An integrated framework for humanoid embodiment with a BCI

Damien Petit, Pierre Gergondet, Andrea Cherubini, Abderrahmane Kheddar

An integrated framework for humanoid embodiment with a BCI

Damien Petit[1,2] Pierre Gergondet[1,2] Andrea Cherubini[2] Abderrahmane Kheddar[1,2]

Abstract— **This paper presents a framework for assisting a disabled user to control a humanoid robot with a brain-computer interface (BCI). With our framework, the robot can interact with the environment, or assist its user. The low frequency and accuracy of the BCI commands is compensated by vision tools, such as objects recognition and mapping technics, as well as shared-control approaches. As a result, the proposed framework offers intuitive, safe, and accurate robot navigation towards an object or a person. The generic aspect of the framework is demonstrated by two complex experiments, where the user controls the robot to serve him a drink, and to raise his arm.**

I. Introduction

Assistive robotics aims at improving the life of elder or physically disabled people in rehabilitation therapy. A wide variety of applications involve robots aiding persons in domestic tasks. For example, a mobile robot can be used to socially interact with elderly people [1], or, a humanoid robot can autonomously retrieve an object in a smart environment on demand [2].

Brain Computer Interfaces (BCI) allow to bypass the usual motor pathways of communication between user and assistive devices, including robots [3]. Humanoid robot surrogates controlled by such technology [4] can help users with severe motor disabilities to regain a certain level of autonomy, by being able to interact with the environment and other people. Being able to control a robotic avatar and perceive the world through and as this avatar covers the are of *embodiment*, a topic which is actively studied within the FP7 European Project VERE[a] within which this work takes place.

Decoding human motor intentions, in order to use them as an input signal for the control of a robot, is a challenging and open issue. In assistive robotics, electromyography (EMG) signals, which result from the activation of muscles, are often used since the interpretation of motor intention is rather accurate [5]. However, in the case of patients with severe motor disability, EMG is not available, thus, electroencephalography-based (EEG-based) BCI are preferable. However the decoding of motor intentions in EEG activity has not been achieved yet. Therefore it is better to perform a goal-oriented control [6] using high-level intentions recognition.

Many assistive applications have been developed using a BCI [7]. For example, a complex gripper can be controlled via BCI with an specifically designed user interface to realize a grasp [8]. Many vehicles, such as a wheelchair [9] or a virtual plane [10], have also been controlled with a BCI. These different works are either specialised in navigation or interaction with objects. Instead, in our work, we propose a vision-based semi-autonomous control framework, for realizing both navigation and interaction, with a BCI-controlled humanoid robot.

Recently, we proposed a BCI-based controller enabling the user to drive a humanoid robot to grasp a can from a desk, and move to a table to drop the can on it [11]. With that controller, precise positioning of the robot was impossible, due on one hand, to the coarse control interface provided by the BCI, and on the other, to the poor perception of the environment that the user had through the robot camera. Besides, the robot could only operate on rigid objects, and did not take into account the presence of the human. In [12], we introduced a novel navigation assistance scheme that addressed these issues.

In this work, we present a more general framework, for BCI-controlled task execution and human-robot interaction. The main novelties are the generality of the framework, that can now be used for both navigation and interaction, and the improvement in the positioning accuracy with respect to our previous work. A versatile vision system has been developed, to generalize object recognition and pose detection, and include human detection and segmentation. The addition of accurate navigation, based on SLAM and on a pose regulation controller, offers the user the choice of a semi-autonomous control mode that guarantees precise positioning. Finally, the immersivity of the system has been greatly enhanced thanks to a head mounted display (HMD) where the user can see through the robot's "eyes" and receive information given by the vision system. With these fundamental contributions, it is possible to perform complex tasks such as allowing the user to control the robot to perform interacton with himself/herself.

The paper is organised as follows. In Sect. II, the components of the framework and their mutual interaction are presented. In Sect. III and IV respectively, the vision and control methods used in this work are detailed. The experiments are presented in Sect. V, and we conclude in Sect. VI.

II. Framework Presentation

In this section, we outline our framework with all the reference frames that are used.

A. Framework

Our framework is outlined in Fig. 1. In our work, we use the human-size humanoid robot HRP-2, remotely controlled

[1] CNRS-AIST Joint Robotics Laboratory (JRL), UMI3218/CRT, Tsukuba, Japan.

[2] CNRS-UM2 LIRMM, Interactive Digital Human group, UMR5506, Montpellier, France.

