unaddressed in this work regarding the problem of task selection and parameterization through BCI, and the navigation in unknown environment.

5.1 BCI and Tasks Selection

This work addresses the issue of mapping tasks to environment objects by conceiving tasks as affordances on objects. From the control feedback and
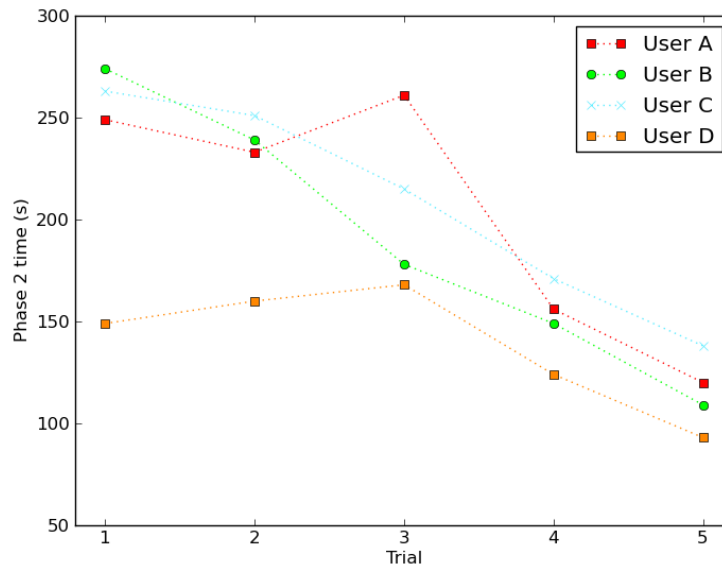
Fig. 6. Walking phase performance for each subject and trial

visual perception we can infer a feasible task among those attached to the objects. However, two main issues are left to be dealt with: task selection among many and task parameterization.

We are now able to understand that the user wants to interact with an object. If we wish to pursue the embodiment of ones conscience into an avatar, we also have to be able to understand *why* the user wants to interact with this object as the visual and control information may not be enough to conclude which afforded task should be selected. A good example to illustrate this issue is the one of a bottle of water. If we want to pour some of its content into a glass we will not grasp it like if we want to put this bottle in a bag. However, in the current state of this work, such a distinction cannot be done, as we are only able to understand that the user wants to "interact with the bottle". This is the problem of task selection among many.

The second unaddressed issue regarding the task selection concerns the task parameterization. The problem here is to select the appropriate gain, i.e. speed of execution, and variant of the task, e.g. grasp with left or right hand. Addressing these problems is difficult, as it requires a much deeper understanding of the user's intention than current EEG-based BCI permits. A possible solution may rely on the measurement of user's satisfaction as the robot executes the demanded task. Measuring the stress of the user for example could be used to modulate the gain of the task while understanding the EEG features related to user's satisfaction will allow to cancel the current

task and try another one to fit the user's expectation. However, from the robot's control perspective, task backtracking is also an open and complex problem that will have to be tackled to allow this kind of parameterization.

5.2 Landmark-assisted navigation

The phase 1 and phase 3 of the experiment's scenario are good examples of the benefits of shared-control in BCI control application. Within a few second of selection, the user can command the execution of complex tasks by the robot while 'relaxing'. In phase 2, the user continuously commands the robot to guide to its final position. The benefit of such method is that it allows the user to reach any position he wants in the robot's environment. However, it is also tiresome for the user, especially over long period of control, and the speed of the robot has to be limited to allow a fine control by the user.

To improve this situation, we will investigate an hybrid scheme of navigation control that mixes full control approach and shared control approach. In this navigation scheme, the object recognition module presented in this paper is used during navigation to detect known objects in the robot's environment, these objects are then proposed for selection to the user as in the first phase of the scenario. If the user selects an object, the 3D model of this object can be tracked in the scene [14] to approach it efficiently without further input from the user. Once the object has been reached, the user once again controls the robot either to reach its final destination or to find another landmark to go to.

