A Non-Linear Body Machine Interface for Controlling Assistive Robotic Arms

Fabio Rizzoglio, Marco Giordano, Ferdinando A. Mussa-Ivaldi, and Maura Casadio

***Abstract*—*Objective:* Body machine interfaces (BoMIs) enable individuals with paralysis to achieve a greater measure of independence in daily activities by assisting the control of devices such as robotic manipulators. The first BoMIs relied on Principal Component Analysis (PCA) to extract a lower dimensional control space from information in voluntary movement signals. Despite its widespread use, PCA might not be suited for controlling devices with a large number of degrees of freedom, as because of PCs' orthonormality the variance explained by successive components drops sharply after the first. *Methods:* Here, we propose an alternative BoMI based on non-linear autoencoder (AE) networks that mapped arm kinematic signals into joint angles of a 4D virtual robotic manipulator. First, we performed a validation procedure that aimed at selecting an AE structure that would allow to distribute the input variance uniformly across the dimensions of the control space. Then, we assessed the users' proficiency practicing a 3D reaching task by operating the robot with the validated AE. *Results:* All participants managed to acquire an adequate level of skill when operating the 4D robot. Moreover, they retained the performance across two non-consecutive days of training. *Conclusion:* While providing users with a fully continuous control of the robot, the entirely unsupervised nature of our approach makes it ideal for applications in a clinical context since it can be tailored to each user's residual movements. *Significance*: We consider these findings as supporting a future implementation of our interface as an assistive tool for people with motor impairments.**

***Index Terms*—Assistive manipulator, autoencoders, human-machine interface, motor learning.**

I. Introduction

EVERY year, around the world, between 250,000 to 500,000 people suffer a spinal cord injury (SCI). Depending on the level of injury, SCI might lead to the loss of the ability to perform selective movements, coordinate body motions, and ultimately limit the performance of functional activities of daily living (ADLs). Common activities, such as walking, grooming and object manipulation, could be facilitated using assistive devices, including electrically powered wheelchairs and robotic manipulators. However, designing interfaces for controlling such devices can be challenging - especially when dealing with systems with many degrees of freedom (DoFs). For instance, the use of a common interface for controlling a powered wheelchair, such as a joystick, does not take into account the limited arm and hand mobility or coordination in individuals with cervical SCI, which might limit the operation of the device [1]. Alternative controllers, like sip-and-puff, head-mounted switches and tongue-based devices are available, although these provide the user with only a limited set of discrete commands [2], [3], [4].

Here, we consider a class of human-machine interfaces, the body machine interface (BoMI) [5], [6], which exploits the fact that, even after a severe injury, most individuals retain some movement, especially of their head and shoulders, that can be used to control external devices. BoMIs convert high-dimensional body signals (*e.g.*, upper body kinematics, muscle activities) into lower-dimensional commands to operate the device. As a result, BoMIs allow individuals with motor disabilities to overcome some of their impairments. Importantly, BoMIs allow recovering continuous control of external devices, as opposite for example to methods based on the recognition of discrete gestures [7].

BoMIs have been tested in situations involving the control of a computer cursor [8], a powered wheelchair [9], a robotic manipulator [10], [11], and quadcopters [6], [12]. They most often rely on a linear dimensionality reduction (DR) technique such as Principal Component Analysis (PCA) [13]. Linear control methods generate full repertoires of actions by summation of simpler actions. Moreover, linear models have a low computational cost and produce consistent results, independent of the choice of structural parameters (often referred to as "hyperparameters"). Obviously, linear models cannot account for non-linear features of the input dataset, and this leads to overestimating the actual dimensionality of the input signal source [14], potentially affecting the control of external devices. Moreover, despite its widespread use, PCA might not be suited for controlling devices with a large number of DoFs, as because of PCs' orthonormality the variance explained by successive components drops precipitously. Therefore, most signal information is captured by the first few PCs while the residual is considered to be noise. Here, we propose using non-linear autoencoder (AE) networks

Manuscript received 21 May 2022; revised 19 October 2022; accepted 3 January 2023. Date of publication 16 January 2023; date of current version 20 June 2023. This work was supported in part by the NSF under Grants 2054406, 1654929, and 1632259, in part by the NIDILRR under Grant 90REGE0005-01, and in part by the NIBIB under Grant R01-EB024058, and in part by the RAISE funded by NextGeneration EU, PNRR Italy, Innovation Ecosystems Program. *(Corresponding author: Fabio Rizzoglio.)*

Fabio Rizzoglio is with the Northwestern University and the Shirley Ryan Ability Lab, Chicago, IL 60611 USA, and also with the University of Genoa, 16145 Genoa, Italy (e-mail: fabio.rizzoglio@northwestern.edu).

Marco Giordano and Maura Casadio are with the University of Genoa, Italy.

Ferdinando A. Mussa-Ivaldi is with the Northwestern University and the Shirley Ryan Ability Lab, USA.

Digital Object Identifier 10.1109/TBME.2023.3237081

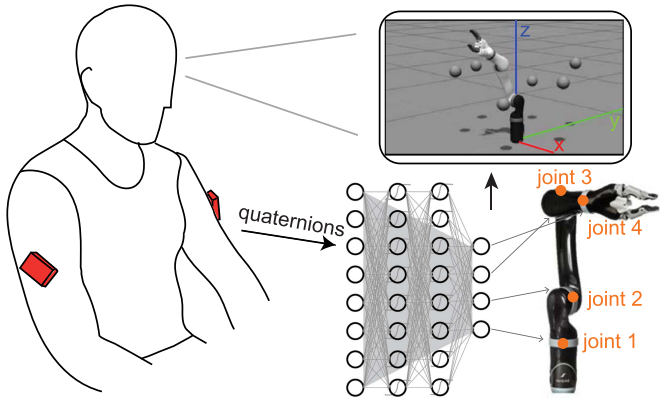

Fig. 1. Setup of the experiment. Quaternions from two IMUs (red boxes) were mapped by the AE encoder to the four joints (orange dots) of the virtual MICO. The participant received visual feedback on a PC monitor.

to overcome these difficulties. AEs are artificial neural networks that compress data into a latent representation by minimizing the reconstruction error. Not only non-linear AEs provide estimates of the dimensionality of their input that are more parsimonious than PCA [15], but their structure can also be customized to distribute variance more uniformly across the latent dimensions. Although there are only a few examples of the use of AEs within a BoMI framework [16], [17], [18], the hypothesis of this study is that the AE architecture and non-linear features offer an efficient platform for operating manipulators with more than two DoFs, such as assistive robotic arms.

Here, we developed a BoMI based on a non-linear AE allowing users to control a 4D virtual robotic manipulator. First, we performed a validation procedure that aimed at selecting an appropriate AE structure that would allow distributing the variance uniformly across the dimensions of the control space. Then, we assessed the users' performance practicing a 3D reaching task by operating the robot with the validated AE. Finally, we characterized the motor strategies employed by the BoMI users over the course of training across two non-consecutive days. A preliminary version of this work has been reported in [19].

II. Methods

A. Experimental Apparatus

The BoMI recorded 8 kinematic signals from the participants' upper arms and transformed them into 4 control signals specifying the configuration of the virtual robotic manipulator. The body signals were generated by two inertial measurement units (IMUs, BN0055 Sensors, *Bosch, Gerlingen-Schillerhfiohe, Germany*) positioned bilaterally on the participants' arms (Fig. 1). The IMU sensors derived orientation in the quaternion format using a sensor fusion algorithm that combined the raw measurements of the embedded accelerometer, gyroscope, and magnetometer. Orientation data from the IMUs were acquired in real-time using a WiPy 2.0 (*Pycom, Guildford*). The software of the WiPy was developed in MicroPython.

