2018 15th International Conference on
Control, Automation, Robotics and Vision (ICARCV)
Singapore, November 18-21, 2018

Universal, Open Source, Myoelectric Interface for Assistive Devices*

Adam Naber, *Student Member, IEEE,* Yiannis Karayiannidis, *Member, IEEE,*
Max Ortiz-Catalan, *Member, IEEE*

Abstract— **We present an integrated, open-source platform for the control of assistive vehicles. The system is vehicle-agnostic and can be controlled using a myoelectric interface to translate muscle contractions into vehicular commands. A modular shared-control system was used to enhance safety and ease of use, and three collision avoidance systems were included and verified in both an included test platform and on a quadcopter operating in a simulated environment. Seven subjects performed the experiments and rated the user experience of the system under each of the provided collision avoidance systems with positive results. Qualitative tests with the quadcopter validated the proposed system and shared-control techniques. This open-source platform for shared control between humans and machines integrates decoding of motor volition with control engineering to expedite further investigation into the operation of mobile robots.**

I. Introduction

Missing limbs, partial loss of muscular control, and weak musculature all inhibit personal mobility and can significantly impact a person's overall quality of life [1]. Traditional powered wheelchairs aim to reduce some of the negative effects of these conditions but still require both dexterity and concentration to safely operate [2], [3]. The use of gesture recognition offers an alternative avenue for vehicular control [4], but still requires volitional limb control to use. Significant research has been done to utilize signals from the brain and any remaining functional musculature to replace or augment lost or impaired mobility and extremity function [5]–[10]. Electroencephalography (EEG) and electromyography (EMG) signals offer opportunities for control of assistive devices, even in cases of tetraplegia, where the patients are unable to move any of their limbs. Unfortunately, such human machine interfaces are prevented from achieving the same information transfer rates as a conventional joystick due to environmental noise and the stochastic nature of the brain and myoelectric signals. These obstacles can make it difficult to safely and efficiently operate a personal vehicle using bioelectric signals alone [6], [7]. EMG signals have been a focus of study due to their relatively high throughput and direct correlation with motor intention. This comes with the caveats of muscle fatigue from constant contractions and poor controllability from the sequential nature of most myoelectric pattern recognition (MPR) algorithms [5], [9]. However, some experimental MPR strategies allow more natural control, such as simultaneous control of different degrees of freedom. In addition, the speed or force in the end effector can be adjusted to the level of muscular contraction, referred to as proportional control [10], [11].

Shared control systems, where a computer system is used to augment user control based on the environment, can be used to lower the mental load and physical fatigue associated with controlling a personal vehicle with EMG signals while increasing the safety and controllability of the device. One simple strategy is to allow the user to set the velocity and direction of the vehicle and have the system stop if it gets too close to an obstacle. This approach is often insufficient to smoothly navigate complex environments, highlighting the need for more advanced systems [12]. In addition, lack of control or feedback over the level of computer assistance provided can lead to frustration from users [2]. A balance must be struck between the agency an operator exerts on the device, and the intelligent interpretation of the received commands with respect to the local environment.

Dynamic control systems have been shown to significantly improve the safety and controllability of assistive vehicles controlled by joystick or computer interface [13], [14]. Rather than having to split a task into discrete operations, namely repeatedly stopping and turning, the system allows the user to specify the overall direction and proceed from there. Such systems also reduce the cognitive workload of navigation by reducing the number of commands required to control a vehicle. Previous work on using dynamic role adaptation to determine the level of control the user exerts on the system showed promising results [15], [16], but much of the research on the topic relies on custom made hardware and software or are Wizard-of-Oz type setups [2], [4]–[6], [17]. This necessitates the development of a generic and extensible interface for personal vehicular control in research for faster production and better comparative analyses.

*Research supported by European Commission (H2020: DeTOP), Stiftelsen Promobilia, VINNOVA., and the Swedish Research Council (VR).

