Article

The International Journal of Robotics Research
1–18

DOI: 10.1177/0278364914561535
ijr.sagepub.com
SAGE

An assistive decision-and-control architecture for force-sensitive hand–arm systems driven by human–machine interfaces

J. Vogel[1], S. Haddadin[2], B. Jarosiewicz[3,4], J.D. Simeral[4,5,6], D. Bacher[5], L.R. Hochberg[4,5,6,7], J.P. Donoghue[3,4,5] and P. van der Smagt[8,9]

Abstract

Fully autonomous applications of modern robotic systems are still constrained by limitations in sensory data processing, scene interpretation, and automated reasoning. However, their use as assistive devices for people with upper-limb disabilities has become possible with recent advances in "soft robotics", that is, interaction control, physical human–robot interaction, and reflex planning. In this context, impedance and reflex-based control has generally been understood to be a promising approach to safe interaction robotics.

To create semi-autonomous assistive devices, we propose a decision-and-control architecture for hand–arm systems with "soft robotics" capabilities, which can then be used via human–machine interfaces (HMIs). We validated the functionality of our approach within the BrainGate2 clinical trial, in which an individual with tetraplegia used our architecture to control a robotic hand–arm system under neural control via a multi-electrode array implanted in the motor cortex. The neuroscience results of this research have previously been published by Hochberg et al.

In this paper we present our assistive decision-and-control architecture and demonstrate how the semi-autonomous assistive behavior can help the user. In our framework the robot is controlled through a multi-priority Cartesian impedance controller and its behavior is extended with collision detection and reflex reaction. Furthermore, virtual workspaces are added to ensure safety. On top of this we employ a decision-and-control architecture that uses sensory information available from the robotic system to evaluate the current state of task execution. Based on a set of available assistive skills, our architecture provides support in object interaction and manipulation and thereby enhances the usability of the robotic system for use with HMIs. The goal of our development is to provide an easy-to-use robotic system for people with physical disabilities and thereby enable them to perform simple tasks of daily living. In an exemplary real-world task, the participant was able to serve herself a beverage autonomously for the first time since her brainstem stroke, which she suffered approximately 14 years prior to this research.

Keywords

Service robots, assistive robotics, medical robots and systems, human-centered and life-like robotics, telerobotics

1. Motivation and problem statement

The application of robotic systems in rehabilitation, prosthetics, and in particular assistive technology has long been an area of research interest. Robotic systems could potentially be very useful in helping people with severe physical disabilities. For instance, for many people with tetraplegia, simple activities required in daily living such as drinking, opening a door, or pushing an elevator button, require the assistance of a caretaker, reducing the independence of the individual. An assistive robotic system (as illustrated in Figure 1) could enable people with tetraplegia to achieve greater independence and thereby increase quality of life.

The number of assistive robots available on the market is limited. Examples of such assistive robotic devices are

[1]Institute of Robotics and Mechatronics, Robotics and Mechatronics Center, German Aerospace Center (DLR), Oberpfaffenhofen, Germany
[2]Institute of Automatic Control, Leibniz University Hanover, Germany
[3]Department of Neuroscience and Institute for Brain Science, Brown University, Providence, RI, USA
[4]Center for Neurorestoration and Neurotechnology, Department of Veterans Affairs, Providence, RI, USA
[5]School of Engineering and Institute for Brain Science, Brown University, Providence, RI, USA
[6]Department of Neurology, Massachusetts General Hospital, Boston, MA, USA
[7]Harvard Medical School, Boston, MA, USA
[8]Fortiss, an-Institut der Technischen Universität München, Germany
[9]BRML, Department of Informatics, Technische Universität München, Germany

Corresponding author:
J. Vogel, Institute of Robotics and Mechatronics, Robotics and Mechatronics Center, German Aerospace Center (DLR), 82234 Oberpfaffenhofen, Germany.
Email: joern.vogel@dlr.de

Fig. 1. An assistive torque-controlled robotic hand–arm system to support people with severe physical disabilities.

the Manus and the iARM by Exact Dynamics (Römer et al., 2004) and the JACO Arm by Kinova Robotics (Maheu et al., 2011). Such systems are generally controlled by a joystick, allowing the operator to move the arm in Cartesian space. A mechanical gripper, which can be opened and closed, is used to interact with objects. A joystick offers a rather intuitive interface, and thus provides some help to people with limited movement capabilities. However, the lack of apt haptic feedback both in the robot control loop and on the human–machine interface level makes it very tedious to perform simple tasks like safely picking up, holding, or releasing an object. While a joystick provides a reliable and transparent interface, the issue of missing feedback becomes more substantial when using interfaces that are less transparent or of varying quality. In these cases it would be beneficial to add locally autonomous capabilities to the robotic system in order to support the user in object interaction. An analogy to the concept of assisted teleoperation exemplifies the utility of this approach. As described by Bohren et al. (2013), teleoperation of a robotic system becomes very tedious when the control channel is low-dimensional. This problem is even more apparent when the only available feedback is given by visual information, which might, at least in the teleoperation scenario, additionally be affected by delays. To improve the usability of such a system, Bohren et al. added perception-based semi-autonomous capabilities to a teleoperated robotic system.

Different levels of autonomy can be conceived, from fully autonomous functionality of the robot, where the user only selects an action to be executed, down to fault detection and isolation, in which the robot would modify its behavior only in the case of erroneous input. The level of autonomy inherently has implications on the control capabilities (transparency) needed for the user interface. Thus, in a highly autonomous system, a low-dimensional input, such as a small number of discrete signals, would be sufficient to enable the user to select the required action from a pool of available robot functionalities. On the other hand, full autonomous functionality of robotic systems is still constrained by limitations in, for example, sensory data processing and scene interpretation and therefore not currently attainable, especially in unstructured domestic scenarios.

Furthermore, a fully autonomous device would limit the user to the functions provided by the system and possibly require external updates to adapt its capabilities to each user's unique environment. Thus we envision a semi-autonomous approach to be most beneficial to people with limited movement capabilities. Ideally, the user should be able to freely and intuitively move the robotic system in space, but the system should support the user through partly autonomous behavior at task level. This supportive behavior would simplify the use of a robotic hand–arm system when controlled by a joystick as well. However, for locked-in or tetraplegic individuals a mechanical interface such as a joystick may not be suitable. Also in people with amyotrophic lateral sclerosis (ALS), spinal muscular atrophy (SMA), or other degenerative diseases, functional limb control capabilities can be extremely limited, and joysticks cannot be used. For these individuals, alternative ways to control technical systems have to be used where the benefits of semi-autonomous systems become even more evident.

To provide this semi-autonomous behavior to robotic systems, we developed a decision-and-control architecture that leverages the potential of so-called soft-robotics features in combination with invariant decisional skill automata. These enable the user to activate certain capabilities such as "grasping", "placing", or "drinking", which can then be executed autonomously.

The generic gateway to our architecture is based on continuous desired Cartesian velocity input signals, and thus can be used in combination with any human–machine interface (HMI) that provides this control modality. In a pilot study, which serves as the template scenario, we used this architecture to enable a person with tetraplegia to autonomously pick up and drink from a bottle by controlling a robotic hand–arm system via the BrainGate2 Neural Interface System. The neural decoding component of this research has previously been published in *Nature* (Hochberg et al., 2012); here, we report on the robotics component of this work and how our decision-and-control architecture can help enable the execution of "real-world" related tasks such as the one presented here.

The paper is organized as follows: Section 2 provides a general overview of HMIs and their applicability for our approach. Section 3 explains in detail our decision-and-control architecture. Section 4 introduces the BrainGate2 Neural Interface System, which we used to demonstrate the functionality and feasibility of our approach. Section 5 explains the research sessions we conducted and Section 6 presents the achieved results. Section 7 gives a short introduction to our new approach towards

an "electromyography (EMG)-based" interface, which will be used in combination with our framework in the future. Finally, Section 8 contains our conclusions.

2. HMIs

Manual interfaces such as a computer mouse, keyboard, or joystick serve as an effective control device for technical systems for able-bodied people. However, for people with physical disabilities, a variety of techniques have been developed to substitute for these manual interfaces. A common application in need of alternative interfaces is the steering of a wheelchair. For people who retain some head movement, for example, a special joystick can be mounted close to the chin and operated using neck movement. For people who are unable to operate a joystick-like device at all, head switches or *sip-and-puff* controls are commonly used as an alternative. A more sophisticated device is the Tongue Drive (Krishnamurthy and Ghovanloo, 2006), in which a patch of touch sensors is fixed on the palate and can be activated with the tip of the tongue. Another way of generating discrete control signals is by measuring the user's muscular activity. In Felzer and Freisleben (2002) this is realized by recording eyebrow movements via piezo-based sensors. EMG can be used as an alternative sensing technology for this approach. All of these alternatives provide viable control capabilities for low-dimensional applications, such as wheelchair steering, but it is difficult to extract control signals with more degrees of freedom (DoF) or continuous signals from these low-dimensional input devices.

We envision continuous control signals to be advantageous for controlling assistive robotic devices. A widespread approach is the use of eye tracking systems (Jacob and Karn, 2003). These devices have proven to be effective in 2D control tasks, for example, moving a cursor on a computer screen. In the low-dimensional application of point and click on a computer screen, they can actually outperform standard HMIs like a computer mouse in terms of speed (Vertegaal, 2008). However, eye tracking does have limitations, for example if a person wants to look somewhere on a computer screen without necessarily making a selection there, when higher-dimensional control signals ($\geq$ 3 DoF) are required (Duchowski et al., 2011), or when eye movements are unreliable or imprecise due to neurological disease or injury. Continuous control signals can also be acquired using surface EMG (sEMG). sEMG is a widely used approach for the control of hand prosthesis, either at the position level (Bitzer and van der Smagt, 2006; Smith et al., 2008), or in the domain of finger forces (Castellini and van der Smagt, 2009). In healthy subjects, recordings of muscle activations are also used to teleoperate robotic systems through position commands (Artemiadis and Kyriakopoulos, 2010; Vogel et al., 2011) or to improve the control of exoskeleton devices (Rosen et al., 2001). The latter approach can also be applied to robot-aided rehabilitation of hand (Mulas et al., 2005) or arm (Andreasen et al., 2005) function after stroke or other injury. With respect to assistive robotic systems, recent work shows that sEMG can also be used as a continuous velocity-based interface for people with severe muscular atrophy. In Vogel et al. (2013), two individuals with SMA were able to control a virtual robotic hand–arm system in a simulated version of our devised architecture.

All of the aforementioned interfaces require that the user has some residual movement or muscle activity. However, for people with no residual movement or muscular activity, acquisition of control signals can be realized by the use of brain–computer interfaces (BCIs).