The kinematic data were then used to control a virtual robot simulated on Gazebo [20]. Specifically, we simulated the MICO robotic arm (*Kinova Robotics, Canada*), a modular robotic manipulator that is suitable for advanced assistive manipulation research and can easily be installed on powered wheelchairs. The control of the virtual robot was achieved via ROS packages [21].

We used an AE to map the 8-dimensional (8D) vector $\boldsymbol{q}$ of body signals - quaternions from the IMUs - into the 4D vector $\boldsymbol{\theta}$ of virtual-MICO's joint angles. An AE is composed of two main elements: an encoder compressing the input into a lower-dimensional latent representation, or code, followed by a decoder converting the latent representation into the output, with the same dimensionality as the input. In the BoMI, the code layer generates the control signal for the external device. For instance, when controlling a 2D computer cursor, an AE with 2 code units (CUs) allows mapping the first CU to control cursor movements along the x-axis, and the second CU allows to control movements along the y-axis. In this study, the external device controlled via the AE had 4 DoFs. Therefore, the BoMI forward map that transformed the eight signals from the body-space signals into the lower-dimensional (4D) control-space was set to be the encoder subnetwork of an AE with a 4D CU layer. We used the first code of the AE to control the movements of the wrist angle of the manipulator – the most distal joint - and proceeding in order toward the base so that the fourth code controlled the robot's base-joint angle, as illustrated in Fig. 1. Each code was re-scaled to fit the maximum extension of each joint angle (360° for the base-joint angle, 180° for the others).

B. Validation of the AE

The validation procedure aimed at selecting the parameters of a 4D AE that would yield (i) minimum information loss and (ii) uniform distribution of variance across the CUs. This step was performed using a dataset obtained via an unsupervised calibration procedure with the same structure as reported in [18], [22]. A subject (age 25, male) that did not participate in the subsequent study was asked to freely move his arms for 60s. The movements performed were intended to span the range of possible upper body movements without executing uncomfortable or extreme gestures. The calibration procedure shared the same experimental setup considered in this study – two IMUs were placed bilaterally on the participant's arms to record the 8D body signal. The kinematic data recorded during calibration consisted of 3000 samples (60s with 50 Hz sampling frequency). The calibration dataset was split into 80% of training points (2400) and 20% of test points (600). The calibration dataset was randomly shuffled along the time dimension before the split to avoid potential discrepancies between the training and test distributions.

The hyperparameters and architecture of the AE needed to be fine-tuned for designing the most effective forward map for the online control of the MICO. The optimal AE network was chosen among two candidate architectures:

- *under-parametrized AE:* first, we trained an AE with fewer parameters (364) than the number of training samples

(2400). The network was designed with four hidden layers (two in the encoder subnetwork and two in the decoder subnetwork), with eight neurons per layer.

- *over-parametrized AE:* then, we trained an AE with a higher number of parameters (2652), that were comparable to the number of training samples (2400). This network also had four hidden layers, but 30 neurons per layer.

Both architectures were based on a nonlinear activation function, a hyperbolic tangent (*tanh*, [23]), for each hidden layer and a linear function for the code and output layers. The initial AE weights were chosen following the Xavier normal initialization [24], while the initial biases were set to zero.

In order to measure the information preserved by each AE during training, we computed a goodness-of-fit metric, the *Variance Accounted For* (*VAF*), defined as the percentage of variance of the calibration dataset explained by the AE:

$$VAF = \left(1 - \frac{var\left(\boldsymbol{q} - \hat{\boldsymbol{q}}\right)}{var\left(\boldsymbol{q}\right)}\right) * 100 \tag{1}$$

where $\boldsymbol{q}$ is the original calibration dataset and $\hat{\boldsymbol{q}}$ is the dataset reconstructed by the AE. A VAF of 100% indicated that the AE could perfectly reconstruct the variance of the calibration dataset.

Then, we quantified the variance of the ith AE CU ($\boldsymbol{v_i}$) and evaluated how it was distributed across all CUs. We defined as *latent variance* ($\boldsymbol{v_{lat_i}}$) the percentage with respect to the overall variance of all considered CUs:

$$\boldsymbol{v_{lat_i}} = \frac{\boldsymbol{v_i}}{\sum_{i=1}^{4} \boldsymbol{v_i}} * 100 \tag{2}$$

Unlike with PCA, training an AE does not guarantee a convex solution [25]. Therefore, the VAF and latent variance might differ depending on the initial conditions of the AE. While the VAF is typically less prone to depend on the initial conditions, the latent variance can change drastically with different initializations [15]. Therefore, to evaluate the impact of different initializations on the VAF and the latent variance, we trained each type of network with five different sets of initial parameters.

The goal of the validation was to select the AE architecture that would yield a high VAF (*i.e.*, $VAF > 90\%$) concurrently with a uniformly distributed variance across code units (*i.e.*, $v_{lat_i} \sim 25\%$ for each CU). In addition to the various nonlinear AEs, we also applied PCA on the same training dataset and compared its performances with those of the nonlinear AEs.

The results from the tuning procedure were obtained considering the test dataset only and are presented in Section III.A. The final choice leaned towards one of the five under-parametrized networks. Therefore, we assigned the encoder sub-network derived after training the selected AE structure on the calibration dataset as the BoMI forward map for the online control of the MICO. The transformations operated by the encoder to obtain the 4D control vector $\boldsymbol{\theta}$ from the 8D body vector $\boldsymbol{q}$ were:

$$\begin{aligned} \boldsymbol{layer}_1 &= \tanh\left(w_1 * \boldsymbol{q} + b_1\right) \\ \boldsymbol{layer}_2 &= \tanh\left(w_2 * \boldsymbol{layer}_1 + b_2\right) \\ \boldsymbol{\theta} &= w_3 * \boldsymbol{layer}_2 + b_3 \end{aligned} \tag{3}$$

where w_j and b_j, $j = 1:3$ were the weights and biases of the encoder subnetwork, respectively.

C. Participants

For testing the online control of the virtual MICO, we enrolled 12 right-hand dominant, unimpaired participants (5 females, age 24±3 yo). They did not have any evidence or known history of postural, musculoskeletal, or neurological disorders, and exhibited normal joint range of motion and muscle strength. They signed a consent form approved by the Institutional Review Board (DIBRIS UNIGE n. 009/2020). All participants were assigned with the same BoMI forward map, derived from the validation procedure.

D. Experimental Protocol

The experimental protocol consisted of a 3D center-out reaching task. Participants performed the task while comfortably seated. An LCD computer screen, positioned in front of them about 1.5 m away at eye level, displayed the position of the virtual manipulator and the targets. The experiment was divided into two identical sessions performed on different days of the same week. The maximum duration of a single session was set to two hours. Participants were asked to reach six targets distributed on a cylindrical surface, at different heights, with the robot's end effector (EE). To start a reaching movement, participants were asked to comfortably put their hands on their lap and not flex their shoulders. The resulting resting position ($\boldsymbol{q}_0$) was then transformed by the BoMI forward map into the robot initial configuration ($\boldsymbol{\theta}_0$, shown in Fig. 1). A trial was considered successful if the target remained between the fingers of the manipulator's EE for 500 ms.

The protocol included training and test phases, alternating as follows:

- *initial test:* a baseline test of participants' performance in the reaching task.
- *initial training:* first phase of extensive training with the interface.
- *mid test:* a second test of participants' performance.
- *final training:* second phase of extensive training with the interface.
- *final test:* a final test of participants' performance at the end of the training.