Adam Naber is with the Biomechatronics and Neurorehabilitation Laboratory in the Department of Electrical Engineering at Chalmers University of Technology, Gothenburg, 412 58 Sweden (e-mail: naber@chalmers.se)

Yiannis Karaviannidis is with the Department of Electrical Engineering at Chalmers University of Technology, Gothenburg, 412 58 Sweden and the Department of Robotics and Perception and Learning at the Royal Institute of Technology, Stockholm 114 28 Sweden (e-mail: yiannis@chalmers.se)

Max Ortiz-Catalan is with the Biomechatronics and Neurorehabilitation Laboratory in the Department of Electrical Engineering at Chalmers University of Technology, Gothenburg, 412 58 Sweden and Integrum AB, Mölndal, 431 37 Sweden (phone: +46-31-772-51-49 e-mail: maxo@chalmers.se)

978-1-5386-9582-1/18/$31.00 ©2018 IEEE

In this work, we provide an open-source, integrated platform for the investigation of safe and generic vehicular control based on myoelectric pattern recognition. The framework is largely based on existing open-source projects which have the benefits of community support and the use of established development frameworks. We implemented three control algorithms for reference and used the framework to implement a user experience study comparing the assistance provided by each using both an included testing platform and a quadcopter as a test vehicle in Gazebo, a robotics simulation environment.

II. Materials and Methods

A. Vehicle, Sensors, and Protocol Selection

An Erle Robotics ErleBrain3 autopilot module was used in this research as hardware-in-the-loop simulator, performing physics calculations, communication, and motor control for the robotics simulator. The on-board firmware used was a branch of the ArduPilot autopilot firmware modified to communicate with a virtual LiDAR unit for collision avoidance. ArduPilot is an open-source, generic autopilot system capable of controlling various vehicle types, including ground, air, and submersible vehicles, with little to no modification. Vehicle agnostic control was deemed important for this work to allow for transferability between different types of assistive devices. The modified source code is available on GitHub along with all collision avoidance and experimental code used in this work [18].

Webcams and ultrasonic range sensors have previously been used as detection modalities [13]–[15], but LiDAR systems have recently become inexpensive enough to include in lower cost, consumer devices. The LiDAR sensor used provided information about obstacles in a 360 degree plane relative to the axis of the vehicle. The range scans were described in polar coordinates with a range of 40 meters, a distance resolution of 1 cm and an angular resolution of roughly 3.6 degrees. These specifications were based on the commercially-available Scanse Sweep LiDAR system.

The Micro Air Vehicle Link (MAVLink) protocol through the DroneKit SDK was used for communication between the vehicle and the base station. MAVLink and DroneKit were chosen, as they are both actively maintained, open-source protocols for communicating with unmanned vehicles and support communication and control of any vehicle type supported by the ArduPilot firmware.

Sensor data received from the LiDAR unit was continuously sent to the base station for processing. The MAVLink protocol supports proximity sensor data in eight 45 degree sectors, with the first sector centered along the vehicle heading direction. Therefore, data sent over the link consisted of the minimum distance along each sector up to a maximum distance of 40 meters. This treated any detected obstacle as taking up an entire 45 degree arc, Fig. 1.

B. User Control

The user interface was designed to allow the user to influence the relative velocity and heading of the vehicle, ϕ_{user} and υ_{user}, respectively, through patterns of muscular contractions. EMG signals were obtained using the BioPatRec research platform coupled with the ADS_BP signal acquisition unit [19], [20]. BioPatRec was also used to provide EMG signal processing, pattern recognition, and commands corresponding to the user's motor intention. Muscle contractions were each linked to target behaviors in the vehicle, with closing and opening the hand controlling velocity and wrist flexion and extension turning the vehicle. Additionally, in the quadcopter simulations, wrist pronation and supination controlled the altitude of the vehicle. In this configuration, user controls were additive, meaning no contraction would have the vehicle travel in a straight line at its current velocity, closing the hand would increase the velocity, and flexing or extending the wrist would turn the vehicle.

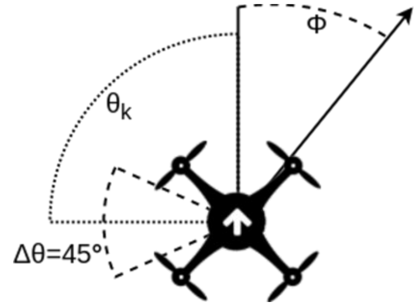

Figure 1: Diagram showing the sector widths ($\Delta\theta$), relative angle of the heading (ϕ), and sector angles (θ_k) with respect to the front of the vehicle (white arrow).