In BCI research, there is a variety of approaches to measure neural activity and extract control signals from it. These methods can be distinguished by the spatial resolution of the neural recording, the achievable temporal resolution, the need for surgery, and the general applicability of the interface with respect to long-term use and mobility. To control an assistive robotic system via BCI, some neural recording techniques are not practical. Methods such as functional magnetic resonance imaging (fMRI) (Weiskopf et al., 2004) and magnetoencephalography (MEG) (Mellinger et al., 2007) require large-scale equipment and specially shielded environments, making them impractical for home or mobile use. Furthermore, fMRI and near-infrared spectroscopy (NIRS) (Ishikawa et al., 2007) provide a prohibitively low temporal and spatial resolution, with delays on the order of seconds (Hornyak, 2006) due to the properties of the underlying biological process, the hemodynamic response (Miezin et al., 2000; Cui et al., 2010). It is known from physiological experiments, especially in the context of tele-robotics, that a temporal delay of more than 250 ms significantly reduces the user's ability to control a robotic system (Kim et al., 2005). Therefore, the BCI recording technique should detect and relay neural events within $\approx$ 0.1 s in order to keep the overall delay low and also allow time for the decoding process. There is a substantial difference in temporal as well as in spatial resolution between implanted and non-implanted technologies (Figure 2).

The key features for different neural recording methods are listed in Table 1. If we take impractical recording techniques and those with poor temporal resolution out of consideration, we are left with the following methods:

- Electroencephalography (EEG);
- Electrocorticography (ECoG);
- Microelectrode arrays (MEAs).

The spatial resolution of EEG is limited relative to implanted interfaces. As a consequence, signals acquired via EEG are usually used for decoding discrete classes or sets of trajectories (Nijboer et al., 2008; Bradberry et al., 2010). Another common approach is in mapping motor imagery of single limbs (such as left hand vs right hand) to single DoF (McFarland et al., 2010). Online naturalistic

Table 1. Brain–machine interfaces and their properties and practicability in controlling assistive robotic devices for domestic use.

	Desired spatial resolution	Desired temporal resolution	Implanted	Practicability
NIRS	–	–	–	×
fMRI	×	–	–	–
EEG	–	×	–	×
MEG	–	×	–	–
ECoG	×	×	×	×
MEA	×	×	×	×
ME	×	×	×	–

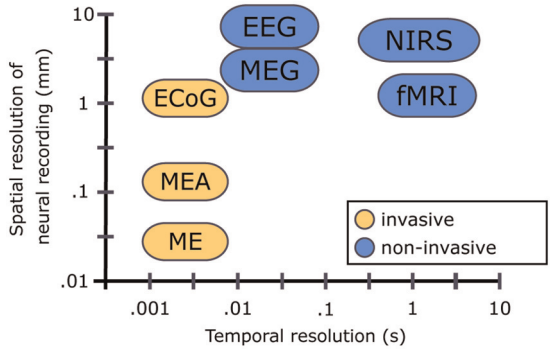

Fig. 2. Overview of different brain–machine interfacing methods and their spatial and temporal resolution. Methods included: electroencephalography (EEG), magnetoencephalography (MEG), near-infrared spectroscopy (NIRS), functional magnetic resonance imaging (fMRI), electrocorticography (ECoG), microelectrode array (MEA) recordings and single microelectrode (ME) recordings. Reproduced with kind permission from IOP Publishing (Gerven et al., 2009).

continuous control has not been attained to date. In order to establish continuous control in Cartesian space, these discrete events can be transformed into graded control signals, but this approach leads to low bit rates. Such methods, still limited to 1D or 2D control in reported cases (Pascual et al., 2012), can be used to control lower-dimensional devices such as a wheelchair or for communication interfaces. However, at their current state of development, they seem to be unsuitable for control of assistive robotic hand–arm systems.

With current technology, higher bit rates can only be achieved with intracranial recording methods such as ECoG or a MEA. ECoG measures neural signals at the cortical surface, obtained by implanting a multi-electrode patch. Its common clinical application is the localization of brain areas involved in epileptic seizures. However, it has also been used as a signal source for BCIs. Most ECoG BCI studies to date distinguished execution or imagery of several different body parts in order to decode multi-dimensional signals online (Miller et al., 2007; Schalk et al., 2008). The more intuitive approach of mapping single limb intended movement to Cartesian control commands (continuous or discrete) has primarily only been applied offline, except for two recent publications: one that describes decoding of 1D single-limb movement direction online from ECoG signals (Milekovic et al., 2012), and one in which online multi-dimensional continuous decoding using single limb movement imagination is reported (Wang et al., 2013).

Higher-dimensional naturalistic BCI control in people has also been obtained with implanted MEAs (Hochberg et al., 2006), which have a much better signal-to-noise ratio than other BCI interface technologies. MEAs have been used to decode neural signals from individuals with tetraplegia to provide control of a computer cursor (Hochberg et al., 2006; Kim et al., 2007, 2008) or assistive robotic devices (Hochberg et al., 2006, 2012; Collinger et al., 2013), even after long-term implantation (Simeral et al., 2011; Hochberg et al., 2012; Jarosiewicz et al., 2013). To implement a naturalistic and intuitive interface capable of continuously controlling a robotic device, the spatial recording resolution needs to be sufficient to record neural activity representing desired motion of a single limb. Previous results with the above-mentioned recording techniques suggest that this control paradigm can more readily be achieved with the high spatial and temporal recording resolution provided by MEAs.

Though BCIs do not yet provide as precise control as direct interfaces such as a joystick, the potential availability of continuous control signals in combination with a more natural control representation make MEA-based BCI a promising interface for assistive systems when other HMIs cannot be used. To help compensate for the present limitations of BCI interfaces, it is necessary to add autonomous supportive behavior to the system to be controlled. The greater the bandwidth and signal stability/reliability of the BCI, the lower the autonomy of the assistive robotic system needs to be. While sensory feedback is expected to enhance the controllability, a bidirectional integration between the human body and an assistive robotic device is still very much a research endeavor. Although the possibility of adding sensory feedback via neural interfaces is being investigated (Thomson et al., 2013; Raspopovic et al., 2014), in current studies the interface remains unidirectional, with no feedback available except for visual observation of commanded actions (Arbib et al., 2008). Some interfacing technologies, like implanted neural interfaces, EEG, or surface EMG, provide shorter feedback delays, which improve control. However, the absence of somesthetic feedback may

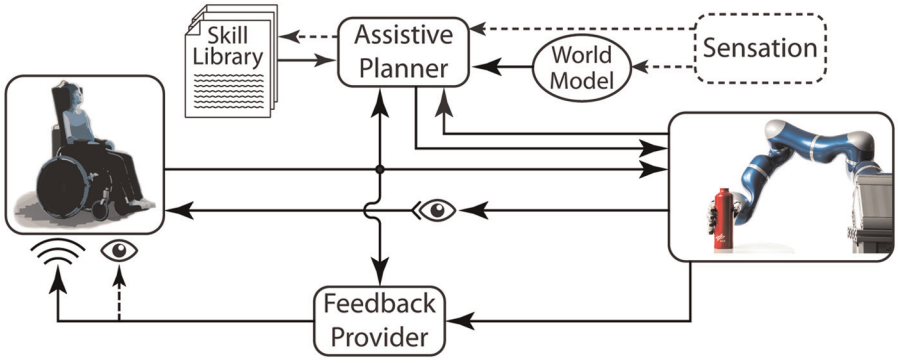

Fig. 3. Schematic overview of our devised assistive decision-and-control architecture. The user controls the robotic system via HMI, while observing the robot's movement. The Assistive Planner can modulate the robot's behavior according to the skills available from the Skill Library. A World Model and external Sensation provide additional information for skill execution. To increase transparency of the system, the Feedback Provider supplies the user with auditory and visual feedback about the robot's state. Note that components drawn with solid lines indicate the current state of implementation, while dashed components are envisioned features for future development.

impede formation of a "tool" extension of the body scheme (Robles-De-La-Torre, 2006; Giummarra et al., 2008). Thus, using assistive systems is not yet as natural as might be possible. A semi-autonomous robotic system providing help in the complex task of object grasping and manipulation can possibly help to increase usability.

3. Technical approach

Accepting the limitations of current HMIs, we set out to devise an assistive robotic decision-and-control architecture to provide semi-autonomous behavior through combinations of various soft-robotics features, freely combinable skill libraries, assistive sensing capabilities, and appropriate, multimodal feedback channels to the user.

Devised architecture: Figure 3 depicts a schematic overview of the core components of our envisaged architecture. It consists of a fully torque-controlled robotic hand–arm system, which is operated via a HMI on continuous velocity domain and a binary trigger signal. Furthermore, it consists of an assistive planner, which modulates the robot's behavior based upon the HMI input. The assistive planner has access to a set of functionalities represented in a skill library. A skill (for example, picking up a cylindrical object) is an activity triggered by the user via a binary trigger signal. A skill is parameterized by information (for example, diameter of the object to be grasped) taken from a world model. The world model also provides information related to safety features to the robotic system, such as the position of the user and the allowed workspace limits. Information needed by the assistive planner in order to estimate the state of skill execution is provided by sensory information of the torque-controlled robotic system. Furthermore, external sensors (e.g. cameras) can provide additional information to implement more sophisticated skills (see Bohren et al., 2013, for examples of perception-based grasping). This external information can also be used to dynamically adapt the world model, for example, with information about recognized objects and the position of the user. Our ultimate future goal for an assistive framework would be to allow the user to teach new skills to the skill library, and thus to fully adapt the system to personal requirements. The architecture also contains a feedback provider, which extends the feedback available to the user beyond visual information about the robot motion. The feedback provider features audible feedback based on the assistive planner and the state of the robot. For instance, contact with workspace limits elicits an acoustic signal to inform the user. Also, visual cues about the state of skill execution can be provided to the user in order to increase usability and transparency of the system.

Basis for implementation: A computer vision system is not yet integrated in the current state of implementation. Thus, the world model is static and manually defined in accordance with the task. To achieve assistive behavior without vision or symbolic information, a torque-controlled robotic system provides many possibilities for sophisticated object and environment interaction due to its sensing of interaction forces. In our assistive framework, the robotic system is controlled via the DLR *Beasty* control suite (Parusel et al., 2011) that encapsulates the high-performance DLR algorithms for interaction control (Albu-Schäffer et al., 2007b), collision detection, and reflex planning (Haddadin et al., 2008). Furthermore, *Beasty* is also equipped with algorithms for intuitive kinesthetic teaching of virtual workspace boundaries that also act on torque level. We extended the system with our assistive planner, combining task execution at the hand and arm level, while being tightly integrated with the real-time control layer. The developed framework offers powerful yet highly flexible assistive skills to enable a robotic device for safe interaction with its surroundings while allowing a closed-loop interface. In the next section, a concise bird's-eye view of the applied controller and continuous velocity control with according safety filters is described.

3.1. Robot system and control

In our work we use the DLR light-weight robot arm LWR-III (Albu-Schäffer et al., 2007a) and the DLR five-finger hand (Liu et al., 2008) as a robotic platform. The LWR-III is a seven-DoF anthropomorphic robotic arm that weighs 14 kg. The integrated joint torque sensors can be used to realize direct torque control which is, for example, a prerequisite to implement true torque-based Cartesian impedance control (Albu-Schäffer et al., 2007a,b). Such a control approach is known to provide stable behavior when the system is in contact situations, but can also be used in situations where the commanding signal is of varying quality, which we exploit in our architecture. Additionally, other soft-robotics features including collision detection and reflex reaction (Haddadin et al., 2008) or virtual workspace limitations (Haddadin et al., 2011) allow for safe human–robot interaction, which is absolutely necessary in assistive robotic scenarios.