If the participant failed to complete the whole protocol within 2 hours, the training phase (either initial or final) would be interrupted before its completion and the final test would still be performed. The details of test and training phases are as follows:

1) Training Phases: In these phases, participants learned how to maneuver the robot. Participants were required to reach six different targets, distributed on a cylindrical surface, with the elliptical base of axes length 0.98 and 0.8 Gazebo units respectively and centered on the base of the manipulator. Targets appeared at three different heights: 0.30, 0.41, 0.51 Gazebo units (from the manipulator base). Position of the targets with respect to the manipulator is shown in Fig. 1. Note that Gazebo units are meant to replicate distances in the real world. Hence, if controlling an actual MICO robot, all the targets position to

reach would be in meters [26]. The six targets were presented twelve times in pseudorandom order, with the condition that each target was not presented again before all six targets had been reached. Both the initial and the final training consisted of $12*6 = 72$ reaching trials, for a total of $72+72 = 144$ training targets. Starting from the resting position, participants were required to reach the peripheral targets with a timeout of 60 seconds. If they failed to complete the trial within 60 s, the target would disappear, and they had to come back to the resting position before the following target was presented.

2) Test Phases: In these phases, we tested participants' ability to transfer the skills acquired during the training phase to conditions where they had to move toward 6 targets that required moving in (i) different directions and/or (ii) different displacement amplitudes (scaling-expansion) with respect to the training targets. The coordinates of the test targets were randomly displaced from their original position during training within a range of 0.15-0.35 m. For each test phase, the six targets were presented three times in pseudorandom order ($6*3 = 18$ targets), for a total of $18+18+18 = 54$ test targets. Starting from the resting position, participants had to reach the presented peripheral target within 20 seconds. If they failed to do so, the target disappeared, and the trial was considered unsuccessful. At the end of each reaching, regardless of it being successful or not, participants were asked to come back to the resting position and get ready for the next target.

E. Outcome Metrics

We computed four metrics to evaluate the performances of the participants during the online test of the BoMI:

- *Reaching Time (RT):* time from the first appearance of a new target to when the target is reached successfully. A reduction of RT indicates improvement in performance.
- *Success rate (SR):* percentage of trials completed within timeout during test blocks.
- *Normalized path length (NPL):* path-length travelled by the robot end effector to reach the target divided by the distance of the target to the starting point. This is an index of straightness of the robot's EE movements.
- *Smoothness index (SI):* number of peaks in the robot end effector speed profile. We considered every peak larger than a threshold set to be 20% of the maximum speed of each trajectory. This is an index of smoothness of the robot's EE movements.

In addition, we evaluated how BoMI users redistributed control authority across right and left arm during the training phases and which control strategies they employed to move the robot efficiently toward the targets.

1) Control Strategies During Training: First, we studied the dimensionality of movements both in terms of IMU and robot joint variance. Among the wide variety of existing dimensionality estimation algorithms [27], here we decided to use PCA and nonlinear AE to investigate how the embedded (linear) and true (nonlinear) dimensionality of movements changed over training with the interface. PCA and AE can estimate dimensionality by finding the number of latent components required to reach a predetermined threshold of explained variance. Rather than setting an arbitrary threshold, here we focused on the movement variance explained by PCA and trained AE with one, two and three latent dimensions. We applied PCA and AE on movement signals recorded during 12 consecutive reaching trials on both IMU and robot-joint spaces. As usual, the data was split into 80% of training points and 20% of test points. Three different AEs were trained, with each AE having a different number of neurons in the code layer (one, two and three respectively). Other than the CU size, the AE architectures were equivalent to that of the under-parametrized AE described in Section II.B. Then, we computed the *Variance Accounted For* (*VAF*), defined as the proportion of variance of the IMU and robot-joint signals explained by a lower number of components extracted with PCA and AE during training phases as specified by (1). Given the same latent dimensionality, we expected a nonlinear AE to account for more variance (*i.e.*, higher VAF) than PCA. In other words, we expected the nonlinear dimensionality estimated by the AE to be inferior to the linear dimensionality estimated by PCA for both body and robot movements.

Applying PCA on the robot-joint space also allowed us to investigate which robot coordination participants employed during the online operation of the robot, *i.e.*, how they organized the movements of each robot joint over training time. We considered the first 3 principal components (PCs) derived from applying PCA on the 4D robot joints kinematic data and evaluated their *loadings*. Since we computed the loadings on the 4D robot joint space, we obtained four loadings for each PC - one for each joint. The higher the loading of a joint, the higher its contribution to the corresponding PC. We computed the loadings across day 1 and day 2 to highlight potential differences in the motor strategies employed by participants across days. Similarly, we also wanted to investigate which body coordination patterns participants employed across the two days of training. To do so, we computed the loadings of the first three PCs derived from the 8 IMU body signals across day 1 and 2. In this case, we obtained eight loadings for each PC - one for each IMU component.

Finally, we computed the variance of each of the four robot joints recorded during training to determine the net amount of motion of the four robot joints over blocks of 12 consecutive reaching trials. Since the robot joints had different ranges of motion, we normalized the variance to the maximum extension of each joint angle (360° for the base-joint angle, 180° for the others) and expressed it in percentage. We refer to this metric as *normalized variance*.

F. Statistical Analysis

To test the effect of practice on the indicators related to performance during training, we ran repeated measures analysis of variance (rANOVA) with training time (1-24: start training day 1 to end training day 2, each level consisted in the average of a block of 12 consecutive trials) as within-subjects factor. For the test phases, we ran rANOVA to test the effect of practice on the success rate with time (1-6: baseline day1, mid test day1, final test day 1, baseline day2, mid test day2, final test day2) as within-subject factors. We tested (Kolmogorov-Smirnov test)

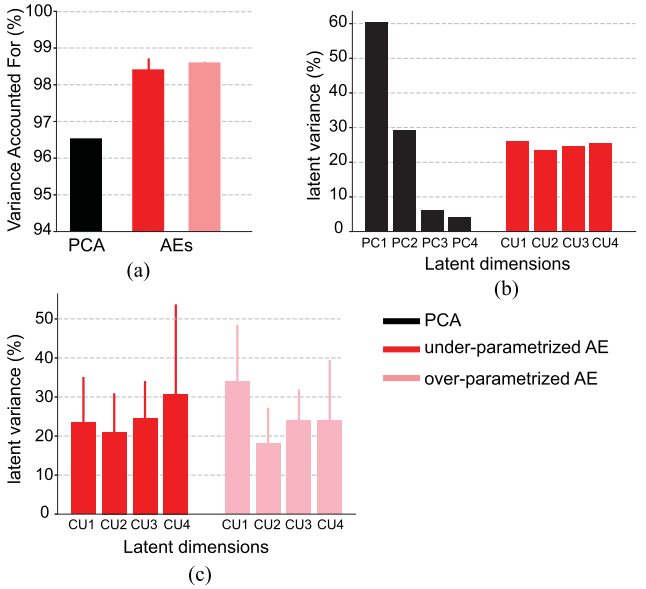

Fig. 2. Analysis of variance on the calibration dataset. (a) Variance accounted for with four latent dimensions by PCA (black bar), under-parametrized AE (red bar) and over-parameterized AE (pink bar). Mean and standard deviation across five different initializations is shown for the nonlinear AEs. (b) Latent variance for PCA (black) and the selected under-parametrized AE (red bars). Nonlinear AE allowed to distribute the variance uniformly along its latent dimensions, while PCA presented the typical drop of variance for the last PCs. (c) Mean and standard deviation of latent variance across the five different initializations for the under-parametrized AE (red) and the over-parametrized AE (pink).