C. Shared Control Approach

Three collision avoidance methods were implemented to augment user control with sensor data: halt-on-proximity (HOP), gain attenuation (GA), and dynamic control (DC). These methods were based on continuous impulsive force, linear function, and potential field proximity constraints, respectively, commonly used in robotic control systems [21]. Parameters for each avoidance method were manually tuned in the provided simulator. Additionally, a hard limit on the vehicle's target velocity, υ_{max}, was set to ensure stable operation. User velocity and angle inputs, υ_{user} and ϕ_{user}, were summed with the outputs of the selected avoidance method, υ_{ca} and ϕ_{ca}, to produce the total output sent to the vehicle. An intermediate velocity value, υ_{int}, is used to allow the collision avoidance system to operate on both the current vehicle state, υ and ϕ, and the user input (1-3).

$$v_{int} = v + v_{user} \tag{1}$$

$$v_{new} = \min\left(v_{max}, v_{int} + v_{ca}\right) \tag{2}$$

$$\phi_{new} = \phi + \phi_{user} + \phi_{ca} \tag{3}$$

The angle of each sensor, θ_k, relative to the vehicle heading, is defined as the midpoint of each 45 degree sector. For brevity, we define a set of sectors, K, that consist of only forward-facing sensors, relative to the direction of movement, ϕ, defined as follows:

$$K = \{k \in \{0, 1, ..7\} \mid sgn(v)cos(\phi - \theta_k) > 0\} \tag{4}$$

The HOP method set a hard threshold such that the resulting vehicle velocity, υ_{new}, dropped to 0 if the distance between the vehicle and an obstacle in the heading direction was smaller than a threshold d_{thr}. Using this method, users could move in one degree of freedom at a time, selecting to turn, move forward, or move backwards. HOP was also used in the GA and DC algorithms with a lower threshold to guarantee the vehicle would not crash into an obstacle, and is defined as follows:

$$v_{ca,HOP} = \begin{cases} 0, & \text{if } d_k > d_{thr} \,\forall\, k \in K \\ -v_{int}, & \text{otherwise} \end{cases} \tag{5}$$

$$\phi_{ca,HOP} = 0 \tag{6}$$

where $v_{ca,HOP}$ is the HOP velocity contribution and d_k is the distance measured in sector k.

GA provided a soft threshold on the vehicle velocity, to maintain a constant minimum time-to-contact, t_c, with any observed obstacles, and is defined as follows:

$$v_{ca,GA} = \min_{k \in K}\left(\min\left(v_{int}, \frac{d_k}{t_c}\right)\right) - v_{int} \tag{7}$$

$$\phi_{ca,GA} = 0 \tag{8}$$

DC operated on both the angular and forward velocities of the vehicle. It constructed virtual repellers that smoothly guided the vehicle around detected obstacles. Forward velocity was limited to allow the vehicle sufficient time to rotate depending on the proximity of the obstacles. This included using $v_{ca,GA}$, defined above, as a soft limit on the velocity. If the vehicle velocity increased above the limit, the system would produce repellors of exponentially greater strength to slow the vehicle. The forward velocity and relative angle of the heading were adjusted based on (9) and (10) respectively as follows:

$$v_{ca,DC} = c_{obs} v_{ca,GA} exp\left[\frac{-v_{ca,GA}^2}{8v_{max}^2}\right] \tag{9}$$

$$\phi_{ca,DC} = \sum_k \lambda_k(\phi - \theta_k) exp\left[\frac{-2(\phi - \theta_k)^2}{\Delta\theta_k^2}\right] \tag{10}$$

$$\text{where } \lambda_k = \beta_1 exp\left[-\frac{d_k}{\beta_2}\right] \tag{11}$$

$$c_{obs} = \frac{c_{v,obs}}{\pi}\left(\frac{\pi}{2} + atan(c_v \xi)\right) \tag{12}$$