In order to incorporate a virtual environment into the control scheme and enable reflex-like reactions to collisions and contacts, we need to generate virtual obstacle forces and be able to estimate the external torques acting along the robot structure. Generally, the rigid-body dynamics of an articulated robot[1] can be described by

$$M(\boldsymbol{q})\ddot{\boldsymbol{q}} + C(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{\tau}_m + \boldsymbol{\tau}_{\text{ext}} \tag{1}$$

where $M(\boldsymbol{q})$ is the inertia matrix, which also includes the dynamics of the robotic hand and grasped objects, $C(\boldsymbol{q},\dot{\boldsymbol{q}})$ contains the Coriolis and centrifugal terms, and $\boldsymbol{g}(\boldsymbol{q})$ is the gravity vector. Furthermore, $\boldsymbol{\tau}_m$ denotes the motor torque and $\boldsymbol{\tau}_{\text{ext}}$ are the external torques due to contact with the environment. The motor torque is defined by $\boldsymbol{\tau}_m := \boldsymbol{\tau}_{d,\text{imp}} + \boldsymbol{\tau}_{d,\text{VE}}$, where $\boldsymbol{\tau}_{d,\text{imp}}$ is the Cartesian impedance controller, given by

$$\boldsymbol{\tau}_{d,\,\text{imp}} = J_{\text{EE}}^{\text{T}}(K\tilde{\boldsymbol{x}} + D\dot{\boldsymbol{x}}) + \boldsymbol{g}(\boldsymbol{q}) \tag{2}$$

$\dot{\boldsymbol{x}}$ is the Cartesian end-effector velocity and $\tilde{\boldsymbol{x}} = \boldsymbol{x} - \boldsymbol{x}_d$ is the position error between the robot position $\boldsymbol{x}$ and the desired position $\boldsymbol{x}_d$. The torque $\boldsymbol{\tau}_{d,\text{VE}}$ generated by the virtual environment for environment protection and representation of workspace limits is defined as

$$\boldsymbol{\tau}_{d,\,\text{VE}} = \sum_{i=1}^{n} J_{\text{EE}}^{\text{T}}\boldsymbol{f}_{\text{V}i} + \mathcal{N}J_{\text{elbow}}^{\text{T}}\boldsymbol{f}_{\text{elbow}} \tag{3}$$

$$\text{where } \boldsymbol{f}_{\text{V}i} = K_{\text{V}i}\tilde{\boldsymbol{x}}_i + D_{\text{V}i}\dot{\boldsymbol{x}}_i \tag{4}$$

The virtual environment consists of workspace boundary wrenches $\boldsymbol{f}_{\text{V}i}$ (virtual walls), which the robot cannot overcome.[2] As the LWR-III is kinematically redundant, we also designed an appropriate nullspace controller for letting the elbow show an intuitive behavior. The elbow forces are projected via $\mathcal{N}$ into the nullspace of the end-effector task space.

In order to account for the workspace limits also on the desired position/velocity level, we define the desired position $\boldsymbol{x}_d$ as

$$\boldsymbol{x}_d = \int_0^t \left[k\dot{\boldsymbol{x}}_{\text{decoded}} - \boldsymbol{c}(\boldsymbol{f}_{\text{V}}, \dot{\boldsymbol{x}}_{\text{decoded}})\right] \tag{5}$$

where $k \geq 0$ is a gain factor that adjusts the responsiveness of the robot with respect to the desired velocity $\dot{\boldsymbol{x}}_{\text{decoded}}$ provided by the HMI. Note that the same controller could be applied at joint level, if required. The state-dependent vector $\boldsymbol{c}(\boldsymbol{f}_{\text{V}}, \dot{\boldsymbol{x}}_{\text{decoded}})$ accounts for hitting workspace limits by stopping integration of the decoded velocity signal if such a boundary is hit and the velocity direction points into the wall:

$$\boldsymbol{c}(\boldsymbol{f}_{\text{V}}, \dot{\boldsymbol{x}}_{\text{decoded}}) = \begin{cases} \frac{-\boldsymbol{f}_{\text{V}}\cdot\dot{\boldsymbol{x}}_{\text{decoded}}}{\|\boldsymbol{f}_{\text{V}}\|\|\dot{\boldsymbol{x}}_{\text{decoded}}\|}\frac{\boldsymbol{f}_{\text{V}}}{\|\boldsymbol{f}_{\text{V}}\|} & , \text{if } -\boldsymbol{f}_{\text{V}}\cdot\dot{\boldsymbol{x}}_{\text{decoded}} > 0 \\ 0 & , \text{else} \end{cases} \tag{6}$$

where $\boldsymbol{f}_{\text{V}} = \sum \boldsymbol{f}_{\text{V}i}$ is the workspace boundary wrench generated by the virtual environment.

This shaping function prevents the desired position (equilibrium point) from moving into the wall, in other words, it maintains the balance between a repelling virtual wall and attracting impedance forces. Finally, enabling the robot to react to physical contact with the environment requires estimating the external torques $\boldsymbol{\tau}_{\text{ext}}$ acting along the entire robot structure. For this, we used the nonlinear disturbance observer introduced in Haddadin et al. (2008). This generates a very accurate estimation of $\boldsymbol{\tau}_{\text{ext}}$, which can be used for various response structures. In particular, it can be used to react to different collision severity levels or exploit the sensory information about external contact forces in more complex patterns for grasping and placing objects.

In this study, the LWR-III is combined with the DLR five-finger hand (Liu et al., 2008). This robotic hand has three DoF in each finger and, similar to the robot, uses torque sensing in each joint. Thus it can be operated in impedance control mode as well, which allows for stable grasping of a variety of objects, even with a limited control signal like a binary grasp trigger. The dynamics of a single finger (denoted by the subscript $_{\text{f}}$) can be described analogously to an arm (equation (1)):

$$M_{\text{f}}(\boldsymbol{q}_{\text{f}})\ddot{\boldsymbol{q}}_{\text{f}} + C_{\text{f}}(\boldsymbol{q}_{\text{f}},\dot{\boldsymbol{q}}_{\text{f}})\dot{\boldsymbol{q}}_{\text{f}} + \boldsymbol{g}_{\text{f}}(\boldsymbol{q}_{\text{f}}) = \boldsymbol{\tau}_{\text{f m}} + \boldsymbol{\tau}_{\text{f ext}} \tag{7}$$

The control of the individual fingers is realized with a simple proportional-derivative (PD) position controller

$$\boldsymbol{\tau}_{\text{f d}} = K_{\text{P}}(\boldsymbol{q}_{\text{f}} - \boldsymbol{q}_{\text{f d}}) + K_{\text{D}}(\dot{\boldsymbol{q}}_{\text{f}} - \dot{\boldsymbol{q}}_{\text{f d}}) \tag{8}$$

cascaded with a PD joint torque controller

$$\boldsymbol{\tau}_{\text{f m}} = K_{\text{T}}(\boldsymbol{\tau}_{\text{f}} - \boldsymbol{\tau}_{\text{f d}}) + K_{\text{J}}\dot{\boldsymbol{\tau}}_{\text{f}} \tag{9}$$

This leads to a spring-like behavior of the individual fingers, which can be used to grasp objects while detecting the

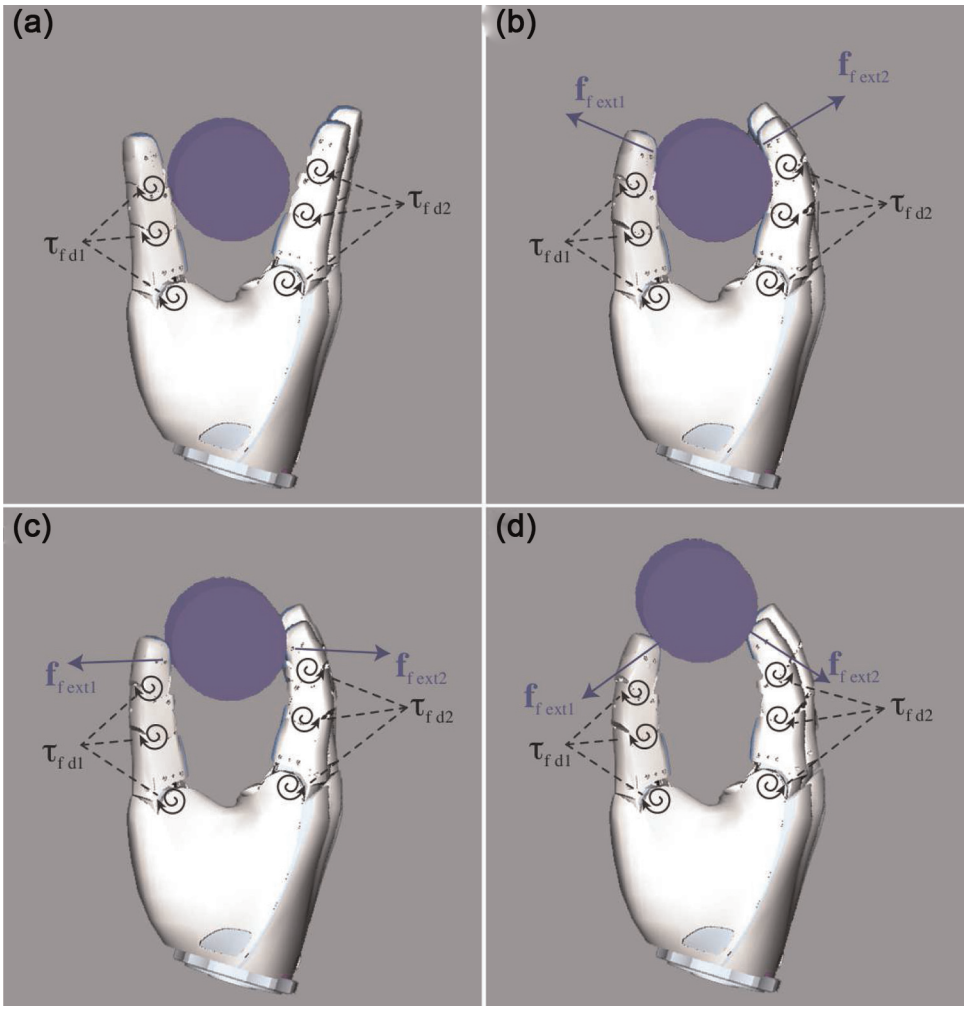

Fig. 4. Different object contact situations: (a) depicts the hand when approaching an object and contact has not been established yet; (b) depicts a valid grasp situation, where the object is fully enclosed by the fingers. In (c) the object is not fully enclosed and the object might be able to tilt/slip, and (d) depicts a grasping situation in which the object is likely to be dropped, because the interaction forces push the object out of the hand.