[a] http://www.vereproject.eu/

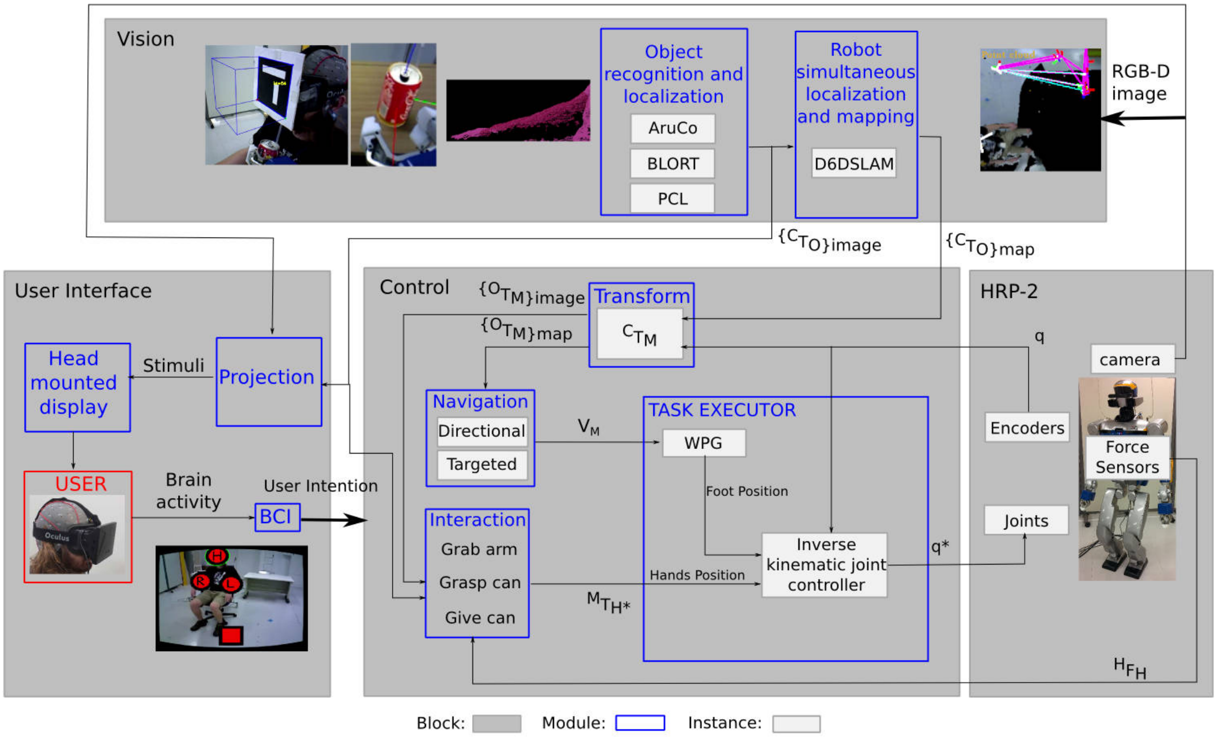

Fig. 1: Data flow of the complete framework.

by a user equipped with a BCI cap and with an HMD. HRP-2 is a 32 degrees of freedom platform, equipped with four 6-axis force/torque sensors (one in each wrist and ankle), and with an RGB-D camera (Asus Xtion) in the head. The RGB-D images acquired by the robot are processed by the `Vision` modules to localize the robot in the environment, and to recognize and localize relevant objects (including human body parts) in the scene[b]. Since the Xtion and HMD intrinsic parameters are known, it is straightforward to project both the robot camera view and the recognized object models in the HMD. The objects are displayed in the form of stimuli, flickering at different frequencies, and related to tasks to be realized by the robot. The `User Interface` display changes according to the current state, which can be selected by the user, or automatically triggered, e.g., by the detection of certain objects, as will explained in the experiments. For instance, recognized objects or directional arrows will flicker, respectively, in object manipulation, and directional navigation states. When the user focuses his/her attention on a flickering stimulus, a peak of energy is observed in the EEG power spectrum, at the corresponding frequency. This peak is detected with an electrode cap (BCI) on the user head, through steady state visually evoked potential (SSVEP). The user intention output by the BCI is sent to the `Control` modules, to realize the corresponding tasks (`Navigation` or `Interaction`) with the robot. Details on the `Vision` and `Control` modules, which represent the major contributions of this work, will be given respectively in Sections III and IV.

B. Reference Frame Definitions

The reference frames used in this work are shown in Figure 2. On the robot we consider: the RGB-D camera frame `C`, the robot center of mass (CoM) frame `M` (with X and Y parallel to the ground), and the robot operating hand (left or right) frame `H`. In the environment, `O` represents the frame linked to the target object. In the figure, we show three possible instances for this frame: an aluminium can `B`, the user forehead `F`, or his/her arm `A`. Throughout this work, points are represented using the homogeneous representation. Coordinate frames are defined in superscript, such as ${}^{A}\mathbf{P}$, and the homogeneous transformation matrix ${}^{B}\mathbf{T}_{A} \in \mathbb{SE}(3)$ transforms points from frame `A` to `B`. The transformation ${}^{B}\mathbf{T}_{A}$ is characterized by translation ${}^{B}\mathbf{P}_{A} = ({}^{B}X_{A}, {}^{B}Y_{A}, {}^{B}Z_{A})$ and rotation matrix ${}^{B}\mathbf{R}_{A}$.