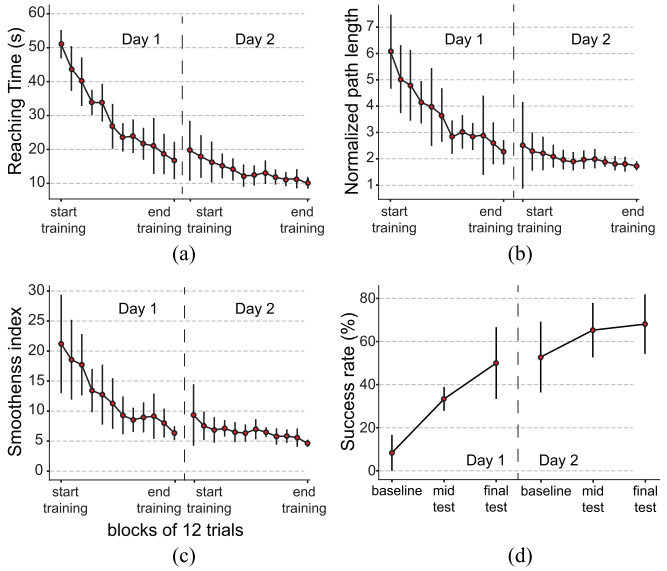

Fig. 3. Participants acquired an adequate level of skill by training with the interface on the first day and retained the performance on the second day. Reaching time (a), normalized path length (b) and smoothness index (c) over training phases, divided into blocks of 12 consecutive trials. (d) Percentage of successful reaching movements over the different test phases. Mean and standard deviation across participants is plotted for each block. A dashed line separates values across day 1 and 2.

the hypothesis that data were normally distributed. We used Mauchly' test [28] to verify the sphericity assumption.

Post-hoc analysis was carried out to verify the overall effect of practice across the two training days and whether users were able to retain their performances across days. To test the effect of learning across days, we ran a Fisher LSD test between the start of training in day 1 (level 1 in the rANOVA for the training metrics and the success rate) and the end of training in day 2 (level 24 for the training metrics and level 6 for the success rate). To test the retention across days, we ran a Fisher LSD test between the end of training in day 1 (level 12 in the rANOVA for the training metrics and level 3 in the rANOVA for the success rate) and the beginning of training in day 2 (level 13 for the training metrics and level 4 for the success rate). The threshold for significance was set at 0.05. For the post-hoc analysis, the threshold for significance was set using the Bonferroni correction (0.025). All analyses were performed in Statistica (Statsoft, Tulsa, OK, USA).

III. Results

A. Validation of the AE

With four latent dimensions, both PCA and nonlinear AE retained over 95% of the variance of the calibration dataset (Fig. 2(a)). As for the AE, the number of parameters did not impact the VAF, as the difference between the under-parametrized and the over-parametrized AEs was negligible (Fig. 2(a)). However, as expected, the latent variance changed depending on the initialization of the AEs, as indicated by the high standard deviation in Fig. 2(c). One of the initializations of the under-parametrized AE yielded a nearly uniform distribution of variance across the code units (Fig. 2(b)). Accordingly, we chose the parameters of this network (specifically of its encoder sub-network) for the online test of the BoMI. Note that if we were to select PCA to be the BoMI forward map, we would obtain a control space in which the variance dropped rapidly across the first principal components (PCs). This would have an impact during the online test, as the movement of the robot joints associated with PC3 and PC4 would be extremely limited or associated with high noise. In fact, a common practice for PCA-based BoMI is to normalize the variance of each PC, so as to make each robot joint variance uniform [29]. However, PC3 and PC4 accounted for only 6% and 4% of the overall calibration movement variance. Thus, they had a significantly smaller signal-to-noise ratio than PC1 and PC2. Accordingly, augmenting their gain would lead to a less efficient control of the associated robot joints.

B. Task-Related Metrics

With practice, all indicators improved between the beginning of the first day and the end of the second day, with a retention across days. Specifically, participants were able to move the robot significantly faster towards the targets ($F(23, 230) = 92.6$, $p < 0.001$, Fig. 3(a)). The largest performance improvement was recorded during the first day of training (from over 50s

to under 20s for a single reaching movement). On the second day of training, performance kept improving and reached a plateau at 10s for a single reaching. In a post-hoc analysis, we found that the performance increase across the training days was significant ($p < 0.001$), while the difference in reaching time between the end of day 1 and the start of day 2 was not ($p = 0.4$), indicating that participants were able to retain their performance across days. Interestingly, there was an effect of training time on both the straightness ($F(23, 230) = 47.5$, $p < 0.001$, Fig. 3(b)) and the smoothness ($F(23, 230) = 24.3$, $p < 0.001$, Fig. 3(c)) of robot trajectories. A post-hoc comparison revealed that the performance improvement from the start of day 1 to the end of day 2 was significant for both metrics ($p < 0.001$ for both NPL and SI). In the second day of training, both movement straightness and smoothness were retained ($p = 0.21$ and $p = 0.2$ respectively).

During the test phase, there was a significant increase in the number of trials completed within 20s as an outcome of training ($F(5, 55) = 47.8$, $p < 0.001$, Fig. 3(d)). The largest improvement was recorded during the first day of training. During baseline, participants were not able to reach almost any of the test targets before timeout (20s), but they managed to complete over 50% of the trials at the end of the day 1. This level of performance was retained at the start of the second day ($p = 0.341$). Participants managed to complete a significantly higher number of trials at the end of day 2, with approximately 70% of them being successful. The overall increase of success rate from the start of the first day to the end of the second day was significant ($p < 0.001$). The level of performance at the end of day 2 is in line with the reaching time achieved by participants during the training sessions at the end of day 2 and the few unsuccessful targets could be due to the test targets being placed in different directions and/or displacement amplitudes with respect to the training ones.

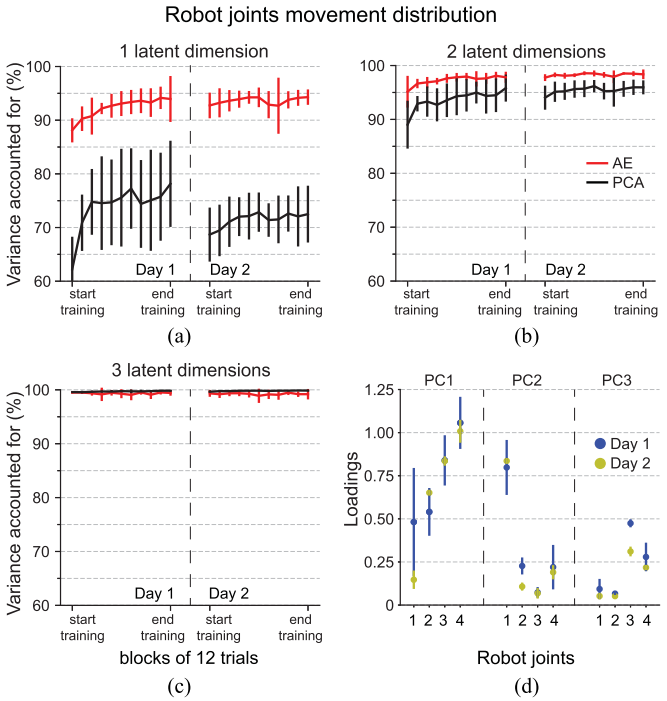

Fig. 4. Analysis on variance on the robot joint movements during training with the interface. VAF with 1 (a), 2 (b), and 3 (c) latent dimensions calculated on 4D robot joints space with nonlinear AE (red) and PCA (black). Mean and standard deviation across participants is shown for blocks of 12 consecutive reaching trials. PCA indicated that robot movements became more linear with interface training but underestimated their true dimensionality. (d) Loadings distribution over the robot joints for the first 3 PCs during day 1 (blue dots) and day 2 (yellow dots). Mean and 95% confidence interval across participants is shown for each loading. The more distal joints had a higher loading than the more proximal ones, especially during day 2. Moreover, the standard deviation across participants of the loadings decreased on day 2, indicating that participants moved the robot joints more consistently with each other after the first day of training.