$$\xi = \sum_k \lambda_k \left(\frac{\Delta\theta_k^2}{4}\right)\left(exp\left[\frac{-2\theta_k^2}{\Delta\theta_k^2}\right] - \sqrt{e}\right) \tag{13}$$

where $\Delta\theta_k$ represents the angular width of sector k and $c_{v,obs}$, c_v, β_1, and β_2 are selectable parameters that describe the linear strength, linear range, angular strength and angular range of the repeller dynamics, respectively. A rigorous analysis of the concept can be found in Bicho and Schöner's work on behavior-based robotics [22].

Bicho and Schöner's work was originally designed for fully autonomous systems and required both repellers and an attractor for path planning. The implementation in the current work ignores the target attractor in lieu of the user input, and only uses repellers to attenuate the velocity of the vehicle enough to accommodate how quickly the vehicle can turn to avoid an obstacle. The angular velocity resulting from the above dynamics is an exponential function of distance. This allows for smooth changes that are easily overridden by the user over long distances or slow speeds but requires conscious effort to override if the system detects an imminent collision.

D. Testing Procedure

Seven able-bodied subjects between the ages of 23 and 31 (μ=27, SD=3.3) participated in this experiment. Surface EMG signals were recorded using four sets of Ag/AgCl disposable electrodes placed with approximately equal spacing along the proximal third of the subject's desired forearm. Four target movements were chosen to provide a set of commands for the vehicle control, with *hand close* and *hand open* controlling the velocity and *wrist flexion* and *wrist extension* controlling the heading direction of the vehicle. The target movements were recorded in BioPatRec using the built-in recording session protocol. The EMG signals were sampled at 1000 Hz and split into 200 ms windows with 150 ms overlaps for feature extraction. Features were calculated using *mean absolute value, window length, signed slope change,* and *zero crossings* from the Hudgins' time-domain feature set [23], [24]. These data were then used to train a linear discriminate analysis classifier in a One-vs-All scheme.

A goal-directed vehicle simulator was developed in Python, and adapted to execute the control commands outputted from the BioPatRec pattern recognition module. The simulator supplied collision sensors measurements similarly to what provided by MAVLink (Fig. 2). After training the pattern recognition system, the subjects were asked to control the vehicle to familiarize themselves with the control scheme. The recording, training and warm-up steps were repeated in cases of poor controllability before the start of the experiment due to poorly placed electrodes or mistakes made during classifier training. The experiment required the subjects to direct the vehicle to target positions plotted on screen within three minutes time. This task was repeated using each of the collision avoidance algorithms with a random execution order.

Testing metrics included: the total time to complete the task, the time spent in collision with an obstacle, the idle time, and an ordinal rating from 0 (unusable) to 10 (helpful and intuitive). The idle time was considered any time the subject was not sending any command. Subjects were asked to focus on the performance of the shared control system rather than the pattern recognition interface when rating each system to determine if there was a preference, as the feasibility of myoelectric signals in vehicular control has already been established [5], [6], [17]. Results for each

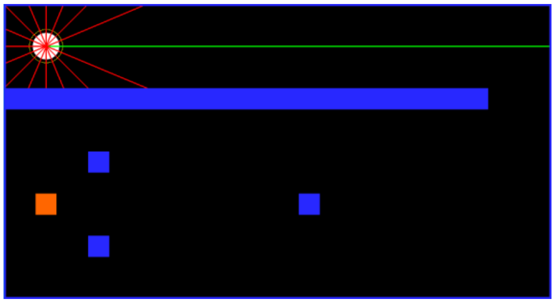

Figure 2: Goal directed vehicle simulator. Blue areas represent obstacles, the white circle represents the vehicle with the heading along the green line, the orange box indicates the target, and the red and green lines indicate the scan lines of the LiDAR unit.

algorithm were compared using one-way ANOVA analysis, corrected with the Tukey-Kramer criterion.