III. Vision

Several image processing modules have been integrated in our framework to realize the different computer vision tasks required for semi-autonomous control. Here, vision is used to recognize and localize objects, and to localize the robot in the environment. All the image processing modules are detailed hereby.

A. Object recognition and localization

The `Object recognition and localization` module gathers the different algorithms for identifying, tracking and localizing objects seen by the RGB-D camera. This module outputs the object identity

$$O = \{B, F, A, \dots\}, \tag{1}$$

[b] Throughout this paper, human body parts are categorized as *objects*

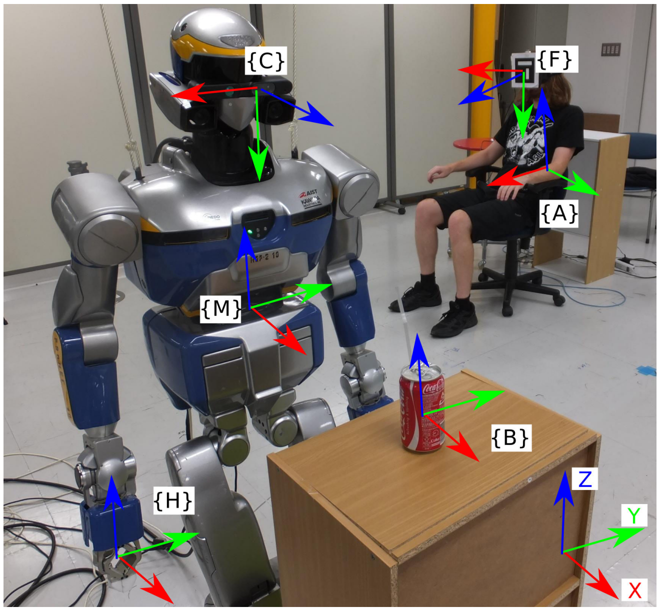

Fig. 2: Reference frames used in this work

and its pose in the camera frame, ${}^{C}\mathbf{T}_{O}$.

Depending on the object category, three object recognition modules have been integrated: an augmented reality tracker for markable objects, an RGB-D tracker for 3D and texture modelled bodies, and a cloud-based tracker for human body parts. In the following, we detail these three algorithms.

1) Marked objects: Augmented reality (AR) markers are used to tag certain objects in the environment. To identify and localize the markers in RGB images, we use the ArUco library[c]. ArUco relies on printed black and white square fiducial markers, instead of natural textures or key points (as in [13]), to provide a very fast, robust and accurate pose estimate. The algorithm consists in detecting the square contour, finding the optimal threshold in the bimodal image, extracting the binary code, and comparing it to the known dictionary, to infer the marker identity O. Finally, the marker pose ${}^{C}\mathbf{T}_{O}$ is estimated by iteratively minimising the image plane projection error of the four square corners, with the Levenberg-Marquardt method. Within our framework, ArUco is indispensable for all tasks that require precise localization of objects that can be tagged, without excessive environment structuring. For example, we use it to derive the human user head position, thanks to an AR marker placed on the HMD (see Fig. 1).

2) 3D and texture modelled objects: For all objects that cannot be tagged, and that must be recognized by relying only on their natural aspect, we use the Blocks World Robotic Vision Toolbox, BLORT [14]. This provides tools to recognize and track objects using RGB images. The recognition is done using a priori learned Scale-Invariant Feature Transforms along with a 3D CAD model of the object. At run time, the object is tracked and localized using a bootstrap filter to compare the current images with the learned model. In our framework, we use BLORT to recognize and localize everyday life objects (e.g. a tool or a bottle), that must be manipulated by the robot.

3) Human body parts: Another fundamental requirement of our framework is physical interaction between user and robot. To this end, a body part tracker is necessary. We cannot rely on the above approaches, since neither AR markers nor special textures (i.e., clothes) can be applied on parts of the user body. Moreover, skeleton trackers such as NITE[d], which require the user to be standing fully in the camera field of view, cannot be used in our scenario, where these assumptions may be broken, and images are shaky, due to the camera motion.

Thus, we decided to use the PCL library [15] to recognize body parts (e.g., the forearm, as in Fig. 1) based on a pre-recorded template. First, we compute the normal to the template point cloud with the Fast Library for Approximate Nearest Neighbours[e], and then the surface curvature variations with a Fast Point Feature Histogram (FPFH) [16]. Then, the body part pose is estimated by the Sample Consensus Initial Alignment (SAC-IA) algorithm (also from [16]): subsets of points are randomly selected from the currently seen point cloud and from the pre-recorded template, and the closest match between the two point sets is computed via FPFH. The transformation between the two point clouds is iteratively refined until convergence, and then used as initial guess for an iterative closest point algorithm, that is more accurate for estimating ${}^{C}\mathbf{T}_{O}$, since there is no point selection. SAC-IA is useful for the first step, since it is robust to large transformations and does not need an initial guess.