6 Conclusion

We presented our method to control a humanoid robot through a brain-computer interface within a multitask scenario. The key of this work is to not rely on a pre-defined set of task and thus differs from classical approach in BCI control application. In place, the tasks are dynamically constructed from the tight collaboration of the visual system, control architecture and user's intention extraction process and are presented to the user in a user-friendly way together with the visual feedback of the robot. In a scenario we designed to illustrate these concepts, four users were able to successfully control a HRP-2 humanoid robot in a multitask scenario.

Future works will focus on resolving the many issues raised during the development of this new architecture in the three different fields it involves. We believe that, extending upon this work, we can provide strong solutions to the problems of user embodiment and whole-body control through brain-computer applications.

Acknowledgment

References

1. Jonathan R. Wolpaw, Niels Birbaumer, Dennis J. McFarland, Gert Pfurtscheller, and Theresa M. Vaughan. Brain-computer interfaces for communication and control. *Clinical Neurophysiology*, 113:767–791, 2002.

2. José del R. Millán, Rudieger Rupp, Gernot R. Müller-Putz, Roderick Murray-Smith, Claudio Giugliemma, Michael Tangermann, Carmen Vidaurre, Febo Cincotti, Andrea Kübler, Robert Leeb, Christa Neuper, Klaus-Robert Müller, and Donatella Mattia. Combining Brain-Computer Interfaces and Assistive Technologies: State-of-the-Art and Challenges. *Frontiers in Neuroprosthetics*, 4:161, 2010.

3. Ola Friman, Ivan Volosyak, and Axel Gräser. Multiple Channel Detection of Steady-State Visual Evoked Potentials for Brain-Computer Interfaces. *IEEE Transactions on Biomedical Engineering*, 54(4):742–750, April 2007.

4. Robert Prueckl and Christoph Guger. A brain-computer interface based on steady state visual evoked potentials for controlling a robot. *Lecture Notes in Computer Science*, 5517:690–697, 2009.

5. François-Benoît Vialatte, Monique Maurice, Justin Dauwels, and Andrzej Cichockia. Steady-state visually evoked potentials: Focus on essential paradigms and future perspectives. *Progress in Neurobiology*, 90:418–438, 2010.

6. Danhua Zhu, Jordi Bieger, Gary Garcia Molina, and Ronald M. Aarts. A Survey of Stimulation Methods Used in SSVEP-Based BCIs. *Computational Intelligence and Neuroscience*, 2010:12, 2010.

7. Nicolas Mansard, Olivier Stasse, Paul Evrard, and Abderahmane Kheddar. A versatile generalized inverted kinematics implementation for collaborative working humanoid robots: The stack of tasks. In *Int. Conf. on Advanced Robotics (ICAR)*, 2009.

8. Erik Murphy-Chutorian and Jochen Triesch. Shared Features for Scalable Appearance-Based Object Recognition. In *IEEE Workshop on Applications of Computer Vision*, 2005.

9. Erik Murphy-Chutorian, Sarah Aboutalib, and Jochen Triesch. Analysis of a Biologically-Inspired System for Real-time Object Recognition. *Cognitive Science Online*, 3.2:1–14, 2005.

10. F. Sharbrough, G.-E. Chatrian, RP. Lesser, H. Lüders, M. Nuwer, and TW. Picton. American Electroencephalographic Society guidelines for standard electrode position nomenclature. *Journal of Clinical Neurophysiology*, 8:200–202, 1991.

11. Yijun Wang, Yu-Te Wang, and Tzyy-Ping Jung. Visual stimulus design for high-rate SSVEP BCI. *Electronic Letters*, 46:1057–1058, 2010.

12. Robert S. Fisher, Graham Harding, Giuseppe Erba, Gregory L. Barkley, and Arnold Wilkins. Photic- and pattern-induced seizures: a Review for the Epilepsy Foundation of America Working Group. *Epilepsia*, 46:1426–1441, 2005.

13. Robert Leeb, Felix Lee, Claudia Keinrath, Reinhold Scherer, Horst Bischof, and Gert Pfurtscheller. Brain-Computer Communication: Motivation, aim and impact of exploring a virtual apartment. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 15:473–482, 2007.
14. A.I. Comport, E. Marchand, M. Pressigout, and F. Chaumette. Real-time markerless tracking for augmented reality: the virtual visual servoing framework. *IEEE Transaction on Visualization and Computer Graphics*, 12(4):615–628, July 2006.