Moreover, the collision avoidance algorithms were preliminarily tested on a quadcopter in a simulated environment. The quadcopter was tested with conservative values for each of the collision avoidance parameters, due to the complex flight dynamics of the system. The user control remained the same as in the simulated environment, with BioPatRec decoding upper arm signals and translating them to vehicle commands. The testing area was an outdoor environment with several large obstacles in between the starting and target locations. Each avoidance method was tested to ensure adequate functionality of the system.

III. Results

A. Comparison of Operation with Different Controllers in Simulation

No statistically significant difference was found in any tested metrics, Fig. 3. However, there was a trend for more intelligent systems, namely GA and DC, to have the lower completion times. All test subjects, except one while testing the HOP method, were able to complete the tasks within the allotted time. Collisions were only reported on the DC avoidance algorithm, indicating that more effort is required to tune the parameters for better safety. Idle time was not significantly affected by the control algorithm, indicating that all tested algorithms impose a roughly equal mental and physical load on the subjects. All algorithms had positive ratings, suggesting that the user experience of each system was intuitive. Tests with more subjects may show more discrepancies and help the researchers optimize the safety and control parameters in the future.

B. Testing the Controller on a Quadcopter

The quadcopter avoided all obstacles in the testing environment under each avoidance algorithm. Qualitative analysis of the flight pattern suggests that the HOP method poses the highest risk of a collision. This method implicitly requires a low distance threshold to allow sufficient controllability in the presence of obstacles, which could be too small for the vehicle to stop when approaching an obstacle at maximum velocity. The DC method exhibited the smoothest trajectory when traversing environments, as shown in Fig. 4.

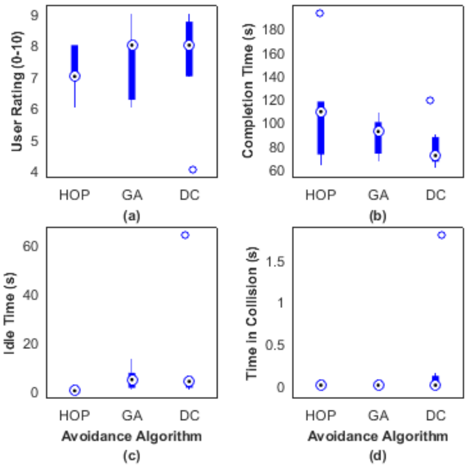

Figure 3: Box plots showing the distributions of (a) user ratings, (b) completion time, (c) idle time, and (d) time spent in collision with an object for each avoidance algorithm. A circle with a dot indicates the median value, the thick bars indicate the interquartile range, and empty circles indicate outlier values.

IV. Discussion & Conclusions

The MAVLink protocol only supports eight directions in the flight plane for proximity sensing. Smoother control for collision avoidance can be achieved by defining custom commands to enhance the protocol. Dynamically modifying the strength of the avoidance system parameters based on the integrity of the user control is also an avenue for further investigation, as it has shown initial promise in previous research [12], [15], [16]. Sanders' work also included the use of lateral control for collision avoidance, which the proposed system is capable of implementing but was left out of the current investigation [15].

In this work, all parameters for the shared control systems were tuned by hand. Future work will include a means of incorporating vehicle dynamics properties to directly formulate parameters that form a safe and controllable system. This can be augmented in real-time with additional sensors that record environmental and user data to ensure a safe and effective system in changing conditions.

Subjects reported that classification errors confusing any movement with an *open-hand* movement often stopped the vehicle during testing and caused frustration. This suggests the use of a velocity ramp, majority voting scheme, or some other method to reduce the effect of single classification errors or erroneous movements [25], [26]. Investigation into simultaneous and proportional control may also show a positive impact on vehicle controllability. Subjects also cited the lack of visual feedback for both classified movements and the amount of control the avoidance system exhibited as causes of dissatisfaction. An on-screen display of the user controls and the avoidance system activation should remedy

