Brain-Computer Interface meets ROS: A robotic approach to mentally drive telepresence robots*

Gloria Beraldo[1], Morris Antonello[1], Andrea Cimolato[2], Emanuele Menegatti[1], Luca Tonin[3]

Abstract— **This paper shows and evaluates a novel approach to integrate a non-invasive Brain-Computer Interface (BCI) with the Robot Operating System (ROS) to mentally drive a telepresence robot. Controlling a mobile device by using human brain signals might improve the quality of life of people suffering from severe physical disabilities or elderly people who cannot move anymore. Thus, the BCI user is able to actively interact with relatives and friends located in different rooms thanks to a video streaming connection to the robot. To facilitate the control of the robot via BCI, we explore new ROS-based algorithms for navigation and obstacle avoidance, making the system safer and more reliable. In this regard, the robot can exploit two maps of the environment, one for localization and one for navigation, and both can be used also by the BCI user to watch the position of the robot while it is moving. As demonstrated by the experimental results, the user's cognitive workload is reduced, decreasing the number of commands necessary to complete the task and helping him/her to keep attention for longer periods of time.**

I. INTRODUCTION

Brain-Computer Interface (BCI) technology relies on the real-time detection of specific neural patterns in order to circumvent the brain's normal output channels of peripheral nerves and muscles [1] and thus, to implement a direct mind-control of external devices. In this framework, current non-invasive BCI technology demonstrated the possibility to enable people suffering from severe motor disabilities to successfully control a new generation of neuroprostheses such as telepresence robots, wheelchairs, robotic arms and software applications [2], [3], [4]. Among the different BCI systems developed in the last years, the most promising ones for driving robotic devices are the so-called endogenous BCIs (e.g., based on Sensorimotor Rhythm (SMR)), where the user autonomously decides when starting the mental task without any exogenous—visual or auditory—stimulation.

In these systems, neural signals are recorded by non-invasive techniques (e.g., Electroencephalography (EEG)) and then, task-related brain-activity is translated into few commands—usually, discrete—to make the robotic device turn right or left. Despite the low number of commands provided by non-invasive BCIs, researchers have demonstrated the possibility to drive mobile devices even in complex situation with the help of a shared control approach [4], [5], [6]. The shared control approach [7] is based on a seamless human-robot interaction in order to allow the user to focus his/her attention on the final destination and to ignore low level problems related to the navigation task (i.e., obstacle avoidance). The coupling between user's intention and robot's intelligence allows to contextualize and fuse the high-level commands coming from the BCI with the environment information from the robot's sensors and thus, to provide a reliable and robust semi-autonomous mentally-driven navigation system.

In the robotic community Robot Operating System (ROS) [8] is becoming the standard *de facto* for controlling different types of devices. ROS is a middleware framework that provides a common infrastructure and several, platform independent, packages (i.e., localization, mapping, autonomous navigation). Indeed, the most important advantages of ROS are its strong modularity and the large and growing community behind: people can design and implement their own ROS package with specific functionalities and thus, distribute it through common repositories.

Although the clear benefits of using ROS, it is still far to be a standard adopted in the BCI community. In BCI literature, most of the studies are based on custom and *ad-hoc* implementations of the robotic part and only few of them clearly reported an integration with common available tools in ROS [9], [10], [11]. The drawback of this tendency is twofold: on one hand, the lack of standardization makes almost impossible to check, replicate and validate experimental results. As a matter of fact in BCI experiments the technology needs to be tested over a large population of end-users with severe disabilities and, usually, requires to be validated by different groups before the acceptance as an effective assistive tool [12]. On the other hand, home-made control frameworks for robotic devices imply the adoption of simplified and naive approaches to fundamental robotic challenges—usually, already solved by the robotic community—and thus, a limitation of possible applications of the current BCI driven neuroprostheses.

This paper aims at showing the benefits of integrating a state-of-the-art BCI system and ROS for controlling a telepresence robot. In Section II, we describe the BCI and the robot adopted as well as our novel navigation algorithm to mentally drive telepresence robots. In contrast to previous works, it exploits an optimal trajectory planner and the availability of the environmental map. Furthermore, it is designed to match the requirements of a semi-autonomous,

*This research was partially supported by Fondazione Salus Pueri with a grant from "Progetto Sociale 2016" by Fondazione CARIPARO, by Consorzio Ethics with a grant for the project "Hybrid Human Machine Interface", and by Omitech s.r.l. with the grant "O-robot".