B. Robot Simultaneous Localization and Mapping

To develop a semi-autonomous control strategy, that lightens the burden of frequent user control, precise robot localization is required. Although the navigation we seek is target-oriented (it should drive the robot to a pose relative to the desired object), we do not want to force the robot to observe the target at all times. Removing this constraint will benefit the control, since then the head and/or torso need not be permanently servoed towards the target, and even the walk can locally deviate, e.g., for obstacle avoidance. Therefore, the approaches mentioned in Sect. III-A cannot be applied, since they all require the target to stay in the camera field of view throughout operation.

To this end, we apply the Visual Simultaneous Localization and Mapping algorithm D6DSLAM [17] to the RGB-D images from the on-board camera. This library unifies voxel-based and key-frame representations, to provide real-time mapping of the environment, along with the camera pose in the map. As soon as the target object is recognized and localized with any of the approaches in Sect. III-A, its pose in the D6DSLAM map is memorized, so that ${}^{C}\mathbf{T}_{O}$ can be estimated, via the camera pose, even when the object is not visible. Currently, only this information, in the form ${}^{C}\mathbf{T}_{O}$, is input to the navigation module, although in the future,

[c] www.uco.es/investiga/grupos/ava/node/26

[d] https://wiki.debian.org/PrimeSenseNite

[e] www.cs.ubc.ca/research/flann/

we plan to exploit the map, as well. The use of ${}^{C}\mathbf{T}_{O}$ for Navigation is explained in the next section.

IV. Control

The Control module receives the user intention and the target object pose from the other modules, and generates the motor commands necessary to move the robot in order to realize the desired tasks. The module covers three aspects, Navigation, Interaction and Task execution, that are detailed below.

A. Navigation

The goal of the Navigation module is to drive the robot in the environment according to the user intention from the User Interface, and to the object pose from Vision. The output is the robot CoM desired velocity, V_M. As is commonly done in the literature [18], we consider the humanoid robot as a nonholonomic system, so that only its forward linear velocity along X_M, v, and angular velocity around Z_M, ω, are controlled within known bounds:

$$V_M = [v, \omega]^{\top} \in [0, v_m] \times [-\omega_m, \omega_m] \subset \mathbb{R}^2. \tag{2}$$

In our framework, two alternative navigation modes have been devised: a directional mode, where the user maintains full control of the robot motion, and a targeted mode, where control is shared with the robot. This second mode is proposed to the user only if a target object has been detected in the scene by Vision. To seamlessly switch between the two modes, a new user command is taken into account only when a walking step (of period $dt = 0.7s$) is completed. To ensure the stability of the walk v_m and ω_m are fixed respectively at $0.1\ m/s$ and $0.15\ rad/s$.

1) Directional Navigation: In this mode, the user can drive the robot forward, and turn left, right or stop, without any assistance. This mode is particularly useful when the user wants to explore his/her surrounding with the robot. Four fixed directional commands can be selected in the User Interface, via the BCI. These are:

$$\left\{ [v^*, 0]^{\top}, [0, \omega^*]^{\top}, [0, -\omega^*]^{\top}, [0, 0]^{\top} \right\}, \tag{3}$$

with v^* and ω^* fixed hand-tuned velocities.

2) Targeted navigation: In this mode, the user indicates a target, detected with the vision system, that he/she wants the robot to interact with. The robot then autonomously drives to the target. The input to this mode is the target object, selected by the user with the BCI, and the object pose in the camera frame, ${}^{C}\mathbf{T}_{O}$, from any of the Vision algorithms. This pose is transformed to the CoM frame, by applying the known (via the joint values $\mathbf{q}$, measured by the robot encoders) transformation ${}^{C}\mathbf{T}_{M}$:

$${}^{O}\mathbf{T}_{M} = \left({}^{C}\mathbf{T}_{O}\right)^{-1} {}^{C}\mathbf{T}_{M}. \tag{4}$$

For each object, we define a desired robot pose for interaction, noted M*, and characterized by ${}^{O}\mathbf{T}^{*}_{M}$. The goal of Targeted navigation is to drive the robot there: ${}^{O}\mathbf{T}_{M} = {}^{O}\mathbf{T}^{*}_{M}$. For instance, to interact with the human left arm, we choose the pose M* shown in Fig.3, with robot facing the human left side, at a distance of 1 m. We also impose the robot to always stay at least 1 m away from the user, i.e., outside circle Γ of radius $R = 1$ m, to make him/her comfortable. The choice $R = 1$ is driven by the human-robotics proxemics research in [19].

In our previous work [12], this mode relied on several waypoints, causing a piecewise linear trajectory, uncomfortable for the user. To solve this problem, a new navigation scheme has been devised, allowing a trajectory closer to what the user expected. This is one of the contributions of this paper.