C. Analysis of Robot Motions

Next, we investigated how the dimensionality of robot joint movements changed with interface training. We found a substantial difference between non-linear AE and PCA for the VAF calculated on the movements of the four robot joints when considering the first latent dimension (Fig. 4(a)). In fact, while AEs values were consistently above 90% with a slight increase over time, PCA values were on average well below 90% and had a higher increase over time. Adding one latent dimension drastically increased the VAF with PCA, but did not have a significant effect on the VAF with nonlinear AE (Fig. 4(b)). When considering three latent dimensions, the VAF was instead similar for both methods and close to 100% (Fig. 4(c)). Taken together, these observations suggest that the robot movements became increasingly linear over time, while their dimensionality, accounted by the nonlinear AE, remained fairly consistent over time (*i.e.*, PCA underestimated the true dimensionality of robot movements).

The loadings distribution over each robot joint showed that in the first day all four joints contributed to the first PC with a predominant contribution of the more distal joints (Fig. 4(d)), while the second PC depended mainly on the base joint. During the second day of training, the loading of the base on PC1 joint drastically decreased with respect to its value on day 1. On the other hand, the loading of PC2 and PC3 remained similar in both days of training. Interestingly, the variance across subjects of all loadings during day 2 decreased drastically from that of day 1, indicating that participants moved the robot joints more consistently with each other after the first day of training.

To get a more comprehensive view of how participants learned to move the robot over training, we also looked at the normalized variance of the robot joints over time. The net amount of motion of the more distal robot joints was superior to that of the more proximal joints (Fig. 5). We also noticed that the movements of the base robot joint were drastically decreased during day 2 and that movements of all the joints were much more consistent across participants during day 2 than during day 1, as the difference (95% confidence interval) across subjects was significantly lower. These results are in line with the robot joints loadings distribution described before.

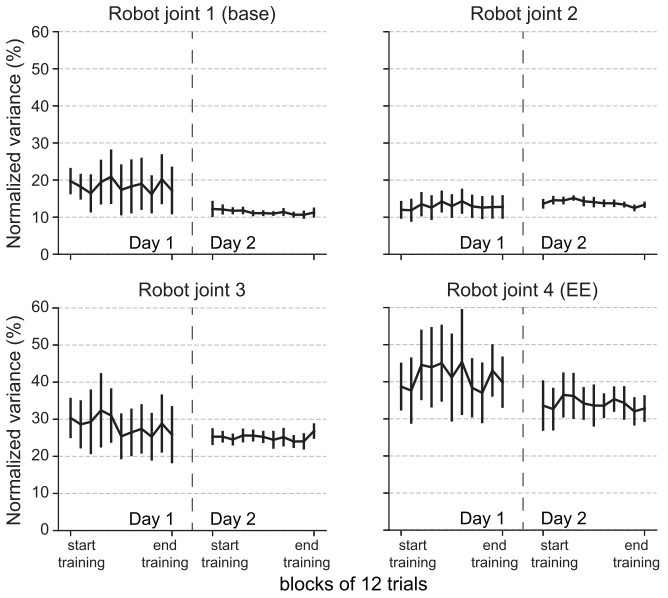

Fig. 5. Analysis on net amount of motion of the robot joint during training with the interface. Percentage of variance of each of the four robot joints recorded during training with respect to the maximum extension of each joint angle is shown for blocks of 12 consecutive reaching trials. Each point represents the mean and 95% confidence interval across participants. The distal joints had higher net motion compared to the proximal joints. During the second day of training, the standard deviation across subjects drastically deceased, indicating that movements of robot joints became more consistent across participants with interface training.

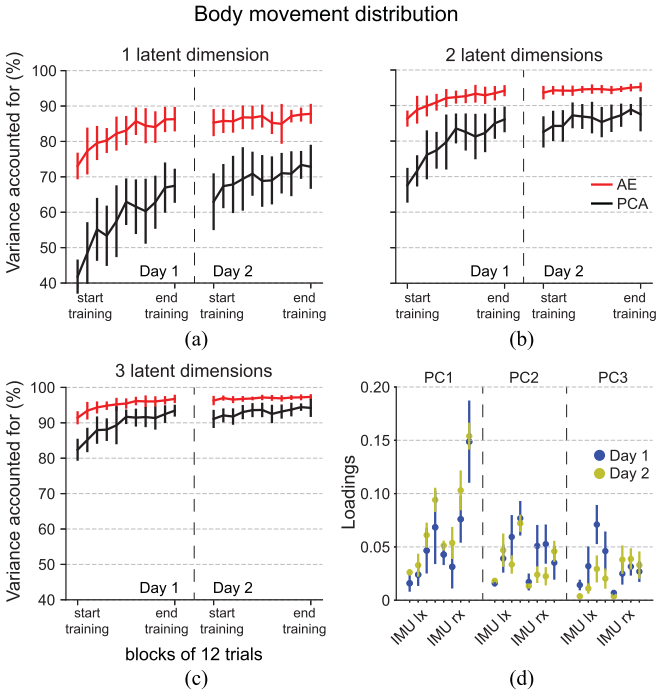

Fig. 6. VAF with 1 (a), 2 (b) and 3 (c) latent dimensions calculated on 8D IMUs space with nonlinear AE (red) and PCA (black). Mean and 95% confidence interval across participants is shown for blocks of 12 consecutive reaching trials. PCA indicated that body movements became substantially more linear with interface training but underestimated their true dimensionality. (d) Loadings distribution over the components of left and right IMU for the first 3 PCs during day 1 (blue dots) and day 2 (yellow dots). The PC1 and PC2 loadings of the IMU placed on the right arm were higher than those of the IMU placed on the left, suggesting that participants employed an asymmetric control. Moreover, the loadings' standard deviation decreased on day 2, indicating that participants movements were more consistent with each other after the first day of training.

D. Analysis in the Signal Space of the Body Motion

At the beginning of the first day, participants started exploring which body movements were more appropriate for reaching the targets. After few training time, the structure of their movement became more linear. In fact, the values of the VAF by PCA with 1 (Fig. 6(a), black lines) and 2 (Fig. 6(b), black lines) latent dimensions were lower at the beginning than at the end of the training, with a sharper increase on the first day. Similarly, the VAF of the AE with 1 and 2 latent dimensions steadily increased and approached a plateau at the end of the first day of training with the interface (Fig. 6(a) and (b), red lines). Adding an additional AE latent dimension had a very minor effect on the VAF, especially during the second day of training (Fig. 6(c), red lines). Conversely, adding the third latent dimension allowed PCA to achieve similar VAF values to those of the AE with 2 latent components (Fig. 6(c), black lines). Therefore, we could infer that PCA underestimated the true dimensionality of the participants' body movements, which the AE estimated to be two from the end of the first day of training onwards.

The loadings distribution over each IMU component for the first 3 PCs during day 1 and day 2 indicated that participants primarily moved their right arm to operate the robot, as the IMU placed on the right arm had the highest loadings for both PC1 and PC2 (Fig. 6(d)). As for the robot joint loadings distribution, we noticed that the variance across participants of the body loadings decreased during day 2, suggesting that participants' body movements became more consistent with each other after the first day of training.

IV. Discussions

In this study, we implemented a nonlinear AE to map arm movements into command signals for controlling a 4D robot. After validating the AE architecture to spread the latent variance uniformly across the four control dimensions, we tested the ability of twelve users to operate the device over two consecutive days of training. The study delivered three main findings: all participants (i) through training with the interface on the first day succeeded to acquire an adequate level of skill when operating the 4D robot; (ii) retained the performance reached by the end of the first day of training; (iii) converged towards a more consistent solution over the course of training. These findings highlight the potential of this interface design as an assistive tool for people with motor impairments.