[1] Intelligent Autonomous System Lab, Department of Information Engineering, University of Padova, Padua, Italy. emg@dei.unipd.it

[2] Politecnico di Milano in collaboration with Istituto Italiano di Tecnologia (IIT), Milan, Italy

[3] Defitech Chair in Brain-Machine Interface (CNBI), École Polytechnique Fédérale de Lausanne (EPFL), Chemin des Mines 9, CH-1202, Geneva, Switzerland luca.tonin@epfl.ch

BCI driven telepresence robot. In Section III, we evaluate the presented methods and we showcase the integration with the BCI system. Finally, in Section IV, we discuss the results achieved with respect to similar BCI based experiments.

II. METHODS

A. Brain-Computer Interface system

In this work, we used a 2-class BCI based on SMRs to control the telepresence robot. The user was asked to perform two motor imagery tasks (imagination of the movement of both hands vs. both feet) to make the robot turn left or right. Contrariwise to other approaches (e.g., based on evoked potentials), such a BCI is based on the decoding of the voluntary modulation of brain patterns without the need of any external stimulation repetitively presented to the user. For this reason, SMR BCIs have been widely exploited to successfully drive mobile devices [4], [5], [6], [13], [14].

The following paragraphs briefly describe the different parts of the BCI system developed and used for the study.

1) EEG acquisition: A health 24-year-old female and BCI beginner tried the experiment that was carried out in accordance with the principles of the Declaration of Helsinki.

EEG signals were recorded with an active 16-channel amplifier at 512 Hz sampling rate, filtered within 0.1 and 100 Hz and notch filtered at 50 Hz (g.USBamp, Guger Technologies, Graz, Austria). Electrodes were placed over the sensorimotor cortex (Fz, FC3, FC1, FCz, FC2, FC4, C3, C1, Cz, C2, C4, CP3, CP1, CPz, CP2, CP4) according to the international 10-20 system layout.

2) Feature extraction and classification: EEG was preprocessed by applying a Laplacian spatial filter. The Power Spectral Density (PSD) of the signal was continuously computed via Welch's algorithm (1 second sliding window, 62.5 ms shift) in the frequency range from 4 to 48 Hz (2 Hz resolution). Thus, the most discriminative features (channel-frequency pairs, subject-specific) were extracted and classified online by means of a Gaussian classifier [15] previously trained during the calibration phase (see Section II-A.3). Finally, the raw posterior probabilities were integrated over time to accumulate evidences of the user's intention according to:

$$p(y_t) = \alpha \cdot p(y_{t-1}) + (1-\alpha) \cdot p(y_t|x_t) \qquad (1)$$

where $p(y_t|x_t)$ is the probability distribution at time t, $p(y_{t-1})$ the previous distribution and α the integration parameter. The probabilities were showed to the user as a visual feedback (Fig. 1B). As soon as one of the bar was filled, the command was delivered to the robot to make it turn right or left.

3) Calibration and online phases: As a common practice in BCI experiments, a calibration phase is required in order to select the features that each subject can voluntary modulate during motor imagery tasks and to train the classifier. In this work, the calibration phase consisted in three runs with 30 trials each where the user was instructed by symbolic cues about the task to be performed (in total 21 minutes). Thus, we analyzed the recorded data, we selected the subject-specific features and we trained the Gaussian classifier. Fig. 1A depicts the spatial and spectral distributions of the most discriminative features (based on fisher score values) selected to train the BCI. The distributions are perfectly coherent with the brain patterns expected during motor imaginary tasks [1].

During the online phase, we evaluated the ability of the BCI to correctly detect user's intentions. The user performed three online runs, where he was asked to control the online BCI feedback on the screen (Fig. 1B).