validity of a grasp. Figure 4(a) shows the hand in which the torques $\boldsymbol{\tau}_{f\,d,i}$ for each finger generate a defined finger stiffness. To grasp an object which is known in size and shape (see Figure 4(b)), interaction forces with the object have to be applied such that the force closure principle described by Nguyen (1986) is fulfilled. Assuming quasi-statics during contact situation ($\ddot{\boldsymbol{q}}_f \equiv \dot{\boldsymbol{q}}_f \equiv \mathbf{0}$) and compensated finger mass $\boldsymbol{g}_f(\boldsymbol{q}_f)$, the interaction wrench may be estimated by $\boldsymbol{f}_{fd} \approx J_f^{-T}\boldsymbol{\tau}_f$. This requires J_f^T to be invertible, which can be ensured by selecting a suitable non-singular grasp configuration. As point contact is assumed, the interaction wrench $\boldsymbol{f}_{fd}$ consists of force components only. Knowing the object's mass and friction properties between finger and object surface, a grasp force $\boldsymbol{f}_{fd}$ can be chosen such that the object does not slip when being grasped. To achieve this force, $\boldsymbol{q}_{fd}$ and K_p have to be chosen such that the resulting $\boldsymbol{\tau}_{fd}$ leads to the desired force $\boldsymbol{f}_{fd}$. The resulting force $\boldsymbol{f}_{fext}$ actually applied to the object is dependent on the position of the object within the hand and can be estimated from the measured joint torques to be $\boldsymbol{f}_{fext} \approx J_f^{-T}\boldsymbol{\tau}_{fext}$. Thus, from $\boldsymbol{f}_{fext}$ it is possible to detect the validity of a grasp. Figure 4(c) shows a grasp configuration in which the object can be grasped, but the grasp configuration does not comply with the force closure principle. Therefore, the grasp is not stable, and the object may be dropped. This is also the case for the grasp configuration depicted in Figure 4(d). Thus, the direction and magnitude of the interaction forces $\boldsymbol{f}_{f\,ext,i}$ may be used to detect unstable grasp configurations:

$$\tilde{\gamma} = \operatorname{acos}\frac{\boldsymbol{f}_{fext}\cdot\boldsymbol{f}_{fd}}{\|\boldsymbol{f}_{fext}\|\|\boldsymbol{f}_{fd}\|} < \gamma_{lim} \tag{10}$$

$$\tilde{f} = \|\boldsymbol{f}_{fext}\| - \|\boldsymbol{f}_{fd}\| \quad \text{with} f_{lim\_low} < \tilde{f} < f_{lim\_high} \tag{11}$$

The combination of $\tilde{\gamma}$ and $\tilde{f}$ corresponds to the principle of a friction cone in which the measured grasp force has to be contained. In the conducted experiments feasible values

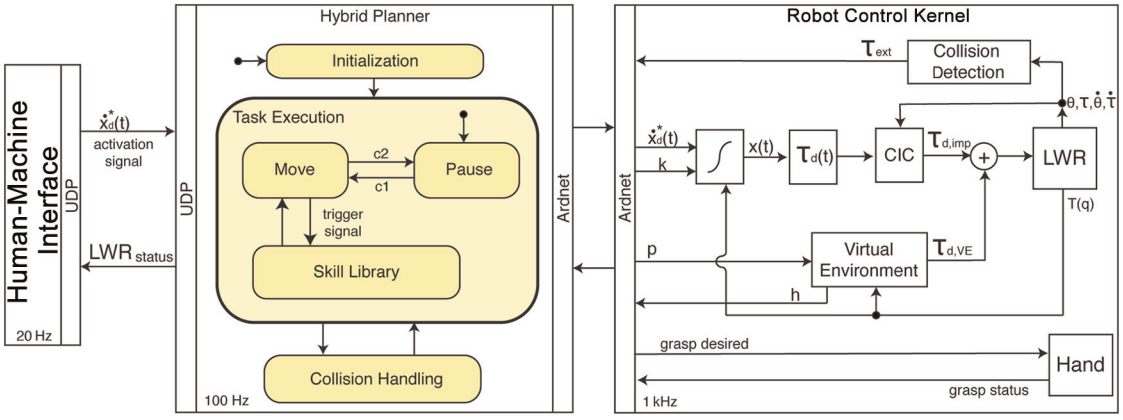

Fig. 5. Simplified block diagram of the overall control architecture and signal flow. Depending on the state of the system, the high-level assistive planner communicates desired motion to the low-level control kernel. Behavior of the robotic arm and hand are handled in the two sub-states "Move" and "Skill Library". In the "Move" substate, the user has free velocity-based control over the robotic system. Via the binary trigger signal, an assistive skill from the "Skill Library" is activated. In the robot control kernel a motion generator interpolates the motion commands to the control rate of 1 kHz and the Cartesian impedance controller calculates the desired torques that are commanded to the hardware. Safety algorithms like collision detection and reaction or the virtual environment are also handled within the robot control kernel.

for the angular limit γ_{lim} and the force limits $f_{\text{lim\_low}}$ to prevent slip, and $f_{\text{lim\_high}}$ to limit maximum force, were empirically found in order to detect grasp stability and automatically avoid the risk of dropping objects.

3.2. High-level control concept

To fully exploit the interaction capabilities provided by a torque-controlled robotic arm and hand, our assistive planner is employed by a high-level control layer in tight interrelation with *Beasty* (Haddadin et al., 2010; Parusel et al., 2011). A schematic overview of the control architecture is depicted in Figure 5. In the low-level control kernel, all algorithms run at a 1 kHz rate and execute the basic motion control, virtual environment, and collision detection with reaction patterns. The connection to the high-level control is realized via the real-time communication protocol Ardnet (Bäuml and Hirzinger, 2008). The control signals stemming from the HMI are communicated to the high-level control via user datagram protocol interface.

Assistive planner: The assistive planner is implemented by means of a state machine in order to coordinate activation and execution of skills. A skill is generally activated by the user via the binary trigger signal. The torque-based sensing capabilities of the robotic system are then used to detect the current state of execution within the skill. Considering pick-and-place of a known cylindrical object as an exemplary task, one can derive subtasks, which can be supported by the robotic system. These subtasks correlate to skills in our skill library. For instance, the torque and position sensors from the hand can be used to evaluate grasp stability, in order to compensate for the lack of haptic feedback (see Section 3.1 equations (10) and (11)). Figure 6 depicts the part of the assistive planner that enables the user to command the hand to execute a grasp or release skill.

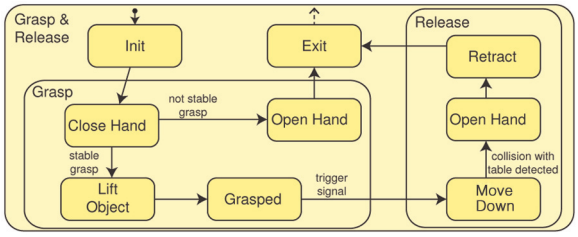

Fig. 6. Simplified visualization of an assistive planner for picking up and putting down an object controlled by one binary activation signal. Dependent on the state of the robot, different skills can be activated by the binary signal.

Grasp and release: When the grasp skill is activated, the robot automatically stops moving and the hand performs a grasping motion. Now two scenarios can apply. In the case of a successful force-closure grasp, the robot automatically lifts the object from the table surface and then control is given back to the user. If grasp stability is not ensured, the hand will open again and the user can reposition before grasping again. As we have not yet integrated a vision system, repositioning has to be performed by the user. With integrated vision, the framework could also support the user in the approach phase towards an object, and thereby guarantee achieving a stable grasp. After picking up the object, the functionality of the trigger signal is changed to activate the "release" skill. This skill, when activated, will command the robot to move downwards until contact with the table is detected by the collision detection algorithm. Only after this contact has been established will the hand

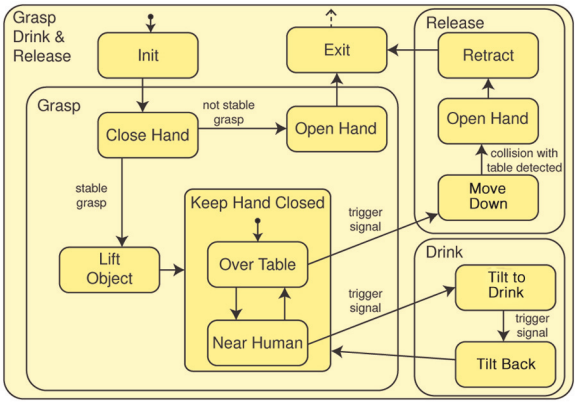

Fig. 7. Simplified visualization of an assistive planner for a drinking task. Dependent on the state of the robot different skills can be activated by the binary trigger signal.

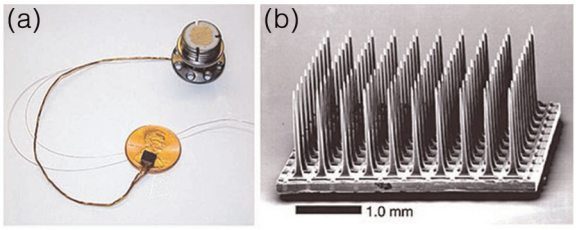

Fig. 8. (a) Electrode array and the pedestal that provides the connection through the skull. (b) Close-up of the electrode array. Reproduced with kind permission from Nature Publishing Group (Hochberg et al., 2006).

open and release the object. To provide further support, a retract motion of the robot is executed automatically to safely release the object. During these predefined release movement patterns, decoded velocity commands are not communicated to the robot in order to simplify the controllability for the user. A demonstration of the grasp and release skills is provided in Extensions 1 and 2.

Drinking skill: This set of skills can now be expanded to enable the user to move the bottle over to the mouth and drink from it. Figure 7 depicts an extended version of the "grasp and release" planner, which enables the user to perform this drinking task. In this task, after successful grasping of the bottle, the next skill that can be activated is dependent on the current position of the robot. When the user moves the object close to the mouth, the next desired action is to drink from it. Thus, the trigger signal will activate the "drinking" skill, in which the robot slightly tilts the bottle towards the user to allow for drinking from a straw. As with the other skills, velocity input is not commanded to the robot during execution of this skill. To bring the bottle back to an upright position and achieve movement control again, another trigger signal has to be evoked by the user. To put the bottle back on the table, the user has to move the robot above the table surface. In this condition, the "release" skill will be associated with the trigger signal. This provides an example of how task-dependent knowledge and a defined skill set can be used to generate automated movement patterns to support task execution.

We have validated this approach in a real application, in which a participant with tetraplegia ("S3") was able to control the assistive hand–arm system through the BrainGate2 Neural Interface System.[3] Via this interface, *intended* upper limb movement is decoded from spiking activity related to imagined arm and hand movement and used to directly control the Cartesian velocity of the robotic arm. The participant had free control of the robot's velocity in 2D or 3D space within the predefined spatial limits of the robotic system. A simultaneously decoded binary activation signal related to grasp intention of the participant serves as a trigger for the assistive skills. The functionality of these skills is illustrated in Extension 3.