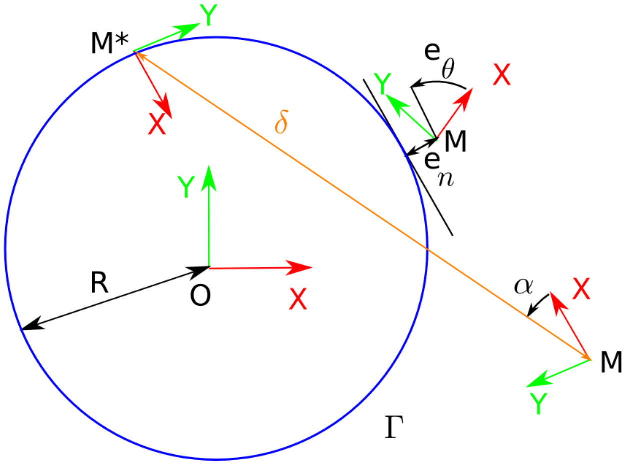

Fig. 3: Relevant variables used for targeted navigation.

The targeted navigation controller is composed of three phases, listed below (all variables are shown in Fig. 3).

1) The robot pivots to face the direction towards M*, i.e., to nullify error $\alpha = \text{atan2}\left({}^{M}Y_{M^*}, {}^{M}X_{M^*}\right)$. We apply:

$$v_1 = 0, \qquad \omega_1 = \begin{cases} \text{sign}(\alpha)\omega_m & \text{if } |\alpha| > \omega_m dt, \\ \frac{\alpha}{dt} & \text{otherwise.} \end{cases} \tag{5}$$

2) The robot walks forward towards M*, to nullify position error $\delta = ||{}^{M}\mathbf{P}_{M^*}||$, while still servoing α:

$$v_2 = \begin{cases} v_m & \text{if } \delta > v_m dt, \\ \frac{\delta}{dt} & \text{otherwise,} \end{cases} \qquad \omega_2 = \omega_1. \tag{6}$$

If the targeted object is a body part, the trajectory obtained with controller (6) must stay outside circle Γ. If Γ is intersected before reaching M*, the robot should follows it towards M* with ω, while maintaining the same linear velocity v_2. Naming e_θ and e_n the signed errors in orientation and normal with respect to Γ (shown in Fig. 3), and $\kappa = \frac{\text{sign}({}^{M^*}Y_M)}{R}$ the signed desired path curvature, we apply the classical noholonomic circle following controller from [20]:

$$w_2 = \left(\frac{\lambda_1 e_n \sin e_\theta}{e_\theta} - \lambda_2 e_\theta + \frac{\kappa \cos e_\theta}{1 + \kappa e_n} \right) v_2. \tag{7}$$

This controller, with properly tuned positive gains $\lambda_{1,2}$, guarantees local stability to the circular path. In all experiments, we used $\lambda_1 = 10$ and $\lambda_2 = 5$.

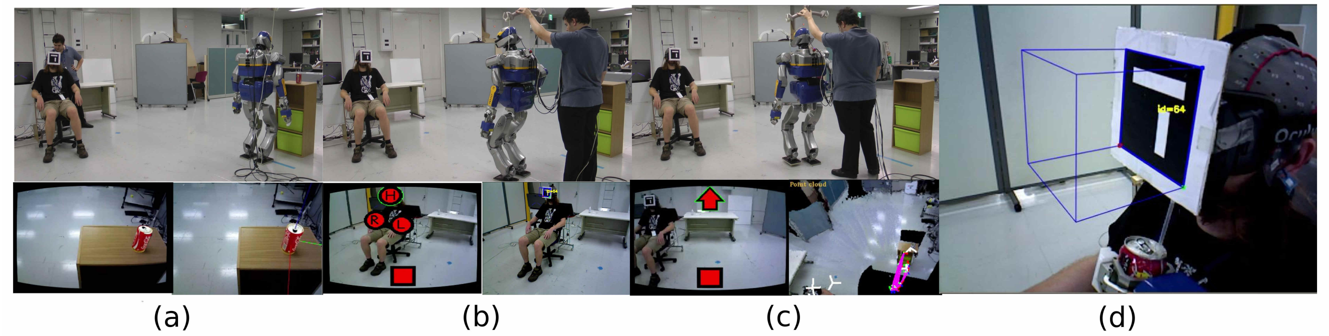

Fig. 4: Consecutive phases of the first experiment: The user controls the robot to serve himself a drink.

3) Once the robot has reached the position $\mathtt{M}^*$, it pivots to align axes X_M and X_M^*, i.e., to nullify their relative angle β. We apply:

$$v_3 = 0, \qquad \omega_3 = \begin{cases} \mathrm{sign}(\beta)\omega_m & \text{if } |\beta| > \omega_m dt, \\ \frac{\beta}{dt} & \text{otherwise.} \end{cases} \tag{8}$$

The transition between phases are triggered when the error norms are below given positive thresholds (e.g., $|\alpha| < \tau_\alpha$ to go from phase 1 to phase 2).