B. Robot

Our telepresence platform is the Pepper robot[1] by Aldebaran Robotics and SoftBank (Fig. 1C). It is an humanoid robot designed for human-robot interaction. It features a 1.9 GHz quad-core Atom processor and 4 GB of RAM. It is 1.210 m high and equipped with an omnidirectional base of size 0.480×0.425 m. For obstacle avoidance, it is provided with two sonars, three bumpers, three laser sensors and three laser actuators. For vision tasks, the robot has two 2D cameras located in the forehead, in particular one at the bottom and one at the top, both with a resolution of 640×480 px. For telepresence purposes, we exploited the top camera to provide a first-person view to the BCI user by means of the RViz graphical interface available in ROS. The Pepper has also an ASUS Xtion 3D sensor in one of its eyes with a resolution of 320×240 px. However, its 3D data are distorted due to a pair of lenses, positioned in front of it. To overcome the limitations of the laser and the RGBD-sensor, we built the environmental maps required for a safe navigation by using data previously acquired [16] and based on a more powerful Hokuyo URG-04LX-UG01 2D laser rangefinder able to measure distances from 20 mm to 5.6 m and the more precise Microsoft Kinect v2. This way, we can still exploit Pepper's sensors for navigation.

C. ROS-based Mapping and Localization

Robot mapping and localization are core functionalities necessary to correctly navigate in both an autonomous or semi-autonomous way. In particular, we built the static environmental maps, which are provided to the Pepper for localization and navigation, with previously acquired data. For the building map process, we exploited two different methods available in ROS: *GMapping* [17], [18], and *OctoMap* [19]. *GMapping* builds a 2D occupancy map of the environment, while *OctoMap* creates a 3D scene representation, which can be down-projected to the ground so as to enrich the 2D occupancy map with higher obstacles visible by the RGBD-sensor but not by the laser. In particular, the localization module is based on the map built with *GMapping*, while the navigation one is based on the 2D down-projected map because of the richer representation of the environment. This way, the trajectory planner can take into account high obstacles and avoid collisions (Fig. 2A). As illustrated in Fig. 2A, despite in both maps the planner trajectories seem

[1] https://www.ald.softbankrobotics.com/en/robots/pepper/find-out-more-about-pepper

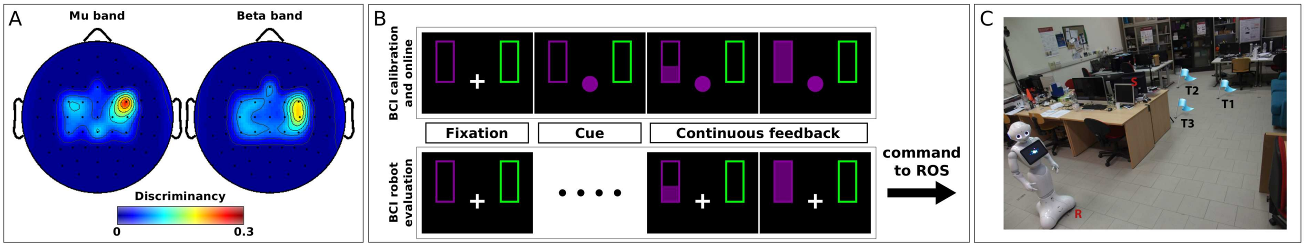

Fig. 1: A) Topographic representation of the discriminant features in μ and β bands used to train the SMR classifier (fisher score values, both hands vs. both feet). B) Schematic representation of the visual paradigm of the SMR BCI. Top row: the protocol exploited during the calibration and online phases. User is instructed to perform the motor imagery task according to a symbolic cue appearing on the screen. Thus, the BCI classification output is remapped into the movement of the bar. When a bar is completely filled, the trial ends. Bottom row: same behavior as before but there is no cue and the user decides autonomously which motor imagery task to perform to control the robot. When a bar is completely filled, the related command is delivered to the ROS infrastructure. C) The telepresence robot platform (Pepper) and the experimental environment with the three target locations.