A. Controlling a Device With Multiple DoFs

Assistive devices such as powered wheelchairs and robotic manipulators can return independence to people with motor impairments. However, controlling a mechanism with multiple DoFs is a specific challenge [11], [30], [31], [32]. Commercially available interfaces can ease the users' burden of controlling the device by making use of a discrete control system. With a

discrete interface, users are given the control over a selection of "modes", where each mode controls a single DoF, or a combination of DoFs. The main advantage of discrete control is its simplicity, as it can be effectively used to control the n-dimensional device. In other words, the user can fully control the device by switching between a few modes. However, discrete controllers do not allow their users to regulate the magnitude of the control signals, thus resulting inadequate for performing dexterous tasks.

To offer greater authority to a device user, one might want to consider a continuous controller. Nonetheless, adopting a continuous control comes with the possibility of introducing an excessive computational and cognitive burden. If the user were to share the control with the machine, this burden could be alleviated and the user autonomy over the device would still be maintained, even if not fully [33]. However, as pointed out in a recent study by Javaremi and Argall [34], there is no one-size-fits-all method for control sharing, as each person is unique in their desired control preference.

In this sense, the approach adopted in our study stands out because, not only it is continuous, but it also allows users to operate the robot in a fully autonomous way. As far as this study is concerned, the computational burden of learning to simultaneously control 4 DoFs was not excessive, as, at the end of the second day of training, participants were able to reach the targets that they explored the most during training in less than 10s (Fig. 3(a)). Moreover, this level of performance is expected to further improve on additional training with the interface, as the learning curve shown in Fig. 3(a) does not seem to have attained convergence yet. The performances are slightly inferior when reaching the test targets (Fig. 3(d)), likely because participants were not used to operate the robot in those directions as much. Again, additional (spatially denser) training should allow participants to further improve performance on the test targets as well. While the performance on the test targets can be seen as more representative of how people with SCI might perform in daily life, it also needs to be said that people with SCI would likely benefit of a more extensive training than that provided in this study. Additionally, our results are comparable to those described in other studies involving the control of the Jaco [35], [36], [37], [38], [39], [40], despite differences in the interface, the controlled DoFs, and the task protocols make a direct comparison with our results rather difficult. Therefore, we argue that the performance levels achieved by participants in this study offer a valid and encouraging baseline that can be further improved with additional training on the interface, by SCI individuals as well.

B. Motion Strategies to Operate the Robot

When users learn to operate a machine via BoMIs, it is crucial to determine how learning takes place. We addressed this point by studying (i) how users distributed their movements over training time and (ii) which strategies they employed to move the robot efficiently toward the targets.

Previous studies suggested that learning to efficiently operate a device with a BoMI comes with a reorganization of users' movements towards a structure whose dimensionality approaches that of the device itself [41], [42]**.** This is not trivial, since the redundant nature of the BoMI forward map would allow participants to maneuver the device with a combination of movements that lies on a higher dimensional space with respect to that of the device itself. However, in these studies the dimensionality of user movements was estimated with a linear method (PCA). PCA can only approximate a participant's control strategy with a linear manifold and may not fully account for the variance of body signals associated with nonlinear control. Hence, in case participants were to employ such nonlinear control strategy, the true dimensionality of participants movements would be underestimated by PCA, thus potentially leading to misleading interpretations. The limitation of using linear methods for this sort of analysis was confirmed by our analysis. In fact, we showed that the true dimensionality of the users' (and robot's) movements was accurately estimated only when using a nonlinear method (AE, Figs. 4 and 6).

Nonetheless, PCA allowed us to further broaden our views on the motor strategies adopted by the participants during the two days of training. Participants started moving the robot with individually distinct strategies, but with practice they converged towards a more consistent solution over the course of training, as indicated by the sharp decrease of the variability across subjects for each PC loading (Figs. 4(d) and 6(d)). Interestingly, we noticed that participants adopted an asymmetric control strategy, as they relied more on the movements of the right arm to operate the robot (Fig. 4(d)). This could either be a result of the participants' personal preference or of the BoMI forward map that we imposed. However, we want to remark that we extracted the BoMI map from a calibration dataset in which each arm contributed equally to the overall movement variance.

Investigating the movement variance of the singular robot joints confirmed that the consistency across participants increased over the course of training (Fig. 5). The robot had 4 joints and the last is the wrist that rotates the end-effector of the robot. The task was 3D and consisted of positioning the end-effector in a 3D space, with the final orientation of the end-effector not being a task variable. However, the end-effector joint was still important in the 3D positioning task because the end-effector itself was not a point, but a two-fingers gripper. The gripper was maintained open and, to solve the task, participants had to place the virtual ball presenting the target inside the two-fingers gripper. Thus, a same task solution could be reached with the gripper fingers in different positions (*e.g.*, both with the same z coordinate or both with the same y coordinate) and – more importantly- by following different trajectories to approach that position. Of course, a subject can choose not to use this possibility and solve the task without using the wrist joint. This will determine not only to always have the gripper fingers on the same position with respect to the target, but it would also require planning the motion accordingly, e.g., by avoiding all trajectories that will result in a collision between the fingers with the target itself.

In summary, the first 3 joints would be sufficient to solve the positioning task in the 3D space, but the redundancy offered

by the additional joint, the wrist, allows users to find solutions (trajectories) that are more comfortable for them, avoiding undesired collisions. In our case, the subjects largely exploited this possibility since the end effector joint was indeed a major contributor to the overall variance of the robot joints trajectories (Fig. 5).

C. Linear vs Non-Linear Dimensionality Reduction in Human-Machine Interfaces

Whether a linear or a nonlinear dimensionality reduction method should be used in a control framework is a question of crucial importance when designing a human-machine interface. In this study, we tried to investigate the differences between these two approaches when estimating kinematics of the upper body and focused on the use of a nonlinear model to control an external device.

Linear DR models, such as PCA and Factor Analysis, have seen the most uptake both in brain machine interfaces [43], [44] and in body machine interfaces [8], [29], [45]. Linear models are convenient due to their low computational cost and, perhaps most importantly, to the possibility of obtaining a full repertoire of actions via summation of simpler actions. However, linear models are intrinsically limited when attempting to reduce the dimensionality -i.e., the complexity- of a nonlinear signal. Since most of the neurophysiological signals have nonlinear properties, linear algorithms are inadequate to derive a low-dimensional latent space [14]. Thus, as both brain and body machine interfaces typically rely on such latent space for representing and controlling an external device, the user's learning with a linear interface could potentially be disrupted.

An alternative approach is to exploit nonlinear DR methods. As demonstrated by recent studies, a nonlinear AE can be used to accurately estimate the dimensionality of full hand kinematics [15] and, more importantly, to efficiently control a 2D computer cursor [16], [17], [18]. Our results confirm these outcomes, as all participants became proficient at operating the 4D robot (Fig. 3). Nevertheless, applications of nonlinear models to human-machine interfaces did not historically arouse the same enthusiasm. A reason for this is that nonlinear models may be difficult to interpret and often need to be specifically tuned to properly suit a particular application. For instance, in this study, finding the most suitable AE structure required a validation procedure, which is naturally time-consuming. As a result, this approach might not be ideal for those contexts where immediacy of use is preferable. On the other hand, if one is willing to sacrifice the convenience of linear models, the intrinsic customizability of nonlinear methods makes them attractive to be tailored to any scenario. For instance, in controlling a device with multiple DoFs, we demonstrated that a properly tuned nonlinear AE can be more suitable than a linear model, as the former allows to uniformly distribute the variance among latent dimensions (Fig. 2). In this study, we decided not to compare the learning performance between AE- and PCA-based interface. The spread of latent variance obtained by PCA was highly unbalanced across its first four principal components (Fig. 2(b)). Specifically, the third and fourth PC (that would have controlled the third and fourth robot joint, respectively) accounted for an incredibly small percentage of movement variance (both less than 5%). In other studies that leveraged PCA as a BoMI map, users were given uniform control over each latent dimension by normalizing the latent variance across the PCs [42], [46]. Since the percentage of variance of the later PCs in this study was so small, normalizing it would have resulted in an excessive increase of noise. As a result, learning how to operate the more distal robot joints with such a PCA-based interface would have been extremely challenging. Importantly, we also showed that even if BoMI users could not exploit the property of superposition offered by linear control, the online operation of the interface was not impaired as all participants managed to acquire an adequate level of skill by the end of the training sessions.