4. Neural interface system

The investigational BrainGate2 Neural Interface System (Hochberg et al., 2006, 2012) consists of a 96-channel MEA (see Figure 8) that was implanted in the motor arm/hand area contralateral to the dominant hand. The implanted array provided neural recordings with good spatial and temporal resolution for many years after implantation (Simeral et al., 2011) suggesting that the BrainGate2 Neural Interface System could provide a viable easy-to-use control interface. Raw neural signals were filtered using an analog bandpass filter (0.3–7500 Hz) and digitized at a rate of 30 kHz. Over the course of time in which the research sessions were conducted we used two different neural features (preprocessing methods) for decoding continuous velocity commands and a trigger state:

- Neural spike rate;
- Multiunit activity (MUA).

In order to calculate neural spike rate, single units were identified from the raw neural signals by first detecting spikes exceeding a baseline threshold value on each single channel of the electrode array. Detected spikes were then classified into single units according to their waveform shapes. This waveform matching was realized via multiple thresholds, which were manually defined for each unit from visual inspection of the signals. Figure 9 shows an example of multiple neural waveforms recorded on a single channel and thresholds corresponding to a single neuronal unit. The sessions with participant S3 were conducted after five years of implantation and typically 8 to 15 single units were identified with mean spike rates ranging from 1 to 30 Hz.

In contrast to the spike rate method, MUA preprocessing applied an initial baseline threshold criterion to the recorded signal on each channel of the MEA but did not further decompose the recorded activity into single units. Thus,

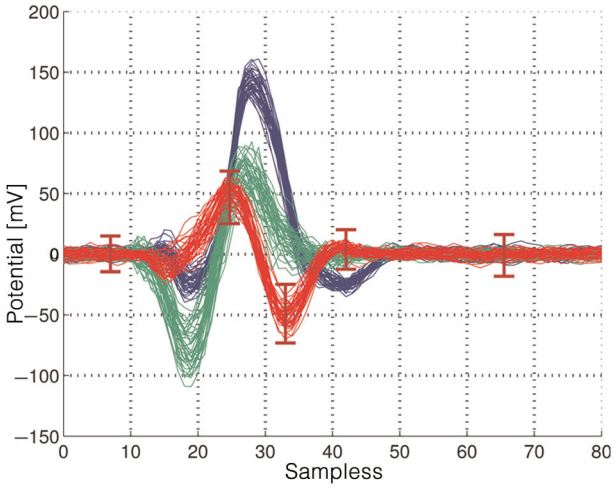

Fig. 9. Artificial example of spike sorting thresholds. The dark red thresholds separate the light red spikes from other spikes recorded on the same channel.

MUA firing rate on each channel generally represents the activity of many cells in the close vicinity of the electrode. Typically, the signal is filtered with a highpass Butterworth filter with $n = 2$ and $\omega_c = 1000$ Hz.

Neural spike rate and MUA rate are simple and fast to calculate and both can be related to movement direction and velocity using a cosine model (Georgopoulos et al., 1982; Moran and Schwartz, 1999). For each neuronal channel acquired from the electrode array, a cosine model is parameterized to estimate direction-dependent velocity. Independent of the feature extraction method, a Kalman filter is used to join the output of the single models $\boldsymbol{z}$ to a global state vector $\boldsymbol{\xi}_{t+1}$ using the previous state vector $\boldsymbol{\xi}_t$ as a prior:

$$\boldsymbol{\xi}_{t+1} = A\boldsymbol{\xi}_t + \boldsymbol{w} \tag{12}$$

$$\boldsymbol{z}_t = H\boldsymbol{\xi}_t + \boldsymbol{v} \tag{13}$$

As velocity is decoded from the neural signals, the state vector $\boldsymbol{\xi}$ is defined as

$$\boldsymbol{\xi} = \dot{\boldsymbol{x}}_{\text{decoded}} \tag{14}$$

The state transition matrix A, which linearly relates the system state $\boldsymbol{\xi}_{t+1}$ to the previous state $\boldsymbol{\xi}_t$, and the observation matrix H, which linearly relates the neural firing rates $\boldsymbol{z}_t$ to the system state $\boldsymbol{\xi}_t$, are both initialized from a training data set by least squares:

$$A = \underset{A}{\operatorname{argmin}} \sum_{k=1}^{M-1} \left\| \boldsymbol{\xi}_{k+1} - A\boldsymbol{\xi}_k \right\|^2 \tag{15}$$

$$H = \underset{H}{\operatorname{argmin}} \sum_{k=1}^{M} \left\| \boldsymbol{z}_k - H\boldsymbol{\xi}_k \right\|^2 \tag{16}$$

The training data set of size M used for this calculation is acquired in the beginning of each experimental session (see Section 5). The matrices A and H are kept constant within an experimental session. The process noise $\boldsymbol{w}$ and measurement noise $\boldsymbol{v}$ are assumed to be zero mean and normally distributed. Further details on the decoding algorithms can be found in Wu et al. (2002, 2006), Kim et al. (2007, 2008), and Hochberg et al. (2012). In addition to the continuous velocity vector, a binary trigger signal is classified from the neural features using a linear discriminant analysis (LDA) classifier. In our framework, this binary signal is used to activate the assistive skills.

5. Research sessions

To validate the functionality of our framework, a number of research sessions were conducted as part of the BrainGate2 clinical trial with one 58-year-old participant "S3". This participant suffered a brainstem stroke more than 14 years prior to the study, leaving her tetraplegic and anarthric (unable to speak). She was implanted with a MEA in the dominant hand area of motor cortex in November 2005. The details of the research sessions can be found in Hochberg et al. (2012). Here, we summarize the aspects of the research session that are essential to understanding the context for the robotics contributions.

A typical experimental session started with the connection of the electrode array to the decoding system followed by a short system initialization routine. When single unit spike rate was used as a neural feature, the manual spike-sorting procedure was conducted. These steps were performed by a clinical technician. The participant was seated in front of the robot workspace as depicted in Figure 10 (on the right). The experimental part of the session was then conducted, consisting of the three following phases:

1. Open-loop decoder calibration;
2. Closed-loop decoder calibration;
3. Task evaluation.

5.1. Open-loop decoder calibration

Open-loop decoder calibration served as an initialization procedure for the cosine tuning models and the Kalman filter used for velocity decoding, as well as the LDA classifier for the binary trigger signal (see Section 4). During the open-loop phase, the participant observed predefined motions of the robotic system and was asked to imagine making these movements with her own arm. The preprogrammed motion of the robot followed a so-called center-out-and-back pattern. In sessions in which 2D control was desired, the pattern consisted of four equidistant targets located in a plane above the table surface. The robot started at a fifth target, the so-called home target, which was located in the center of the other target locations. From here, the robot moved to one target randomly selected from the peripheral targets with a uniform probability

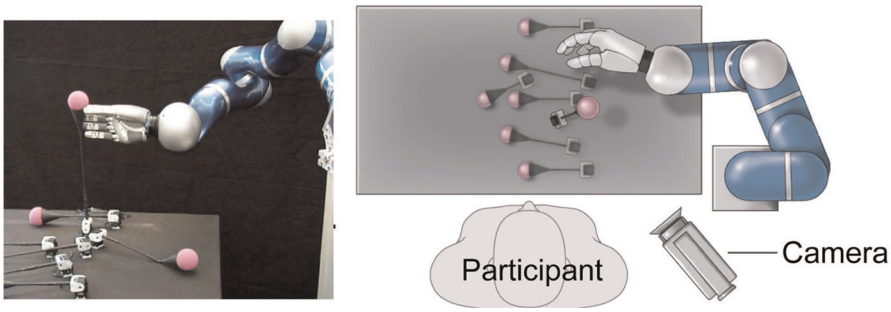

Fig. 10. Automated target placement and robotic system (left). Top view of the experimental setup with the participant sitting in front of the robot workspace (right). Reproduced with kind permission from Nature Publishing Group (Hochberg et al., 2012).

distribution. After reaching the peripheral target, the robot moved back to the home target. Upon reaching the center position, the hand performed a grasp motion during which the participant was asked to imagine a firm grasp with her own hand.

In the case of a 3D session, one additional target was added to the open-loop calibration, which was located above the original center target, so that the targets span a 3D space. The overall training algorithm was as follows.

```
A=[ T1; T2;...; Tn] ; %list of n targets
Center=[ Tc] ; %center target
Ap=randperm(A); %target permutation
for(i=1 to n)
      move_to (Ap(i));
      wait (PAUSE_TIME);
      move_to (Center);
      if (grasp_training)
            close_hand ();
            wait (GRASP_TIME);
            open_hand ();
      end
      wait (PAUSE_TIME);
end
```

To promote accuracy in the imagined reaching movements and to promote consistent movement onset times, each upcoming trajectory of the robot was cued by a foam ball, which was placed at the target location using a motorized target placement system (see Figure 10). Additionally, an auditory cue was provided when the robot started to move. The duration of the open-loop calibration usually did not exceed five minutes. An initial parameterization of the decoding algorithm (Wu et al., 2006) was calculated from neural activity recorded during this open-loop task (see also Hochberg et al., 2012).

5.2. Closed-loop decoder calibration

A closed-loop calibration phase was used to refine the decoder parameterization that had been acquired in the open-loop phase (Jarosiewicz et al., 2013). In this phase the participant attempted to move the robotic hand sequentially to targets presented in the workspace. In the course of this research, the closed-loop training paradigm employed two different target configurations. In addition to the center–out configuration described above, a second configuration presented a home target located on the right side of the robot workspace with six targets on the circumference of a quarter sphere around the home target. In both configurations targets were positioned such that they were equidistant to their respective home target. Similar to the open-loop training, the robot started at the home target position and the participant was asked to move to one, randomly selected, peripheral target. After acquiring the target, or after a 20 s time limit, the robot automatically repositioned at the exact target location and the participant was asked to move it back to the home target.

To adjust for inaccuracies in the initial decoder estimates of intended movements, erroneous motion (motion perpendicular to the instantaneous target direction) was attenuated by calculating an error attenuated desired velocity $\dot{\boldsymbol{x}}_{\text{ea}}$ from the decoded velocity vector $\dot{\boldsymbol{x}}_{\text{decoded}}$ and its projection $\boldsymbol{r}'$ on the target direction vector $\boldsymbol{r}$:

$$\dot{\boldsymbol{x}}_{\text{ea}} = \boldsymbol{r}' + \alpha\,(\dot{\boldsymbol{x}}_{\text{decoded}} - \boldsymbol{r}') \tag{17}$$

$$\boldsymbol{r}' = \frac{\dot{\boldsymbol{x}}_{\text{decoded}} \cdot \boldsymbol{r}}{\boldsymbol{r} \cdot \boldsymbol{r}}\,\boldsymbol{r} \tag{18}$$

The parameter $\alpha \in [0, 1]$ was used to scale the amount of attenuation and was gradually increased from 0 to 1 during the closed-loop training. The parameterization of the decoder was then updated iteratively during the closed-loop training procedure. After approximately 30 minutes of this training, full Cartesian control over the semi-autonomous robotic system in combination with skill triggering was granted to the user.