Algorithm 1 The shared control navigation algorithm

```
$last\_time$ ← current time
while ( IsRobotActive()) do
    if current time − $last\_time$ > $CLEAR\_TIME$ then
        ClearCostMap()
        $last\_time$ ← current time
    end if
    if (DefaultBehaviour()) then
        Robot goes forward by a fixed step $S_x$
        if (!Succeeded) then
            Call RecoveryBehaviour()
        end if
    else                                        ▷ BCI command arrival
        Cancel the current goal
        Robot turns to the right direction
        if (!Succeeded) then
            Call RecoveryBehaviour()
        end if
    end if
end while
procedure RecoveryBehaviour()
    The robot goes back for a fixed time $T$
    It turns counter-clockwise by a fixed angle $A$
end procedure
```

similar, the path found in the *Gmapping* based map is less reliable with high obstacles than the one from *OctoMap*. For instance, in a map built with *GMapping* only legs of tables are considered, while in the 2D down-projected map from *OctoMap* they are featured by their flat surfaces.

For localization, the *Adaptive Monte Carlo Localization (AMCL)* [20] was adopted, with an adaptive sampling scheme, to make the computation efficient. The *Humanoid Robot Localization (HRL)* technique [21], which is based on the AMCL but uses the 3D *OctoMap*, is also evaluated.

D. ROS-based navigation

Our algorithm allows a semi-autonomous navigation based on a shared control for BCIs. The main target is twofold: to help the user to successfully drive the robot and, at the same time, to make him/her feel to have the full control. Indeed, since the control through an uncertain channel like BCI can be complicated, our integration between user and robot is designed so that it allows the former to only focus on the final destinations; while the latter will deal with obstacle detection and avoidance, deciding the best trajectory. For these purposes, we exploited the ROS `navigation stack`[2] to localize and move the robot in the environment according to its sensors, odometry and the static map provided.

In details, the default behaviour of the robot consists in moving forward and avoiding obstacles when necessary. The user can control it by his/her brain activities, delivering voluntary commands (left and right) to turn it to the corresponding direction. The user's intention is decoded by the BCI system and the related command is sent to the ROS node dealing with navigation through an UDP packet.

The logic of our algorithm is described in the following pseudo-code. At every iteration, our algorithm sends new navigation goals to the robot to ensure the robot capability of avoiding obstacles in the environment—especially the dynamic obstacles not represented in the static map. This way, the planner in the `navigation stack`, can (re)compute the best trajectory to reach the target destination even if dynamic obstacles are presented in the path. In details, we used the *Dynamic Window Approach* [22] for local planner and *Dijkstra's algorithm* to compute the global planner function. Furthermore, before sending a new goal to the robot, the corresponding position in the map is checked: the goal is sent to the robot only if it matches with a free cell in the map, which means that that cell is not occupied by an obstacle. Otherwise, the RecoveryBehaviour()

[2] http://wiki.ros.org/navigation

procedure is called to avoid deadlocks by slightly moving the robot. In detail, the RECOVERYBEHAVIOUR() makes the robot go back (if it is possible) and keep turning counter-clockwise until required. The recovery rotation takes place always in the same direction (by fixed angle A) to make the robot able to rotate around itself and, thus, to escape from this undesirable situation. Furthermore, the rotation is carried out incrementally by sending velocity commands to the robot. If the robot cannot go back and/or turn due to obstacles, the on-board short-range sonars will stop it.

Even if the target goal corresponds to a free cell in the map, the robot may not be able to reach it for different technical reasons (e.g., lost connection, temporary missing frame transformations or unseen obstacles). In this kind of situations, the procedure RECOVERYBEHAVIOUR() is called to unstuck the robot and to ensure a continuous navigation.

Finally, the planner may not still find a valid path due to dynamic obstacles previously stored in the cost maps but not currently present in the environment. To avoid such a situation, a clearing operation of the cost maps is done every $CLEAR\_TIME$.

E. Experimental design

The experiment was carried out in a typical working space with different obstacles like tables, chairs, sofa, cupboards, people (Fig. 1C). The user was seated at position S and the robot started from position R. We defined three target positions $T1$, $T2$, $T3$. The user was instructed to move the robot from S, going through $T1$, $T2$, $T3$, by only sending mental commands through the BCI. The default behaviour of the robot was to move forward and to avoid possible obstacles in its path. The user perform two repetitions of the task.