D. Limitations and Future Clinical Applications

Developing new technologies to be applied in clinical context is a leading motivation for this work. However, before testing this paradigm with people with spinal cord injury, stroke, and other causes of disability, it is essential to provide solid proof of their inherent soundness, which is the prime goal of this study. Moreover, there is an important application of this instrument to the basic investigation of motor learning. Our study significantly expands on previous efforts that investigated human motor learning when controlling devices with few degrees of freedom using some linear map [8], [29], [45]. Here, not only we used a nonlinear map, whose application in BoMIs is far less common and its effectiveness far-less proven, but we also tested the control of a device with 4 DoFs, which is obviously a far more challenging task than controlling simpler systems. The test of the BoMI in unimpaired participants serves the specific purpose of providing a normative baseline for understanding of how these different challenges can be solved by the motor control system. This is an essential preliminary step to evaluate performance in a vulnerable population, such as people with SCI or other neurological conditions.

In this study, the BoMI map was designed using a fully unsupervised method. The unsupervised nature of the BoMI makes it ideal for applications in a clinical context since it is based only on exploratory movements and allows to incorporate the subject's residual abilities. Here, we wanted to follow the same philosophy and extend it to conditions that had never been tested before. In particular, we wanted to investigate a purely unsupervised method that was more appropriate for the control of a device with more than 2-3 DoFs, as PCA is inherently limited because the spread of variance along its latent/control dimensions was quite unequal. On the other hand, autoencoder networks, although far-less common in body machine interface applications, provided a uniform variance distribution and resulted in a much more appealing unsupervised technique for controlling devices with many DOFs, despite the drawback of being more time consuming than linear methods due to the necessity of a validation process.

A promising alternative to a fully unsupervised approach was proposed by Macchini et al., where elements of supervised

learning were included during the BoMI map design [12]. When compared to a purely unsupervised approach, the resulting "hybrid" one allowed to speed up learning of BoMI users controlling a quadcopter at the start of training. However, as users practiced with both types of interfaces over time, the difference between the hybrid and the unsupervised approach became marginal by the end of the session. These results are consistent with ours, as we also found that all the participants struggled to operate the robot with the unsupervised interface at first (Fig. 3, day 1, start training), but became significantly much more efficient with training (also Fig. 3). Nonetheless, our proposed platform could be further improved by designing an interface that is based on nonlinear autoencoders but includes elements of supervised learning, and is also adaptive, as the process of co-adaptation has shown to guide the redundancy resolution towards increased movement efficiency [18], [41].

The BoMI map used in this study was derived from the AE validation procedure performed using data from the single-male subject (who did not participate in the learning study presented in the manuscript). While the overall philosophy of BoMIs is to base their operation on a subject-specific mapping, here we decided to use the same BoMI map for all participants for the following reasons. First of all, we wanted all participants to start from a common initial condition to ensure a fair comparison of their learning curves during the operation with the interface. Our group has used this design in previous studies as well [18], [47], [48]. Secondly, a subject-specific mapping implemented in real life is time consuming, as the process of AE validation requires to train several AE models. When testing the interface in a clinical context, however, it will be necessary to perform the AE validation procedure using a calibration dataset collected on the impaired participant to ensure the BoMI map to be better suited to their reduced movement distribution.

V. Conclusion

In the context of non-invasive human-machine interfaces, the use of non-linear methods for generating commands to operate external devices has not been broadly studied. In this study, we filled this gap of knowledge by envisioning a novel interface that applies nonlinear autoencoder networks to convert arm movements into a control signal for a 4D virtual manipulator. We hypothesized that the customizability offered by an autoencoder makes it a versatile control platform for complex multi-dimensional devices. Our findings support this hypothesis, as a cohort of participants were able to acquire an adequate level of skill by training with the interface and retain such performance over two non-consecutive days of training. Importantly, the approach tested here provided users with a fully continuous control of the robot. Moreover, its entirely unsupervised nature makes it ideal for applications in a clinical context since it can be tailored to each user's residual movements.

References

[1] L. Fehr, W. E. Langbein, and S. B. Skaar, "Adequacy of power wheelchair control interfaces for persons with severe disabilities: A clinical survey," *J. Rehabil. Res. Develop.*, vol. 37, no. 3, pp. 353–360, 2000.

[2] T. F. Bastos-Filho et al., "Towards a new modality-independent interface for a robotic wheelchair," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 22, no. 3, pp. 567–584, May 2014.

[3] U. Koike et al., "Development of an intraoral interface for human-ability extension robots," *J. Robot. Mechatronics*, vol. 28, no. 6, pp. 819–829, 2016.

[4] L. N. S. A. Struijk et al., "Wireless intraoral tongue control of an assistive robotic arm for individuals with tetraplegia," *J. Neuroeng. Rehabil.*, vol. 14, no. 1, 2017, Art. no. 110.

[5] M. Casadio, R. Ranganathan, and F. A. Mussa-Ivaldi, "The body-machine interface: A new perspective on an old theme," *J. Motor Behav.*, vol. 44, no. 6, pp. 419–433, Nov. 2012, doi: 10.1080/00222895.2012.700968.

[6] J. Miehlbradt et al., "Data-driven body–machine interface for the accurate control of drones," *Proc. Nat. Acad. Sci. USA*, vol. 115, no. 31, pp. 7913–7918, Jul. 2018, doi: 10.1073/pnas.1718648115.

[7] P. Geethanjali, "Myoelectric control of prosthetic hands: State-of-the-art review," *Med. Devices (Auckland, NZ)*, vol. 9, 2016, Art. no. 247.

[8] F. Rizzoglio et al., "A hybrid body-machine interface integrating signals from muscles and motions," *J. Neural Eng.*, vol. 17, Jun. 2020, Art. no. 046004, doi: 10.1088/1741-2552/ab9b6c.

[9] E. B. Thorp et al., "Upper body-based power wheelchair control interface for individuals with tetraplegia," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 24, no. 2, pp. 249–260, Feb. 2016, doi: 10.1109/TNSRE.2015.2439240.

[10] S. Jain et al., "Assistive robotic manipulation through shared autonomy and a body-machine interface," in *Proc. IEEE Int. Conf. Rehabil. Robot.*, 2015, pp. 526–531, doi: 10.1109/ICORR.2015.7281253.

[11] J. M. Lee et al., "Diversity of learning to control complex rehabilitation robots using high-dimensional interfaces," 2022.

[12] M. Macchini, F. Schiano, and D. Floreano, "Personalized telerobotics by fast machine learning of body-machine interfaces," *IEEE Robot. Automat. Lett.*, vol. 5, no. 1, pp. 179–186, Jan. 2020, doi: 10.1109/LRA.2019.2950816.

[13] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics Intell. Lab. Syst.*, vol. 2, pp. 37–52, 1987, doi: 10.1016/0169-7439(87)80084-9.

[14] J. B. Tenenbaum, V. De Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000, doi: 10.1126/science.290.5500.2319.