5.3. Task evaluation

In the task evaluation phase a predefined standard assessment task was performed multiple times in order to calculate success rates and other measures of performance. During this phase, the parameterization of the decoder

remained unchanged. These tasks had been performed with the robotic framework:

- 2D/3D positioning;
- 2D/3D reaching;
- 2D/3D reaching and grasping;
- 2D glass pick-and-place;
- 2D drinking.

Positioning and reaching: In the positioning task, the tip of the thumb of the robotic hand had to be placed within 4 cm of the target in 2D or 3D space. Auditory feedback was used to notify the participant that the target area was obtained. In the reaching task, the participant additionally had to evoke a trigger signal when the tip of the thumb was within the target radius. In these two tasks the target areas were visualized with the target placement system built from foam balls mounted on flexible wooden rods. However, to successfully complete a task it was not necessary to interact with the physical target as long as the target area was reached.

Reaching and grasping: Physical interaction with the target was required in the grasping task. Here, the goal was to squeeze the foam ball located at the target position. To achieve this, the participant had to move the robot hand such that the ball was located between the thumb and index finger of the hand and evoke a grasp trigger. In contrast to the first two tasks, the absolute position of the target was not evaluated, as the foam ball could be pushed within the constraints of its mounting. Instead, the success of the trial was calculated from an appropriate interpretation of the joint torques of the robotic hand.

Grasping and drinking: Besides these research assessment tasks, which allowed us to quantify the control performance with repeated, stereotyped movements, the participant also used our assistive framework to demonstrate the performance of potentially useful manipulation tasks. One was to pick up a glass from a table and place it at a defined target location. In the other, the participant was asked to pick up a filled bottle, move it close to her mouth, drink from a straw, and put the bottle back on the table. In these "real-world" tasks, the assistive skills of our framework, as described in Section 3.2, could be employed to support task execution and thereby reduce the control effort for the user.

6. Results

In this section, we present the results from experimental sessions in which S3 controlled the robotic hand–arm system in combination with our decision-and-control architecture integrated with the BrainGate2 Neural Interface System. Results from the more synthetic ball-grasping assessment task are discussed in more detail by Hochberg et al. (2012), who focus mainly on the neural control aspects of this research. Here, we give only a short presentation of that specific task for the sake of completeness. The benefits of our assistive decision-and-control architecture are more evident within the "real-world" tasks of pick-and-place or drinking.

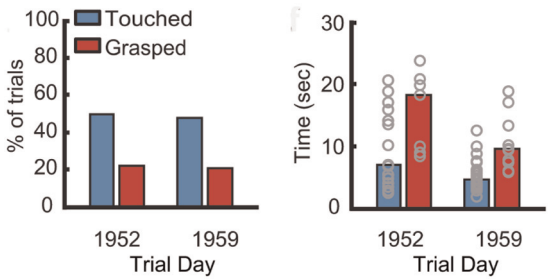

Fig. 11. Results for 3D grasp task from two trial days (touched = target foam ball has been touched with the hand; grasped = target has been physically squeezed). Left: success rate; right: time to target for successful trials. Reproduced with kind permission from Nature Publishing Group (Hochberg et al., 2012).

6.1. 3D grasp task

To evaluate and quantify the general performance of the BrainGate2 Neural Interface System with respect to 3D robot control, we asked the participant to reach and grasp a target foam ball in 3D space. Figure 11 depicts the success rates for two experimental sessions with 32 and 48 trials, respectively, in which S3 performed this task. These sessions used MUA as the neural feature for decoding. In addition to detecting grasp state from the torque sensing in the fingers, we also determined when the fingers established contact with the target ball through visual inspection of video recordings. Though the participant was able to move the robotic arm to contact the target ball ("touched") in 50% of the trials, she was only able to attain a full grasp in 20% of the trials. We expect this to be the result of the target ball being large compared to the maximum aperture of the hand (6 cm ball radius vs 8 cm aperture), making the task very hard. While this result shows that BrainGate-enabled 3D control of a robotic system is possible, the participant did not gain practical benefit from the assistive skills provided by our framework in this particular task. Of course, the soft-robotics features provide safety in any scenario. However, the actual assistive skills of our framework were used in this task only for evaluating the success of the task and did not improve or even affect the usability of the robotic system in this particular setting. Footage of the 3D grasp task can be found in Extension 4.

6.2. 2D glass pick-and-place

In order to validate the supportive capabilities of the assistive planner in combination with true torque-based impedance control, we examined the performance of the system

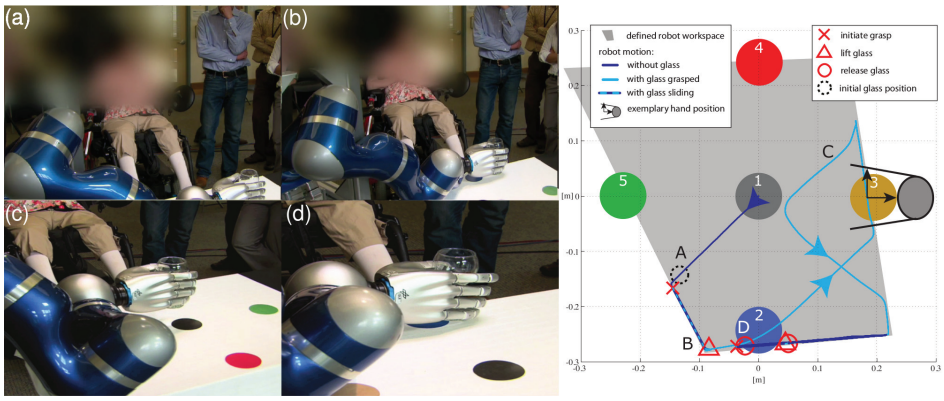

Fig. 12. (Left) Photographs from a glass pick-and-place session: (a) moving to the glass, (b) grasp the glass, (c) move with glass, (d) put down. Right: reconstructed tool center point (TCP) motion from top view. Capital letters mark the locations of the corresponding photographs of panels (a)–(d) along the trajectory.

in a realistic pick-and-place scenario. Figure 12 depicts excerpts from a glass pick-and-place task participant S3 performed in one of the experimental sessions at an early stage of the study. Footage of this task is available in Extension 2. During this session, we used the spike-sorting technique described in Section 4 as a neural feature for the decoding. The colored circles (1–5) mark different targets to which the participant was instructed to command the robot during this session. The quadrangular gray-shaded area shows the desired workspace of the robot with respect to the robot's tool center point (TCP). As the TCP was located in the middle of the robotic hand, it was still possible to move a portion of the hand above a target which was visually mainly outside the TCP workspace (as depicted at target 3). In this specific task, the hand of the robot originated at target 1 and the participant was instructed to move towards the randomly placed glass, grasp it, and bring it back to target 1. It is observed that after picking up the glass the participant moved with a short detour to the desired target but then redirected to target 2. Clearly, the virtual workspace boundaries supported the recovery of the task in this particular setup, because several targets were located at the border of the workspace, thereby artificially decreasing the DoF required for the task. Therefore, targets were not positioned on workspace boundaries after this initial pick-and-place test.

6.3. 2D drinking task

In order to explore how a set of available assistive skills can help in a more sophisticated "real-world" task of controlling a robotic system via a neural interface system to achieve activities of daily living, we extended the pick-and-place task to give the participant the capability to independently drink from a straw. The task was to pick up a bottle of coffee, move it to her mouth, drink from it through a straw, and finally place the bottle back on the table. The dimensions of control were restricted to two translational DoF parallel to the tabletop plane and the simultaneously decoded activation signal was interpreted by the assistive planner as depicted in Figure 7 (Section 3.2). The bottle to be grasped had a diameter of 7.2 cm, which corresponds to 90% of the robotic hand's aperture. Therefore, grasping the bottle required very precise alignment of the hand with the bottle. The decoded activation signal was used to activate the grasp skill upon which grip stability was validated and ensured by the hand controller (see Section 3.1).

After conducting the open- and closed-loop decoder calibration phases, we acquainted the participant with the task layout for approximately 14 minutes. During this period, we demonstrated the state-dependent functionality of the activation signal to the participant. Once the participant was familiar and comfortable with the task setup, we ran a total of six trials of the drinking task within a total time frame of eight minutes. In four of these attempts the participant successfully grasped the bottle, brought it to her mouth, drank coffee from it through a straw, and replaced the bottle on the table. Figure 14 shows four stills taken during the first successful drinking attempt. This was the first time that the participant had been able to independently drink for over 14 years. The two unsuccessful attempts (numbers 2 and 5 in the sequence) were aborted by intervention of the investigators to prevent the robot from pushing the bottle off the table while the participant was trying to align the aperture of the hand precisely with the bottle. Footage of the drinking task is available in Extension 3.

Figure 13 illustrates the phases of this drinking task and the workflow of the assistive skills for the successful trials 1, 3, 4, and 6. The top of each panel depicts the different assistive skills that were activated by the participant. The gray tinted area, containing star and circular shapes, shows whether the transitions between activities were triggered by user command or by automatic decision-making based on the sensor readings from the robotic system. For the first and second successful trials the assistive activities are

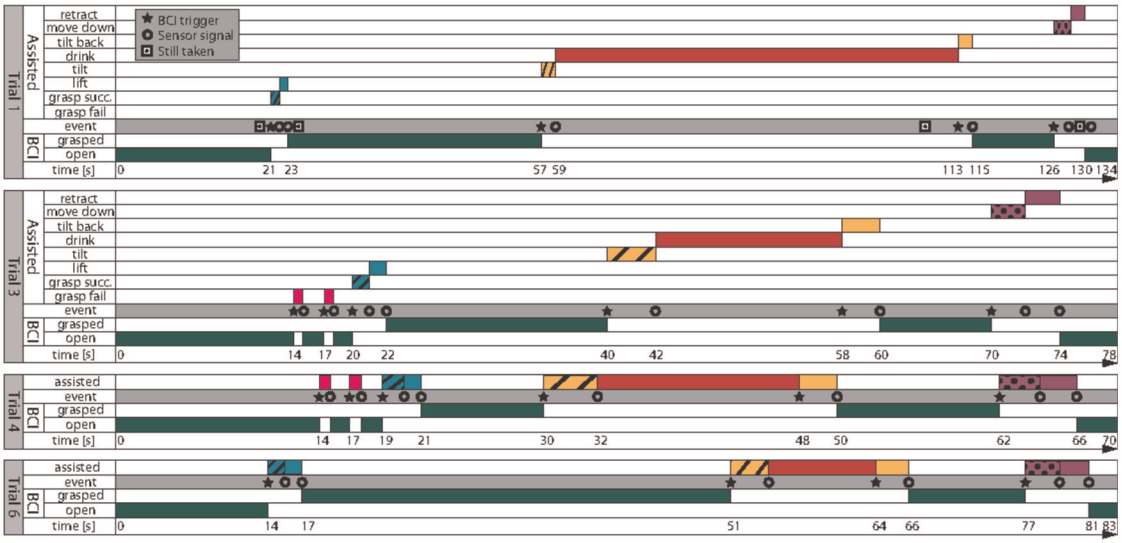

Fig. 13. Four panels of time-line analysis for the successful drinking trials. The top two panels depict in detail the contingent of assistance given by the framework for the first and second successful trials. Each row in the top of a panel illustrates the periods during which the assistive skills were executed. The symbols in the gray tinted row show whether a transition had been evoked by the user (star shape) or by sensory information (circular shape). Additionally the top panel includes the timings of the stills depicted in Figure 14. The bottom of each panel shows the periods during which the user was in free velocity control, either with or without having the object grasped. The bottom two panels depict the third and fourth successful trials, respectively, in a collapsed form. Here, the phases of the assistive skills and their duration can be identified from the color-coding. It is evident that the assistive skills affect a large part of each trial. Note that the time axis is different between the panels.