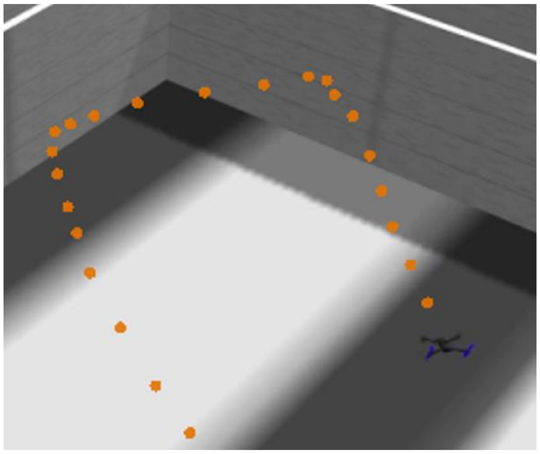

Figure 4: Quadcopter in Gazebo simulator under the influence of the DC collision avoidance method. Orange dots indicate the path followed, starting near the bottom left, when directed to move forward. Note that the quadcopter slowed and turned to avoid the walls without user intervention.

these issues and will be implemented in future work.

There is enough on-board processing power in the autopilot module to create a system that does not require a base station to operate, allowing the signal acquisition system to be connected directly over a Bluetooth or Wi-Fi link. This would require implementing the pattern recognition routines in firmware, but it would reduce the cost and increase the portability of the system. This system also has the potential to control computer interfaces if combined with graphical user interface automation and image recognition technologies.

Here we presented an open-source, integrated platform for the investigation of safe and intuitive vehicular control for people with reduced mobility based on myoelectric pattern recognition. The implemented collision avoidance algorithms all performed satisfactorily through validation in a user study. Their functionality was also confirmed by tests in a virtual environment. All relevant code and documentation for the simulator, collision avoidance system, and communication layer between BioPatRec and DroneKit were uploaded to a GitHub repository available for public use [18].

Acknowledgment

The authors would like to thank the participants who provided data, valuable feedback, and ideas for future work. The authors would also like to thank the Chalmers SEED Project program for creating the opportunities for this collaboration.

References

[1] M. Finlayson and T. van Denend, “Experiencing the loss of mobility: Perspectives of older adults with MS,” *Disabil. Rehabil.*, vol. 25, no. 20, pp. 1168–1180, 2003.

[2] I. M. Mitchell, P. Viswanathan, B. Adhikari, E. Rothfels, and A. K. Mackworth, “Shared control policies for safe wheelchair navigation of elderly adults with cognitive and mobility impairments: Designing a wizard of oz study,” in *American Control Conference*, 2014, pp. 4087–4094.

[3] R. A. C. Simpson, Richard C., Edmund F. LoPresti, “How many people would benefit from a smart wheelchair?,” *J. Rehabil. Res. Dev.*, vol. 45, no. 1, pp. 53–72, 2008.

[4] L. F. Sanchez, H. Abaunza, and P. Castillo, “Safe navigation control for a quadcopter using user’s arm commands,” *Int. Conf. Unmanned Aircr. Syst.*, pp. 981–988, 2017.

[5] Jeong-Su Han, Z. Zenn Bien, Dae-Jin Kim, Hyong-Euk Lee, and Jong-Sung Kim, “Human-machine interface for wheelchair control with EMG and its evaluation,” in *Engineering in Medicine and Biology Society*, 2003, vol. 2, pp. 1602–1605.

[6] I. Moon, M. Lee, J. Chu, and M. Mun, “Wearable EMG-based HCI for electric-powered wheelchair users with motor disabilities,” in *IEEE International Conference on Robotics and Automation*, 2005, pp. 2649–2654.

[7] B. Rebsamen, E. Burdet, C. Guan, H. Zhang, C. L. Teo, Q. Zeng, M. Ang, and C. Laugier, “A brain-controlled wheelchair based on P300 and path guidance,” in *IEEE/RAS International Conference on Biomedical Robotics and Biomechatronics*, 2006, pp. 1101–1106.

[8] K. Choi and A. Cichocki, “Control of a wheelchair by motor imagery in real time,” in *International Conference on Intelligent Data Engineering and Automated Learning*, 2008, pp. 330–337.

[9] B. Peerdeman, D. Boere, H. Witteveen, R. H. in ’t Veld, H. Hermens, S. Stramigioli, H. Rietman, P. Veltink, and S. Misra, “Myoelectric forearm prostheses: state of the art from a user-centered perspective.,” *J. Rehabil. Res. Dev.*, vol. 48, no. 6, pp. 719–738, 2011.