B. Interaction

Various actions can be realized by the robot for Interaction with the environment, and/or with the user. We distinguish between two categories: actions for touching/grasping objects in the environment (e.g., touching a body part or grasping a can), and actions for handing over objects to the user (e.g., giving a tool). The chosen action depends on the user intention output of the User Interface. The object pose ${}^C\mathbf{T}_O$, output by Vision, is also necessary to provide feedback to the controller. For Interaction, we use the output of Object Localization, since it is more accurate than the one from Robot Localization. This requires the object to be in the field of view of the camera. However, this is not an issue since, when interacting with an object, the robot is close enough to it, to have it in the camera field of view. All Interaction actions in our framework are realized by setting desired poses of the robot hands in the CoM frame, ${}^M\mathbf{T}_H^*$, and these are realized with the Inverse kinematic joint controller that will be explained just below.

The control objective, for both Object touching and grasping, consists in regulating the robot hand on the object, so that the two reference frames coincide, i.e., ${}^O\mathbf{T}_H^* = \mathbf{I}$. This is equivalent to servoing the hand so that:

$${}^M\mathbf{T}_H^* = {}^M\mathbf{T}_C\,{}^C\mathbf{T}_O\,{}^O\mathbf{T}_H^* = {}^M\mathbf{T}_C\,{}^C\mathbf{T}_O, \tag{9}$$

with ${}^M\mathbf{T}_C$ derived from the encoders, as mentioned in Sect. IV-A.2. For implementation reasons, in some cases (shown in Sect. V), we use N intermediate way points ${}^M\mathbf{T}_{Hi}^*$, $i = 1, \ldots, N$, to reach ${}^M\mathbf{T}_H^*$. In the case of grasping, the convergence to ${}^M\mathbf{T}_H^*$ is followed by the hand gripper closure. Also, in some cases, the force/torque sensor data (external wrench on the hand, expressed in $\mathtt{H}$, ${}^H F_H$) is compared to a threshold τ_F to interrupt the grasping/touching motion:

$$||{}^H F_H|| > \tau_F. \tag{10}$$

Typically, when interacting with body parts, this is necessary to realize a pleasant touch [21]. In future work, we plan to device force feedback controllers to realize more human-like actions.

C. Task execution

A task executor, is used to realize the operations requested by the Navigation and Interaction modules. This includes both making the robot walk, and controlling the motion of its hands. To make the humanoid walk, we use the walking pattern generator from [22], which receives the desired speed V_M, and computes the corresponding desired foot positions. These foot positions, just like the desired hand positions ${}^M\mathbf{T}_H^*$ from (9), correspond to tasks to be realized by our inverse kinematic joint controller, the stack of tasks (SoT, [23]). In the SoT, the tasks are defined as state error vectors in the sensory space, and projected in the robot joint space via the robot kinematic jacobian. The SoT ouputs the desired joint values, $\mathbf{q}^*$. This tasks function control paradigm is well adapted here, since the explicit trajectories of the interaction motions are unknown beforehand.

V. EXPERIMENTS

In this section, we present the experiments devised to assess our semi-autonomous control framework in assistive robotics scenarios. In both experiments, we rely on SSVEP to extract the user intention from the EEG signal (measured by eight electrodes), with up to four stimuli at frequencies of 6 Hz, 8 Hz, 9 Hz, and 10 Hz. To validate our framework, two challenging tasks are tackled. In a first experiment, the user controls the robot to take a drink from a desk, carry it through the room, and hand it to his mouth. In the second experiment, the user steers the humanoid near himself, in order to raise his arm. Both experiments are shown in the video attached to this paper.

In the first experiment, shown in Fig. 4, the user controls the robot to take a can from a desk, and carry it to his mouth, so that he can take a sip. In this experiment, all the computer vision algorithm are used. First, we use BLORT to recognize the can B, and estimate its pose ${}^{C}\mathbf{T}_{B}$. Once the can is recognized and localized, a stimulus shaped as the can model is projected on the HMD perceived by the user (see Fig. 4.a). When the user focuses on this stimulus, and his intention is detected by the BCI, the robot grasps the can. Then, the user controls the robot with the directional navigation mode described in IV-A.1, to 'look for himself' in the environment, while the AruCo algorithm is running to detect the marker located on the HMD he wears. To keep the walk safe and reactive, we have set $v^* = 0.1\ m/s$, and $\omega^* = 0.15\ rad/s$. As soon as the marker is localized, the display changes, to propose the user the various body parts he can interact with. In this experiment, the user selects his head F, to make the robot serve him the drink (see Fig. 4.b). Then, targeted navigation is activated, to assist the user in driving the robot near himself. This relies on the pose ${}^{F}\mathbf{T}_{M}$, given by D6DSLAM (see Fig. 4.c). The localization provided by D6DSLAM proved accurate and robust enough, in spite of the camera sway during robot walk. This is shown by the CoM trajectory, plotted in Fig. 5, that leads to the desired pose, while avoiding the safety circle Γ. At the end of this phase, the robot is well placed near the user, and the interaction phase is triggered. Precise can positioning requires an accurate estimate of the user head. To this end, the ArUco algorithm is used to determine, once again via the marker on the HMD, the pose of the forehead, ${}^{C}\mathbf{T}_{F}$. Then, the hand control starts, to drive the can close to the forehead, while accounting for the mouth and straw length offsets. Finally, the user can drink from the can (see Fig. 4.d).