III. RESULTS

A. Navigation Performances

In this work, we considered different combinations of the two kind of maps provided as input to the robot and the method used for localization. More precisely, we examined the performance in terms of number of delocalizations and collisions against obstacles in the experimental environment by simulating 150 random commands delivered by the BCI system. Table I depicts the results achieved.

The combination between the *2D down-projected map* from *Octomap* and *GMapping* together with the AMCL localization method represents a good compromise (3 number of delocalizations and 2 collisions) between providing a more detailed map to the robot and using a 2D and fast computation localization algorithm (Table I, second row). Indeed, although the third approach (with HRL localization method) allowed the lowest number of possible collisions, it requires higher computational power. This is due not only to the 3D *OctoMap* used for navigation, but also to the fact that HRL does not exploit an adaptive sampling scheme adjusting the number of samples.

Navigation Map	Localization Map	Localization Method	Number of delocalizations	Number of collisions
GMapping	*GMapping*	AMCL	6	5
2D down-projected map from OctoMap	*GMapping*	AMCL	3	2
2D down-projected map from OctoMap	3D *Octomap*	HRL	10	1

TABLE I: Evaluation of different combinations of the two kind of maps provided as input to the robot and the method used for localization. Data were acquired by simulating 150 random commands delivered by the BCI system for each approach.

B. BCI driven telepresence

For the integration of BCI and ROS, we analyzed the number of BCI commands delivered by the user to reach the targets $T1$, $T2$, $T3$ and the corresponding times (Fig. 2B). In average, the user delivered 3.0±1.3 commands and employed 34.5±32.2 s (median and standard error) to reach each target. The number of commands and the time required were low for all targets in each repetitions (except for $T1$, second repetition, where the user sent few wrong BCI commands to the robot).

Furthermore, we evaluates the importance of shared control by comparing the BCI with a manual control. In this case, we asked the user to repeat the experiment controlling the Pepper robot with discrete commands sent by the keyboard but without the assistance of the shared control for obstacle avoidance. The ratio between the number of commands in the two modalities (BCI with shared control and the manual without shared control) was 80.9% and the ratio between times was 114.5%.

It is possible to notice that the number of commands increased in the manual modality. This means that without shared control, the BCI user has to send more commands to the robot, increasing the necessary cognitive workload. Especially, in that case in which the robot is blocked because in its neighborhood there are some obstacles that make it stuck. However, the time spent is less using the keyboard, due to the time required by the BCI system before delivering a commands to the robot.

IV. DISCUSSION

The main objective of this study was to demonstrate for the first time the potentialities and the perspectives related to the integration between ROS and a BCI system. Modularity of ROS allows robotic community to exchange and distribute repositories besides the platform adopted [8]. This particular aspects is what makes ROS very appealing in assistive robotics. ROS is able to provide a common infrastructure where developers can either decide to share their novel approaches or adopt external tools through use of common repositories. In this context, this work aimed at promoting collaboration of multiple disciplines in order to design a semi-autonomous EEG driven navigation for telepresence robots. Moreover, integration of the BCI with ROS allowed testing the system on Pepper robotic platform, never experienced in BCI driven teleoperation. Second purpose of this