[10] M. Ortiz-Catalan, B. Hakansson, and R. Branemark, “Real-time and simultaneous control of artificial limbs based on pattern recognition algorithms,” *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 22, no. 4, pp. 756–764, 2014.

[11] S. Muceli and D. Farina, “Simultaneous and proportional estimation of hand kinematics from EMG during mirrored movements at multiple degrees-of-freedom,” *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 20, no. 3, pp. 371–378, 2012.

[12] Q. Li, W. Chen, and J. Wang, “Dynamic shared control for human-wheelchair cooperation,” in *IEEE International Conference on Robotics and Automation*, 2011, pp. 4278–4283.

[13] L. Tonin, T. Carlson, R. Leeb, and J. Del R. Millan, “Brain-controlled telepresence robot by motor-disabled people,” in *IEEE International Conference on Engineering in Medicine and Biology*, 2011, pp. 4227–4230.

[14] T. Carlson and J. Del R. Millan, “Brain-controlled wheelchairs: A robotic architecture,” *IEEE Robot. Autom. Mag.*, vol. 20, no. 1, pp. 65–73, 2013.

[15] D. A. Sanders, “Using Self-Reliance Factors to Decide How to Share Control Between Human Powered Wheelchair Drivers and Ultrasonic Sensors,” *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 25, no. 8, pp. 1221–1229, 2017.

[16] Y. Li, K. P. Tee, W. L. Chan, R. Yan, Y. Chua, and D. K. Limbu, “Continuous Role Adaptation for Human Robot Shared Control,” *IEEE Trans. Robot.*, vol. 31, no. 3, pp. 672–681, 2015.

[17] C. S. L. Tsui, P. Jia, J. Q. Gan, O. Hu, and K. Yuan, “EMG-based hands-free wheelchair control with EOG attention shift detection,” in *IEEE International Conference on Robotics and Biomimetics*, 2007, pp. 1266–1271.

[18] A. Naber, “SharedControl,” *Github Repository*, 2018. [Online]. Available: https://github.com/biopatrec/sharedcontrol.

[19] M. Ortiz-Catalan, “BioPatRec,” *Github Repository*, 2017. [Online]. Available: https://github.com/biopatrec/biopatrec.

[20] E. Mastinu, B. Hakansson, and M. Ortiz-Catalan, “Low-cost, open source bioelectric signal acquisition system,” in *IEEE International Conference on Wearable and Implantable Body Sensor Networks*, 2017, pp. 19–22.

[21] S. A. Bowyer, B. L. Davies, and F. Rodriguez Y Baena, “Active constraints/virtual fixtures: A survey,” *IEEE Trans. Robot.*, vol. 30, no. 1, pp. 138–157, 2014.

[22] E. Bicho and G. Schöner, “The dynamic approach to autonomous robotics demonstrated on a low-level vehicle platform,” *Rob. Auton. Syst.*, vol. 21, no. 1, pp. 23–35, 1997.

[23] B. Hudgins, P. Parker, and R. N. Scott, “The Recognition Of Myoelectric Patterns For Prosthetic Limb Control,” in *IEEE International Conference on Engineering in Medicine and Biology*, 1991, pp. 2040–2041.

[24] M. Ortiz-Catalan, R. Brånemark, and B. Håkansson, “Biologically inspired algorithms applied to prosthetic control,” in *IASTED International Conference, Biomedical Engineering*, 2012, pp. 7–15.

[25] K. Englehart and B. Hudgins, “A robust, real-time control scheme for multifunction myoelectric control,” *IEEE Trans. Biomed. Eng.*, vol. 50, no. 7, pp. 848–54, 2003.

[26] A. M. Simon, L. J. Hargrove, B. A. Lock, and T. A. Kuiken, “A decision-based velocity ramp for minimizing the effect of misclassifications during real-time pattern recognition control,” *IEEE Trans. Biomed. Eng.*, vol. 58, no. 8, pp. 2360–2368, 2011.