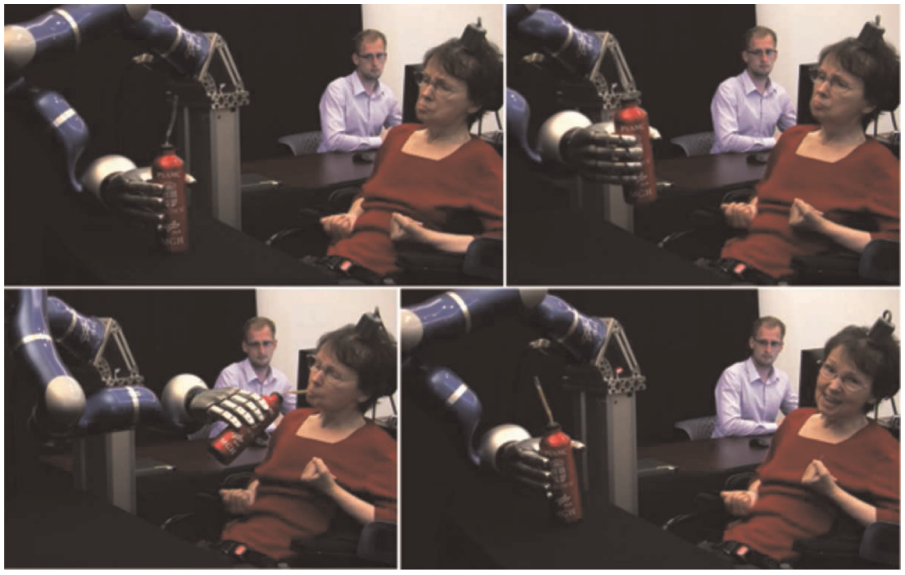

Fig. 14. Screenshots from video footage showing S3 during the first successful completion of the drinking task. See Figure 13 for the related times within task execution for the single pictures. Reproduced with kind permission from Nature Publishing Group (Hochberg et al., 2012).

arranged individually, to illustrate their sequence within the task. In the panel of the first trial, the times are also marked at which the still images of Figure 14 were taken. In the bottom of each panel, the phases in which the participant was in control, either with or without holding the bottle, are illustrated. The timestamps at which skills were activated are marked in the very bottom of a panel. From the first panel, it can be seen that the participant took 21 s to successfully grasp the bottle. After 57 s, the participant activated the drinking skill and the robot tilted the bottle to bring the straw close to the participant's mouth. The participant took about 50 s to drink in this first trial, and during this period the position of the robot remained unchanged. When finished with drinking, the participant commanded

the robot to tilt the bottle back and moved the robot above the table surface in order to activate the release skill and thereby finish the task after a total of approximately 130 s.

In the beginning of the second successful trial, the participant executed two grasp commands, which were detected as unsuccessful (object not stably grasped), but after 20 s, the third grasp activation lead to a successful grasp and she completed the task within a total of 80 s. Similar times can be observed for trials numbers 4 and 6, which were also completed successfully. Comparing the phases of the task in which the participant is in full control of the robotic system with those in which assistive skills were executed, it is notable that the assistive skills comprise a large portion of the task (47%, 35%, 40%, and 25% of the time for the four successful tasks, respectively), thereby actively helping the participant to perform the task.

7. Application using sEMG

While the experimental sessions conducted within the BrainGate2 clinical trial represent the first scenario in which our assistive framework was used by an individual with tetraplegia, another goal of our DLR research team is to develop a surface electromyography (sEMG)-based interface through which people with movement disability could control a robotic hand–arm system using our framework. This approach aims to harness the residual EMG signal stemming from arm muscles that can still be voluntarily activated but do not suffice to move the limbs. In a pilot experiment (see Figure 15 and Vogel et al., 2013), using a full dynamics simulation of the DLR-LWR III visualized on a 3D monitor, we explored the potential of this approach with two 45-year old individuals with SMA type IIa. This disease leads to degeneration of motor neurons in the spinal cord, such that the corresponding muscle fibers can no longer be activated and ultimately degenerate. In effect, after many years of progressive disease no voluntary limb movement is possible. We equipped the participants with six or seven sEMG electrodes (see Figure 16), respectively. To gather ground truth data for system training, we asked each participant to attempt to make movements cued by a visual stimuli paradigm similar to the BrainGate2 sessions.

After training a neural network on these data, the participants were asked to perform a number of reaches in a virtual reality environment by commanding the simulated robotic hand–arm system on a 3D monitor, using their remaining EMG activation. In this setup, a target was presented at a point randomly chosen in the 3D workspace but 30 cm away from the current robot position. The task was to reach for the target location, meaning to place the simulated robot hand within 3 cm of the virtual target and to execute a grasp trigger at that position. In this preliminary set of experiments, the participants performed two sequences of 30 reaching movements each. As this task is very similar to the ball-grasping tasks conducted within the BrainGate2 trial, the relevant capabilities of our system, which become apparent only in more complex tasks, have not been fully tested with this interface. In future work we will combine the newly developed sEMG-based interface with the assistive framework and have the participants perform more sophisticated and more "real-world" related tasks, with the support of our decision-and-control architecture and a real robotics system.

Fig. 15. Participant with SMA controlling a simulated robotic hand–arm system in simulation on a 3D monitor.

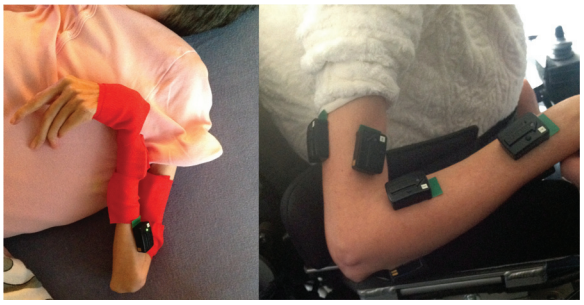

Fig. 16. Participants with SMA equipped with sEMG electrodes.

8. Conclusion

We presented an assistive decision-and-control architecture that is designed to enhance the usability of assistive robotic hand–arm systems via semi-autonomous, soft-robotics-enabled capabilities. The goal of this research is to considerably reduce the control effort when people with physical disabilities control a robotic system via a HMI. To validate the functionality of our architecture, we demonstrated that a participant with tetraplegia was able to safely control an assistive robotic system through an intracortical neural interface. In particular, the participant was able to serve herself a drink of coffee, for the first time since she suffered a brain stem stroke 14 years prior to these experiments. In this task, the participant had to control the robot in 2D continuous space and activate assistive skills via a binary trigger signal. Analysis of the ratio of assisted versus unassisted control shows that our architecture has an important effect on the behavior of the robot and thereby helps the user to succeed in the task. This demonstrated

drinking task suggests that our decision-and-control framework, together with a truly torque-controlled robot, is potentially well-suited for scenarios where safe and intuitive physical human–robot interaction is required. Our architecture uses the following soft-robotics features:

- **Low-weight design**: Provides intrinsically safer and sensitive robot behavior in human–robot interaction scenarios.
- **Cartesian impedance control**: Task space impedance control provides an optimal interface for low-dimensional (e.g. 2D translation + binary activation signal) continuously decoded HMI data. The actively well-damped robotic system additionally smooths the low-sampled noisy input data.
- **Joint impedance control in the hand**: Allows for stable grasping and detection of objects.
- **Collision detection and reaction**: Enables safe physical human–robot interaction and thereby provides safety to the user. Furthermore, proprioceptive contact interpretation allows for simplifications in object handling and realization of the assistive skills, for example when putting an object down on the table.
- **Virtual workspace boundaries with auditory feedback**: Provide safety to the user and the environment, as they prevent the robotic system from moving into sensitive or inaccessible areas of the workspace.

Furthermore, the implemented assistive planner provides a flexible interface to embed task-dependent knowledge within the robot control and enables the execution of complex tasks with a limited control interface. Abstraction of robotic concepts like grasp stability to easy-to-understand robotic behavior increases the safety and simplifies the usability of the robotic system.

In future work, we plan to extend the available skill set to allow for more sophisticated scenarios that require a more complex sequence of actions. Therefore, it will be desirable to further increase the autonomy level of the framework, to overcome the limitations of the reduced dimensionality of the interface, while at the same time maintaining the flexibility continuous control provides. Integration of a vision system can allow for more adaptability within skill execution, or allow for more elaborate assistive skills, such as supporting alignment of the hand with the object to be grasped. Another line of research we plan to pursue is within the feedback provider of our framework. While the auditory feedback already helps the user to interpret the behavior of the robot (for example, when workspace limits are reached) a true haptic feedback might be a helpful extension for people with physical disabilities but intact sensation. Also, additional visual feedback provided via an external display can supply the user with information about the state of the system and thereby enhance the usability.

Acknowledgements

We would like to thank Tobias Ende for his illustration and Sven Parusel for his support.

Funding

This work has been partially funded by the European Commission's Sixth Framework Programme as part of the project SAPHARI (grant number 287513) and by the European Commission's Seventh Framework Programme as part of the Project THE Hand Embodied (grant number 248257).

Notes

1. Please note that the LWR-III actually has to be described by a flexible joint model for the low-level torque control loop (Albu-Schäffer et al., 2007b). However, for the sake of clarity we will simplify the control design to the rigid-body case, as the more complex treatment of joint elasticities does not add any insights at this point.
2. Please note that we omit the discussion of virtual forces along the robot structure for the sake of clarity. However, *Beasty* is also equipped with workspace boundaries acting along the entire robot structure.
3. CAUTION: investigational device. Limited by federal law to investigational use.

References

Albu-Schäffer A, Haddadin S, Ott C, et al. (2007a) The DLR lightweight robot: Design and control concepts for robots in human environments. *Industrial Robot: An International Journal* 34: 376–385.

Albu-Schäffer A, Ott C and Hirzinger G (2007b) A unified passivity-based control framework for position, torque and impedance control of flexible joint robots. *The International Journal of Robotics Research* 26(1): 23–39.

Andreasen D, Alien S and Backus D (2005) Exoskeleton with EMG based active assistance for rehabilitation. In: *9th international conference on rehabilitation robotics*, pp. 333–336.

Arbib M, Metta G and van der Smagt P (2008) Neurorobotics: From vision to action. In: Siciliano B and Khatib O (eds) *Handbook of Robotics*. Berlin: Springer-Verlag, pp. 1453–1480.