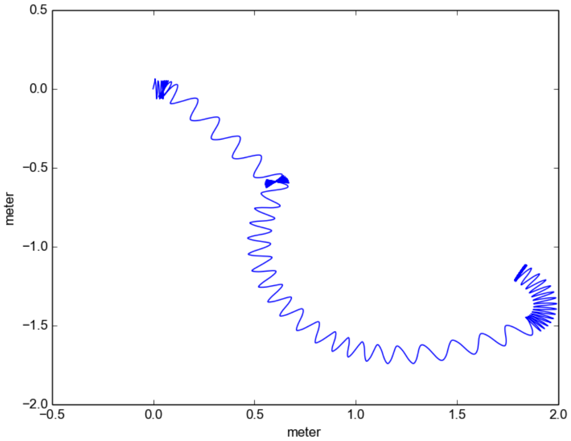

Fig. 5: Trajectory of the robot CoM in the user forehead frame F, obtained from the encoder measures.

In the second experiment, shown in Figure 6, the user controls the robot toward himself and uses it to raise his forearm. The robot is driven towards the forearm using, again, targeted navigation. The accurate forearm pose ${}^{C}\mathbf{T}_{A}$ is given by the algorithm presented in III-A.3. In this case, the robot hand motion passes through three waypoints. The first waypoints is used to raise and orient the hand of the gripper without approaching the forearm. To reach the second waypoint, the robot hand translates along the X_M axis and stops when it encounters the user forearm (i.e., when the external wrench is above a threshold, as indicated in (10)). Once touched, the hand gripper is closed and the robot hand is translated upward along the Z_M axis, towards the third waypoint. After 5 seconds, the robot hand goes downward along the Z_M to the second waypoint, and the robot hand finally returns to the first waypoint.

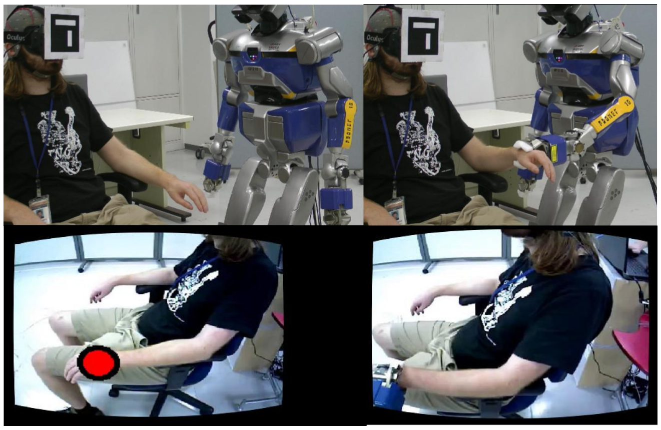

Fig. 6: The user arm is grabbed and raised by himself using the humanoid robot surrogate.

VI. Conclusion

In this paper, we have presented an integrated framework, enabling a user to remotely control a humanoid robot with a BCI, to assist him/her in domestic tasks. The feeling of embodiment is given to the user via a HMD. To bypass the BCI bandwidth limitations, we have made the framework semi-autonomous, by integrating various visual object recognition and localization algorithms, and a targeted navigation mode. To our knowledge, this is the first time that navigation and interaction (both with the environment objects and with the user) have been concurrently integrated in a BCI-based framework. Future research will focus on assessing the framework from the disabled users viewpoint, and exploiting the environment map to improve navigation.

Acknowledgment

This research is supported by the European Union FP7 Integrated Project VERE (No. 257695)

References

[1] M. Merten and A. Bley, "A mobile robot platform for socially assistive home-care applications," in *7th German Conference on Robotics ROBOTIK 2012*, pp. 233–238.

[2] E. Cervera, A. A. Moughlbay, and P. Martinet, "Localization and Navigation of an Assistive Humanoid Robot in a Smart Environment," in *IROS 2012 International Workshop on Assistance and Service Robotics in a Human Environment*, pp. 2–6.

[3] J. Vidal, "Toward Direct Brain-Computer Communication," *Annual Review of Biophysics and Bioengineering*, vol. 2, pp. 157–180, 1973.

[4] C. J. Bell, P. Shenoy, R. Chalodhorn, and R. P. N. Rao, "Control of a humanoid robot by a noninvasive brain-computer interface in humans.," *Journal of Neural Engineering*, vol. 5, pp. 214–20, June 2008.

[5] R. A. R. C. Gopura, D. S. V. Bandara, J. M. P. Gunasekara, and T. S. S. Jayawardane, *Electrodiagnosis in New Frontiers of Clinical Research*. InTech, May 2013.