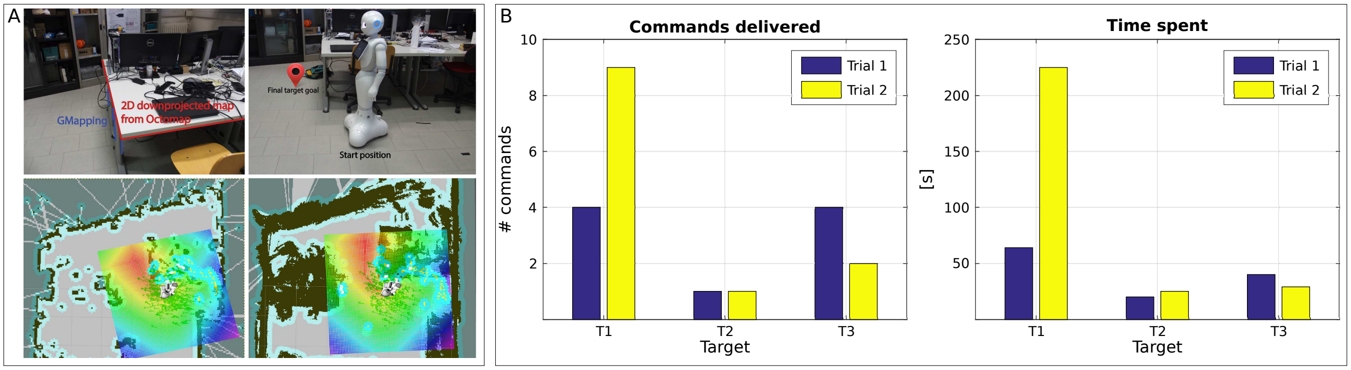

Fig. 2: A) Top row: Comparison detection of a desk performed with *GMapping* and *2D down-projected map* from *OctoMap* (left). Start position and target position for evaluation of robot trajectory (right). Bottom row: Best trajectory proposed by the planner when *GMapping* (left) or *2D down-projected map* from *Octomap* (right) is provided to the robot. B) Number of commands sent and time spent by the user in the two attempts carried out to reach the targets $T1$, $T2$, $T3$.

work was the development of a novel approach for assistive robotic navigation based on multiple-maps input under BCI shared control. Our approach demonstrated the possibility to make obstacle avoidance more reliable and, therefore, the navigation safer.

Results comparison between other BCI studies may be complicated and not always meaningful, due to different testing conditions. However, it is worth to notice that with respect to previous works, results are consistent in terms of ratio between BCI and manual input both for time intervals and number of commands. Our work reported a 114.5% ratio for time intervals between the two modalities, in line to [4] where it was estimated 109±11% for the end-users and 115±10% for healthy ones. Similar trend results are reported in [6] where both type of users in average achieved 118.5±19% ratio between the two modalities. Furthermore, the 19.1% decreasing in number of commands recorded in our experiment was in perfect agreement with [4] and [6] where similar reduction was reported. These preliminary results suggest the possibilities and the advantages of using ROS in BCI driven telepresence applications.

The proposed BCI system is one of the few working on the top of a ROS framework [9], [10], [11] and, among them, the only one supporting an endogenous SMR based BCI. As in the case of previous works [4], the designed semi-autonomous control reduced the user's fatigue (in terms of number of commands required to reach the target). Furthermore, ROS was fundamental in our approach to enable communication among different software and hardware modules and it was essential to overcome BCI limitations by exploiting well-established robotic solutions for obstacle avoidance and navigation.

Modern BCI teleoperating systems are not mature enough to be exploited in the daily life despite the promising results. This divergence is due mainly to different complexities between testing conditions and home-like environments. High density of obstacles and non-uniform space distribution make impracticable to use mentally driven systems in such situations. In fact, platform control could result stressful and exhausting for the end-user, even with obstacle avoidance assistance. In order to provide relief to the user in such conditions, our proposal was to include a localization algorithm in navigation. Direct interaction with this module output conveyed better understanding of robotic platform state and allowed the user to plan in advance the navigation, dealing with delays in command delivery. Maps localization, moreover, was designed to admit path planning strategies in the obstacle avoidance algorithm, promoting an evolution of the shared control approach. Previous implementations of the shared control were able to detect the obstacle and modify the trajectory in order to avoid collisions [4], [5], [6]. However, since algorithm was nor provided with an intermediate goal nor with favorite direction, once the hurdle was evaded, it was user's burden to put the teleoperated device back on track. Contrariwise, our novel implementation permits to identify an obstacle and plan accordingly a new trajectory in order to avoid collision but, at the same time, not deviating from the direction imposed from the user. Fundamental for navigation in hostile areas, involving for example moving obstacle, was the *recovery procedure* (Section II-D). This feature, combined with the path planner in partial target computation, avoided algorithm to fail in case of conflicts in the occupancy map.