Artemiadis PK and Kyriakopoulos KJ (2010) EMG-based control of a robot arm using low-dimensional embeddings. *IEEE Transactions on Robotics* 26(2): 393–398.

Bäuml B and Hirzinger G (2008) When hard realtime matters: Software for complex mechatronic systems. *Robotics and Autonomous Systems* 56(1): 5–13.

Bitzer S and van der Smagt P (2006) Learning EMG control of a robotic hand: Towards active prostheses. In: *Proceedings of the 2006 IEEE international conference on robotics and automation*, pp. 2819–2823.

Bohren J, Papazov C, Burschka D, et al. (2013) A pilot study in vision-based augmented telemanipulation for remote assembly over high-latency networks. In: *2013 IEEE international conference on robotics and automation (ICRA)*, pp. 3631–3638.

Bradberry T, Gentili R and Contreras-Vidal J (2010) Reconstructing three-dimensional hand movements from noninvasive

electroencephalographic signals. *The Journal of Neuroscience* 30(9): 3432–3437.

Castellini C and van der Smagt P (2009) Surface EMG in advanced hand prosthetics. *Biological Cybernetics* 100(1): 35–47.

Collinger J, Wodlinger B, Downey J, et al. (2013) High-performance neuroprosthetic control by an individual with tetraplegia. *Lancet* 381(9866): 557–564.

Cui X, Bray S and Reiss A (2010) Speeded near infrared spectroscopy (NIRS) response detection. *PLoS ONE* 5(11): e15474.

Duchowski AT, Pelfrey B, House DH, et al. (2011) Measuring gaze depth with an eye tracker during stereoscopic display. In: *Proceedings of the ACM SIGGRAPH symposium on applied perception in graphics and visualization*, pp. 15–22.

Felzer T and Freisleben B (2002) HaWCoS: The hands-free wheelchair control system. In: *Proceedings of the fifth international ACM conference on assistive technologies*, pp. 127–134.

Georgopoulos A, Kalaska J, Caminiti R, et al. (1982) On the relations between the direction of two-dimensional arm movements and cell discharge in primate motor cortex. *The Journal of Neuroscience* 2(11): 1527–1537.

Gerven M, Farquhar J, Schaefer R, et al. (2009) The brain–computer interface cycle. *Journal of Neural Engineering* 6: 041001.

Giummarra M, Gibson S, Georgiou-Karistianis N, et al. (2008) Mechanisms underlying embodiment, disembodiment and loss of embodiment. *Neuroscience & Biobehavioral Reviews* 32(1): 143–160.

Haddadin S, Albu-Schaffer A, De Luca A, et al. (2008) Collision detection and reaction: A contribution to safe physical human-robot interaction. In: *IEEE/RSJ international conference on intelligent robots and systems*, pp. 3356–3363.

Haddadin S, Parusel S, Belder R, et al. (2010) Holistic design and analysis for the human-friendly robotic co-worker. In: *2010 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pp. 4735–4742.

Haddadin S, Suppa M, Fuchs S, et al. (2011) Towards the robotic co-worker. In: *International symposium on robotics research*, pp. 261–282.

Hochberg L, Bacher D, Jarosiewicz B, et al. (2012) Reach and grasp by people with tetraplegia using a neurally controlled robotic arm. *Nature* 485(7398): 372–375.

Hochberg L, Serruya M, Friehs G, et al. (2006) Neuronal ensemble control of prosthetic devices by a human with tetraplegia. *Nature* 442(7099): 164–171.

Hornyak T (2006) Thinking of child's play. *Scientific American* 295(3): 30–30.

Ishikawa A, Shimizu K and Birbaumer N (2007) Temporal classification of multichannel near-infrared spectroscopy signals of motor imagery for developing a brain–computer interface. *NeuroImage* 34: 1416–1427.

Jacob RJ and Karn KS (2003) Eye tracking in human-computer interaction and usability research: Ready to deliver the promises. *Mind* 2(3): 4.

Jarosiewicz B, Masse NY, Bacher D, et al. (2013) Advantages of closed-loop calibration in intracortical brain–computer interfaces for people with tetraplegia. *Journal of Neural Engineering* 10(4): 046012.

Kim S, Simeral J, Hochberg L, et al. (2007) Multi-state decoding of point-and-click control signals from motor cortical activity in a human with tetraplegia. In: *3rd international IEEE/EMBS conference on neural engineering*, pp. 486–489.

Kim S, Simeral J, Hochberg L, et al. (2008) Neural control of computer cursor velocity by decoding motor cortical spiking activity in humans with tetraplegia. *Journal of Neural Engineering* 5: 455.

Kim T, Zimmerman P, Wade M, et al. (2005) The effect of delayed visual feedback on telerobotic surgery. *Surgical Endoscopy* 19(5): 683–686.

Krishnamurthy G and Ghovanloo M (2006) Tongue drive: A tongue operated magnetic sensor based wireless assistive technology for people with severe disabilities. In: *2006 IEEE international symposium on circuits and systems*, p. 4.

Liu H, Wu K, Meusel P, et al. (2008) Multisensory five-finger dexterous hand: The DLR/HIT hand II. In: *IEEE/RSJ international conference on intelligent robots and systems*, pp. 3692–3697.

McFarland DJ, Sarnacki WA and Wolpaw JR (2010) Electroencephalographic (EEG) control of three-dimensional movement. *Journal of Neural Engineering* 7(3): 036007.

Maheu V, Archambault P, Frappier J, et al. (2011) Evaluation of the JACO robotic arm: Clinico-economic study for powered wheelchair users with upper-extremity disabilities. In: *2011 IEEE international conference on rehabilitation robotics (ICORR)*, pp. 1–5.

Mellinger J, Schalk G, Braun C, et al. (2007) An MEG-based brain-computer interface (BCI). *Neuroimage* 36(3): 581.

Miezin F, Maccotta L, Ollinger J, et al. (2000) Characterizing the hemodynamic response: Effects of presentation rate, sampling procedure, and the possibility of ordering brain activity based on relative timing. *Neuroimage* 11(6): 735–759.

Milekovic T, Fischer J, Pistohl T, et al. (2012) An online brain–machine interface using decoding of movement direction from the human electrocorticogram. *Journal of Neural Engineering* 9(4): 046003.

Miller K, DenNijs M, Shenoy P, et al. (2007) Real-time functional brain mapping using electrocorticography. *Neuroimage* 37(2): 504–507.

Moran D and Schwartz A (1999) Motor cortical representation of speed and direction during reaching. *Journal of Neurophysiology* 82(5): 2676–2692.

Mulas M, Folgheraiter M and Gini G (2005) An EMG-controlled exoskeleton for hand rehabilitation. In: *9th international conference on rehabilitation robotics*, pp. 371–374.

Nguyen V-D(1986) *The synthesis of stable force-closure grasps*. Master's Thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, MA.

Nijboer F, Sellers E, Mellinger J, et al. (2008) A P300-based brain-computer interface for people with amyotrophic lateral sclerosis. *Clinical Neurophysiology* 119(8): 1909–1916.

Parusel S, Haddadin S and Albu-Schaffer A (2011) Modular state-based behavior control for safe human-robot interaction: A lightweight control architecture for a lightweight robot. In: *2011 IEEE international conference on robotics and automation (ICRA)*, pp. 4298–4305.

Pascual J, Velasco-Alvarez F, Muller KR, et al. (2012) First study towards linear control of an upper-limb neuroprosthesis with an EEG-based brain-computer interface. In: *2012 annual international conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 3269–3273.

Raspopovic S, Capogrosso M, Petrini FM, et al. (2014) Restoring natural sensory feedback in real-time bidirectional hand prostheses. *Science Translational Medicine* 6(222): 222ra19.

Robles-De-La-Torre G (2006) The importance of the sense of touch in virtual and real environments. *IEEE Multimedia* 13(3): 24–30.

Römer G, Stuyt H, Peters G, et al. (2004) 14 processes for obtaining a "Manus" (ARM) robot within the Netherlands. In: Bien ZZ and Stefanov D (eds) *Advances in Rehabilitation Robotics* (Lecture Notes in Control and Information Science, vol. 306). Berlin: Springer, pp. 221–230.

Rosen J, Brand M, Fuchs MB, et al. (2001) A myosignal-based powered exoskeleton system. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans* 31(3): 210–222.

Schalk G, Miller K, Anderson N, et al. (2008) Two-dimensional movement control using electrocorticographic signals in humans. *Journal of Neural Engineering* 5: 75.

Simeral J, Kim S, Black M, et al. (2011) Neural control of cursor trajectory and click by a human with tetraplegia 1000 days after implant of an intracortical microelectrode array. *Journal of Neural Engineering* 8: 025027.

Smith RJ, Tenore F, Huberdeau D, et al. (2008) Continuous decoding of finger position from surface EMG signals for the control of powered prostheses. In: *30th annual international conference of the IEEE Engineering in Medicine and Biology Society*, pp. 197–200.

Thomson EE, Carra R and Nicolelis MA (2013) Perceiving invisible light through a somatosensory cortical prosthesis. *Nature Communications* 4: 1482.

Vertegaal R (2008) A Fitts Law comparison of eye tracking and manual input in the selection of visual targets. In: *Proceedings of the 10th international conference on multimodal interfaces*, pp. 241–248.

Vogel J, Bayer J and van der Smagt P (2013) Continuous robot control using surface electromyography of atrophic muscles. In: *2013 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pp. 845–850.

Vogel J, Castellini C and van der Smagt P (2011) EMG-based teleoperation and manipulation with the DLR LWR-III. In: *2011 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pp. 672–678.

Wang W, Collinger JL, Degenhart AD, et al. (2013) An electrocorticographic brain interface in an individual with tetraplegia. *PLoS ONE* 8(2): e55344.

Weiskopf N, Mathiak K, Bock S, et al. (2004) Principles of a brain-computer interface (BCI) based on real-time functional magnetic resonance imaging (fMRI). *IEEE Transactions on Biomedical Engineering* 51(6): 966–970.

Wu W, Black M, Gao Y, et al. (2002) Inferring hand motion from multi-cell recordings in motor cortex using a Kalman filter. In: *SAB'02-workshop on motor control in humans and robots: On the interplay of real brains and artificial devices*, pp. 66–73.

Wu W, Gao Y, Bienenstock E, et al. (2006) Bayesian population decoding of motor cortical activity using a Kalman filter. *Neural Computation* 18(1): 80–118.

Appendix: Index to Multimedia Extensions

Archives of IJRR multimedia extensions published prior to 2014 can be found at http://www.ijrr.org, after 2014 all videos are available on the IJRR YouTube channel at http://www.youtube.com/user/ijrrmultimedia

Table of Multimedia Extensions

Extension	Media Type	Description
1	Video	Demonstration of the "grasp" and "release" skill in a 1D pick-and-place task conducted within the BrainGate2 clinical trial.
2	Video	Demonstration of the "grasp" and "release" skill in a 2D pick-and-place task conducted within the BrainGate2 clinical trial.
3	Video	Drinking demonstration showing the full functionality of the assistive skills within the BrainGate2 clinical trial.
4	Video	3D reach and grasp conducted within the BrainGate2 clinical trial.