[6] A. S. Royer and B. He, "Goal selection versus process control in a brain-computer interface based on sensorimotor rhythms.," *Journal of Neural Engineering*, vol. 6, p. 016005, Mar. 2009.

[7] J. D. R. Millán, R. Rupp, G. R. Müller-Putz, R. Murray-Smith, C. Giugliemma, M. Tangermann, C. Vidaurre, F. Cincotti, A. Kübler, R. Leeb, C. Neuper, K.-R. Müller, and D. Mattia, "Combining Brain-Computer Interfaces and Assistive Technologies: State-of-the-Art and Challenges.," *Frontiers in Neuroscience*, vol. 4, pp. 1–33, Jan. 2010.

[8] J. Weisz, C. Elvezio, and P. Allen, "A User Interface for Assistive Grasping," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems IROS*, Nov. 2013.

[9] F. Galán, M. Nuttin, E. Lew, P. W. Ferrez, G. Vanacker, J. Philips, and J. del R. Millán, "A Brain-Actuated Wheelchair: Asynchronous and Non-Invasive Brain-Computer Interfaces for Continuous Control of Robots," *Clinical Neurophysiology*, vol. 119, no. 2007, pp. 2159–2169, 2008.

[10] A. Akce, M. Johnson, O. Dantsker, and T. Bretl, "A brain-machine interface to navigate a mobile robot in a planar workspace: enabling humans to fly simulated aircraft with EEG.," *IEEE Transactions on Neural Systems and Rehabilitation Engineering : a publication of the IEEE Engineering in Medicine and Biology Society*, vol. 21, pp. 306–18, Mar. 2013.

[11] P. Gergondet, A. Kheddar, C. Hintermüller, C. Guger, and M. Slater, "Multitask Humanoid Control with a Brain-Computer Interface: user experiment with HRP2-2," *International Symposium on Experimental Robotics ISER*, 2012.

[12] D. Petit, P. Gergondet, A. Cherubini, M. Meilland, A. I. Comport, and A. Kheddar, "Navigation Assistance for a BCI-controlled Humanoid Robot," in *IEEE International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (IEEE-CYBER 2014)*, 2014.

[13] D. Wagner, G. Reitmayr, A. Mulloni, T. Drummond, and D. Schmalstieg, "Real-Time Detection and Tracking for Augmented Reality on Mobile Phones," *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, no. 3, pp. 355–68, 2010.

[14] T. Mörwald, J. Prankl, A. Richtsfeld, M. Zillich, and M. Vincze, "BLORT - The Blocks World Robotic Vision Toolbox," in *Best Practice in 3D Perception and Modeling for Mobile Manipulation (in conjunction with ICRA 2010)*, 2010.

[15] R. B. Rusu and S. Cousins, "3D is here: Point Cloud Library (PCL)," in *IEEE Int. Conf. on Robotics and Automation ICRA*, IEEE, May 2011.

[16] R. B. Rusu, N. Blodow, and M. Beetz, "Fast Point Feature Histograms (FPFH) for 3D Registration," in *IEEE Int. Conf. on Robotics and Automation ICRA*, pp. 3212–3217, IEEE, May 2009.

[17] M. Meilland and A. I. Comport, "On unifying key-frame and voxel-based dense visual SLAM at large scales," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems IROS*, IEEE/RSJ, 2013.

[18] A. Faragasso, G. Oriolo, A. Paolillo, and M. Vendittelli, "Vision-Based Corridor Navigation for Humanoid Robots," *IEEE Int. Conf. on Robotics and Automation ICRA*, pp. 3190–3195, May 2013.

[19] J. Mumm and B. Mutlu, "Human-Robot Proxemics: Physical and Psychological Distancing in Human-Robot Interaction," in *IEEE International Conference on Human-Robot Interaction*, pp. 331–338, 2011.

[20] C. Canudas de Wit, B. Siciliano, and G. Bastin, *Nonlinear feedback control*. Communications and Control Engineering, Springer, 1996.

[21] G. K. Essick, F. McGlone, C. Dancer, D. Fabricant, Y. Ragin, N. Phillips, T. Jones, and S. Guest, "Quantitative assessment of pleasant touch.," *Neuroscience and Biobehavioral Reviews*, vol. 34, pp. 192–203, Mar. 2010.

[22] A. Herdt, H. Diedam, P.-B. Wieber, D. Dimitrov, K. Mombaur, and M. Diehl, "Online Walking Motion Generation with Automatic Footstep Placement," *Advanced Robotics*, vol. 24, no. 5-6, pp. 719–737, 2010.

[23] N. Mansard, O. Stasse, P. Evrard, and A. Kheddar, "A Versatile Generalized Inverted Kinematics Implementation for Collaborative Humanoid Robots: The Stack of Tasks," in *Int. Conf. on Advanced Robotics ICAR*, no. 8, 2009.