Future directions of the proposed work will be to first improve Pepper 3D vision, that have been the main limitation of the platform. Extrinsic and intrinsic calibration of RGB-D cameras and the related point cloud noise reduction could be resolved using *RGB-D Calibration packet* proposed by [23]. This improvement will allow 3D point cloud localization integration, augmenting its reliability in obstacle avoidance and adaptability to complex environment. With regard to navigation, it is intention of the authors testing as input a full *3D OctoMap*, either for localization and trajectory estimation toward partial target. This should generate a similar navigation control to `3d_navigation stack` ROS package, not available for recent versions of the robotic operating system. Final improvements will be addressed to reduce workload on the user. First approach to pursuit will be to integrate classification in object recognition, correlating class of obstacles to actions to take. People detection and tracking could represent additional features to relieve users from the burden of navigation and control attention. Additional relaxation in user control can be provided using Intentional Non-Control [24], which should detect when the user does not want to perform any motor task. Algorithm therefore should act in slowing or even stopping integration of the command delivery output, when such condition is present.

The benefits of integrating ROS in BCI driven devices are not limited to telepresence purposes. For instance, the robustness and the reliability provided by ROS can be exploited to encourage the use of the BCI in more sensitive domains such as car applications [25], rehabilitation [26], pediatric interventions [27] and pain mitigation [28].

ACKNOWLEDGMENT

This research was partially supported by Fondazione Salus Pueri with a grant from "Progetto Sociale 2016" by Fondazione CARIPARO, by Consorzio Ethics with a grant for the project "Hybrid Human Machine Interface", and by Omitech srl with the grant "O-robot". We thank Dott. Roberto Mancin and Omitech s.r.l. for hardware support.

REFERENCES

[1] U. Chaudhary, N. Birbaumer, and A. Ramos-Murguialday, "Brain-computer interfaces for communication and rehabilitation," *Nature Review Neurology*, vol. 12, no. 9, pp. 513–525, 2016.

[2] N. Birbaumer, N. Ghanayim, T. Hinterberger, I. Iversen, B. Kotchoubey, A. Kubler, J. Perelmouter, E. Taub, and H. Flor, "A spelling device for the paralysed," *Nature*, vol. 398, no. 6725, pp. 297–298, 1999.

[3] F. Galán, M. Nuttin, E. Lew, P. Ferrez, G. Vanacker, J. Philips, and J. del R. Millán, "A brain-actuated wheelchair: Asynchronous and non-invasive braincomputer interfaces for continuous control of robots," *Clinical Neurophysiology*, vol. 119, no. 9, pp. 2159 – 2169, 2008.

[4] R. Leeb, L. Tonin, M. Rohm, L. Desideri, T. Carlson, and J. del R. Millán, "Towards independence: A bci telepresence robot for people with severe motor disabilities," *Proceedings of the IEEE*, vol. 103, no. 6, pp. 969–982, June 2015.

[5] L. Tonin, R. Leeb, M. Tavella, S. Perdikis, and J. d. R. Milln, "The role of shared-control in bci-based telepresence," in *2010 IEEE International Conference on Systems, Man and Cybernetics*, Oct 2010, pp. 1462–1466.

[6] L. Tonin, T. Carlson, R. Leeb, and J. del R. Millán, "Brain-controlled telepresence robot by motor-disabled people," in *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, Aug 2011, pp. 4227–4230.

[7] K. Goodrich, P. Schutte, F. Flemisch, and R. Williams, "Application of the h-mode, a design and interaction concept for highly automated vehicles, to aircraft," in *25th IEEE Digital Avionics Systems Conference*, 2006, pp. 1–13.

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

[9] M. Bryan, J. Green, M. Chung, L. Chang, R. Scherer, J. Smith, and R. P. N. Rao, "An adaptive brain-computer interface for humanoid robot control," in *2011 11th IEEE-RAS International Conference on Humanoid Robots*, Oct 2011, pp. 199–204.

[10] K. D. Katyal, M. S. Johannes, S. Kellis, T. Aflalo, C. Klaes, T. G. McGee, M. P. Para, Y. Shi, B. Lee, K. Pejsa, C. Liu, B. A. Wester, F. Tenore, J. D. Beaty, A. D. Ravitz, R. A. Andersen, and M. P. McLoughlin, "A collaborative bci approach to autonomous control of a prosthetic limb system," in *2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Oct 2014, pp. 1479–1482.