[15] A. A. Portnova-Fahreeva et al., "Linear and non-linear dimensionality-reduction techniques on full hand kinematics," *Front. Bioeng. Biotechnol.*, vol. 8, pp. 1–18, 2020, doi: 10.3389/fbioe.2020.00429.

[16] C. Pierella et al., "Linear vs non-linear mapping in a body machine interface based on electromyographic signals," in *Proc. IEEE Int. Conf. Biomed. Robot. Biomechatronics*, 2018, pp. 162–166, doi: 10.1109/BIOROB.2018.8487185.

[17] I. Vujaklija et al., "Online mapping of EMG signals into kinematics by autoencoding," *J. Neuroeng. Rehabil.*, vol. 15, no. 1, 2018, Art. no. 21, doi: 10.1186/s12984-018-0363-1.

[18] F. Rizzoglio et al., "Building an adaptive interface via unsupervised tracking of latent manifolds," *Neural Netw.*, vol. 137, pp. 174–187, 2021, doi: 10.1016/j.neunet.2021.01.009.

[19] M. Giordano et al., "Controlling an assistive robotic manipulator via a non-linear body-machine interface," in *Proc. Int. Conf. NeuroRehabil.*, 2020, pp. 685–689.

[20] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2004, pp. 2149–2154.

[21] M. Quigley et al., "ROS: An open-source robot operating system," in *Proc. ICRA Workshop Open Source Softw.*, 2009, vol. 3, no. 3.2, p. 5.

[22] M. Casadio et al., "Functional reorganization of upper-body movement after spinal cord injury," *Exp. Brain Res.*, vol. 207, no. 3/4, pp. 233–247, 2010, doi: 10.1007/s00221-010-2427-8.

[23] B. L. Kalman and S. C. Kwasny, "Why tanh: Choosing a sigmoidal function," in *Proc. Int. Joint Conf. Neural Netw.*, 1992, pp. 578–581, doi: 10.1109/ijcnn.1992.227257.

[24] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," *J. Mach. Learn. Res.*, vol. 9, pp. 249–256, 2010.

[25] S. S. Du et al., "Gradient descent provably optimizes over-parameterized neural networks," in *Proc. 7th Int. Conf. Learn. Represent.*, 2019, pp. 1–19.

[26] A. Campeau-Lecours et al., "Kinova modular robot arms for service robotics applications," in *Rapid Automation: Concepts, Methodologies, Tools, and Applications*. IGI Global, 2019, pp. 693–719.

[27] E. Altan et al., "Estimating the dimensionality of the manifold underlying multi-electrode neural recordings," *Plos Comput. Biol.*, vol. 17, no. 11, pp. 1–23, 2021, doi: 10.1371/journal.pcbi.1008591.
[28] J. W. Mauchly, "Significance test for sphericity of a normal n-variate distribution," *Ann. Math. Statist.*, vol. 11, no. 2, pp. 204–209, 1940.
[29] A. Farshchiansadegh et al., "A body machine interface based on inertial sensors," in *Proc. 36th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, 2014, pp. 6120–6124, doi: 10.1109/EMBC.2014.6945026.
[30] R. E. Cowan et al., "Recent trends in assistive technology for mobility," *J. Neuroeng. Rehabil.*, vol. 9, no. 1, pp. 1–8, 2012, doi: 10.1186/1743-0003-9-20.
[31] R. R. Kaliki, R. Davoodi, and G. E. Loeb, "Evaluation of a noninvasive command scheme for upper-limb prostheses in a virtual reality reach and grasp task," *IEEE Trans. Biomed. Eng.*, vol. 60, no. 3, pp. 792–802, Mar. 2013, doi: 10.1109/TBME.2012.2185494.
[32] R. Ranganathan et al., "Age-dependent differences in learning to control a robot arm using a body-machine interface," *Sci. Rep.*, vol. 9, 2019, Art. no. 1960, doi: 10.1038/s41598-018-38092-3.
[33] D. Gopinath, S. Jain, and B. D. Argall, "Human-in-the-loop optimization of shared autonomy in assistive robotics," *IEEE Robot. Automat. Lett.*, vol. 2, no. 1, pp. 247–254, Jan. 2017.
[34] M. N. Javaremi et al., "Characterization of assistive robot arm teleoperation: A preliminary study to inform shared control," 2020, *arXiv:2008.00109*.
[35] A. Lebrasseur et al., "Assistive robotic device: Evaluation of intelligent algorithms," 2018, *arXiv:1812.07342*.
[36] A. Campeau-Lecours et al., "Intuitive adaptive orientation control for enhanced human-robot interaction," *IEEE Trans. Robot.*, vol. 35, no. 2, pp. 509–520, Apr. 2019, doi: 10.1109/TRO.2018.2885464.
[37] S. Li et al., "Intuitive control of a robotic arm and hand system with pneumatic haptic feedback," *IEEE Robot. Automat. Lett.*, vol. 4, no. 4, pp. 4424–4430, Oct. 2019, doi: 10.1109/LRA.2019.2937483.
[38] H. Jiang et al., "3D joystick for robotic arm control by individuals with high level spinal cord injuries," in *Proc. IEEE 3th Int. Conf. Rehabil. Robot.*, 2013, pp. 1–5, doi: 10.1109/ICORR.2013.6650432.
[39] L. V. Herlant, R. M. Holladay, and S. S. Srinivasa, "Assistive teleoperation of robot arms via automatic time-optimal mode switching," in *Proc. 11th ACM/IEEE Int. Conf. Hum.-Robot Interact.*, 2016, pp. 35–42, doi: 10.1109/HRI.2016.7451731.
[40] L. N. S. A. Struijk and R. Lontis, "Comparison of tongue interface with keyboard for control of an assistive robotic arm," in *Proc. Int. Conf. Rehabil. Robot.*, 2017, pp. 925–928, doi: 10.1109/ICORR.2017.8009367.
[41] D. De Santis and F. A. Mussa-Ivaldi, "Guiding functional reorganization of motor redundancy using a body-machine interface," *J. Neuroeng. Rehabil.*, vol. 17, 2020, Art. no. 61.
[42] C. Pierella et al., "Learning new movements after paralysis: Results from a home-based study," *Sci. Rep.*, vol. 7, no. 1, pp. 1–11, 2017, doi: 10.1038/s41598-017-04930-z.
[43] A. D. Degenhart et al., "Stabilization of a brain–computer interface via the alignment of low-dimensional spaces of neural activity," *Nature Biomed. Eng.*, vol. 4, pp. 672–685, 2020, doi: 10.1038/s41551-020-0542-9.
[44] G. Santhanam et al., "Factor-analysis methods for higher-performance neural prostheses," *J. Neurophysiol.*, vol. 102, no. 2, pp. 1315–1330, 2009.
[45] P. Carrington, A. Hurst, and S. K. Kane, "Wearables and chairables: Inclusive design of mobile input and output techniques for power wheelchair users," in *Proc. SIGCHI Conf. Hum. Factors Comput. Syst.*, 2014, pp. 3103–3112.
[46] F. Abdollahi et al., "Body-machine interface enables people with cervical spinal cord injury to control devices with available body movements: Proof of concept," *Neurorehabil. Neural Repair*, vol. 31, no. 5, pp. 487–493, 2017, doi: 10.1177/1545968317693111.
[47] D. De Santis, P. Dzialecka, and F. A. Mussa-Ivaldi, "Unsupervised coadaptation of an assistive interface to facilitate sensorimotor learning of redundant control," in *Proc. 7th IEEE Int. Conf. Biomed. Robot. Biomechatronics*, 2018, pp. 801–806, doi: 10.1109/BIOROB.2018.8487912.
[48] J. M. Lee et al., "Learning to control complex robots using high-dimensional interfaces: Preliminary insights," 2021, *arXiv:2110.04663*.