[11] F. Arrichiello, P. D. Lillo, D. D. Vito, G. Antonelli, and S. Chiaverini, "Assistive robot operated via p300-based brain computer interface," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, May 2017, pp. 6032–6037.

[12] C. Brunner, N. Birbaumer, B. Blankertz, C. Guger, A. Kbler, D. Mattia, J. del R. Milln, F. Miralles, A. Nijholt, E. Opisso, N. Ramsey, P. Salomon, and G. R. Mller-Putz, "Bnci horizon 2020: towards a roadmap for the bci community," *Brain-Computer Interfaces*, vol. 2, no. 1, pp. 1–10, 2015.

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

[14] J. Meng, S. Zhang, A. Bekyo, J. Olsoe, B. Baxter, and B. He, "Noninvasive electroencephalogram based control of a robotic arm for reach and grasp tasks." *Scientific Reports*, no. 6, p. 38565, 2016.

[15] J. del R. Millán, P. W. Ferrez, F. Galán, E. Lew, and R. Chavarriaga, "Non-invasive brain-machine interaction," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 22, no. 05, pp. 959–972, 2008.

[16] M. Carraro, M. Antonello, L. Tonin, and E. Menegatti, "An open source robotic platform for ambient assisted living." in *AIRO@ AI* IA*, 2015, pp. 3–18.

[17] G. Grisettiyz, C. Stachniss, and W. Burgard, "Improving grid-based slam with rao-blackwellized particle filters by adaptive proposals and selective resampling," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, April 2005, pp. 2432–2437.

[18] G. Grisetti, C. Stachniss, and W. Burgard, "Improved techniques for grid mapping with rao-blackwellized particle filters," *IEEE Transactions on Robotics*, vol. 23, no. 1, pp. 34–46, Feb 2007.

[19] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "Octomap: An efficient probabilistic 3d mapping framework based on octrees," *Autonomous Robots*, vol. 34, no. 3, pp. 189–206, Apr. 2013.

[20] D. Fox, W. Burgard, F. Dellaert, and S. Thrun, "Monte carlo localization: Efficient position estimation for mobile robots," in *Proceedings of the Sixteenth National Conference on Artificial Intelligence (AAAI'99).*, July 1999.

[21] A. Hornung, K. M. Wurm, and M. Bennewitz, "Humanoid robot localization in complex indoor environments," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct 2010, pp. 1690–1695.

[22] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robotics Automation Magazine*, vol. 4, no. 1, pp. 23–33, Mar 1997.

[23] F. Basso, A. Pretto, and E. Menegatti, "Unsupervised intrinsic and extrinsic calibration of a camera-depth sensor couple," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, May 2014, pp. 6244–6249.

[24] L. Tonin, A. Cimolato, and E. Menegatti, "Do not move! entropy driven detection of intentional non-control during online smr-bci operations," in *Converging Clinical and Engineering Research on Neurorehabilitation II.* Springer International Publishing, 2017, pp. 989–993.

[25] Y. Yu, Z. Zhou, E. Yin, J. Jiang, J. Tang, Y. Liu, and D. Hu, "Toward brain-actuated car applications: Self-paced control with a motor imagery-based brain-computer interface," *Computers in Biology and Medicine*, vol. 77, no. Supplement C, pp. 148 – 155, 2016.

[26] J. J. Daly and J. E. Huggins, "Brain-computer interface: Current and emerging rehabilitation applications," *Archives of Physical Medicine and Rehabilitation*, vol. 96, no. 3, Supplement, pp. S1 – S7, 2015.

[27] J. D. Breshears, C. M. Gaona, J. L. Roland, M. Sharma, N. R. Anderson, D. T. Bundy, Z. V. Freudenburg, M. D. Smyth, J. Zempel, D. D. Limbrick, W. D. Smart, and E. C. Leuthardt, "Decoding motor signals from the pediatric cortex: Implications for brain-computer interfaces in children," *Pediatrics*, vol. 128, no. 1, pp. e160–e168, 2011.

[28] N. Yoshida, Y. Hashimoto, M. Shikota, and T. Ota, "Relief of neuropathic pain after spinal cord injury by braincomputer interface training," *Spinal Cord Series and Cases*, vol. 2, no. 16021, 2016.