This is the author's version of the article that has been published by ASME in the Proceedings of the 2020 Design of Medical Devices Conference, Minneapolis, MN, USA.

Proceedings of the 2020 Design of Medical Devices Conference
DMD2020
April 6, 7-9, 2020, Minneapolis, MN, USA

DMD2020-9009

A FRAMEWORK FOR MAPPING AND CONTROLLING EXOSKELETON GAIT PATTERNS IN BOTH SIMULATION AND REAL-WORLD

Lowell Rose[1], Michael C. F. Bazzocchi[1], Connal de Souza[1], Julie Vaughan-Graham [3], Kara Patterson[3], Goldie Nejat[1,2]

1. Autonomous Systems and Biomechatronics Lab, Department of Mechanical and Industrial Engineering, University of Toronto, Toronto, ON, Canada
2. Toronto Rehabilitation Institute, Toronto, ON, Canada
3. Department of Physical Therapy, University of Toronto, Toronto, ON, Canada

ABSTRACT

Stroke is a leading cause of disability, and robotic lower body exoskeletons have been developed to aid in gait rehabilitation. The simulation modeling and testing processes are often developed and deployed separately. This introduces additional steps which can hinder on-the-fly customization of gait patterns required for individualized gait rehabilitation. In this paper, we present a centralized control architecture which integrates both the simulated model and the exoskeleton hardware for lower body exoskeletons. The architecture allows for ease of simulating, adapting, and deploying gait patterns on an exoskeleton for use in gait rehabilitation, and allows for the on-the-fly customization and verification of gait patterns by physiotherapists during rehabilitation. Experiments validate the use of our overall control architecture to both model and control a physical exoskeleton, while following desired gait patterns.

1. INTRODUCTION

In recent years, lower body exoskeletons have been successfully employed for gait rehabilitation of persons post-stroke and spinal cord injury [1, 2, 3, 4]. Persons post-stroke often experience reduced motor control, primarily on one side of their body and as a result may develop atypical and compensatory motor behavior affecting their overall gait pattern [5]. Improving motor performance and thus function is a key goal of post-stroke rehabilitation. Early studies have found that exoskeletons are effective tools for over ground gait training [6], providing an opportunity for increased practice of more typical movement patterns thereby enhancing motor learning and movement recovery whilst simultaneously reducing atypical and compensatory motor behavior.

Gait rehabilitation with an exoskeleton provides repetition, intensity of practice and task specificity, which are critical elements of an interventi thought to encourage positive neuroplastic change. This in turn optimizes motor learning and movement recovery [6]. Post-stroke gait rehabilitation with exoskeletons has been found to consistently improve mobility and function [7].

The majority of rehabilitation-focused lower body exoskeletons, available commercially or for research, are limited to actuated hip and knee joints, with a passive or spring activated ankle joint [8]. However, physiotherapist recommendations and post-stroke rehabilitation studies have found that an active ankle joint is an important aspect of gait rehabilitation to address foot-drop and propulsion issues [9]. The H2 exoskeleton developed by Technaid S.L [10], is one such exoskeleton that consists of an actuated ankle joint, in addition to actuated hip and knee joints. The H2 is thus capable of deploying a fully actuated gait pattern during rehabilitation.

For lower body exoskeletons, models of the exoskeleton devices are often generated in simulation in order to validate various control schemes [11, 12] prior to implementation on the physical device. Software packages such as OpenSim [13] and Adams [16] have been used to simulate exoskeletons, but cannot be used to control physical hardware. In order to directly implement the simulated control strategies on hardware platforms, often new software tools to generate or transfer a model to hardware must be employed. Otherwise, if not transferred, the simulated development efforts are often redundantly duplicated on the hardware control platform, which can delay the testing and validation of an exoskeleton or custom gait patterns required for rehabilitation.

In this paper, we present the development of a unique control architecture for an exoskeleton, that seamlessly integrates the software needed for simulation and hardware implementation. The novel control architecture has two main advantages over other systems developed for lower body exoskeletons:

[1] Contact author: lowell.rose@mail.utoronto.ca

(i) The control architecture provides a centralized software structure (containing control schemes and gait patterns) for both the simulated model and the exoskeleton hardware, therefore only a single software structure needs to be adapted in order to customize gait patterns for patients. This is an advantage over other techniques used for lower body exoskeletons that require replication of software in separate simulation and hardware control architectures [13, 14, 16]. In our unified control architecture, desired gait patterns or control parameters only need to be inputted/updated once in order to customize the trajectory in both simulation and hardware, thus facilitating on-the-fly testing, adaptation, and implementation in clinical settings.

(ii) The hardware control is not reliant on the execution of the simulation. Namely, other control approaches using simulators such as Simulink [15, 22] require the output of the simulation for control of the hardware. The simulations' speed depend on model complexity and computational power [17], and therefore if the deployed model is too slow to meet execution frequency, control commands (e.g., joint angle positions) can be missed, resulting in incorrect hardware execution. Such processing delays and synchronization problems can lead to instabilities [18]. This can be problematic in a clinical setting, where incorrect motion can result in patient injury [4, 19].

1.1 Related Work

Lower body exoskeleton modeling has generally focused on determining the dynamics and forces present when using an exoskeleton [13, 14, 16, 20, 21], or for evaluating a specific controller design [11, 15, 22, 23] in simulation. Simulation software packages that have been used include AnyBody Modeling System (AMS) [14], SimMechanics [11, 21], Adams [16], Simulink [15, 22], and OpenSim [13, 23].

For example, in [14], the ARKE exoskeleton was simulated with a human musculoskeletal model in AMS software, focusing on developing a ground reaction force based control method. In [11, 21], SimMechanics was used to generate a dynamic model and controller for an exoskeleton in simulation, in order to test a novel neural network based controller. In [16], the multibody dynamics simulation software, Adams, was used to model an exoskeleton and verify the proposed design through comparing simulated and desired joint angles, as well as interaction forces between the exoskeleton and a musculoskeletal human model.

In [13], a model consisting of the kinematic and dynamic properties of an existing exoskeleton was generated in OpenSim to verify the kinematic design and actuator performance. A musculoskeletal human model was able to be accurately represented to investigate the interaction between the user and the device, and actuator torques were found. A modification to this exoskeleton design was proposed, and a new dynamic model was tested in simulation. In [23], the robot operating system (ROS) was used with the H2 exoskeleton to obtain the online estimation of a patient's muscle forces through electromyography (EMG). A musculoskeletal model was then simulated in OpenSim using this data.

The aforementioned work has focused on the modeling and control solely in simulations. Alternatively in [15], a model of an exoskeleton was simulated using Simulink and converted with the QUARC software into a representation that can be transferred onto a physical exoskeleton. In [22], a similar approach was conceptually presented. However, the use of a Simulink based method has potential drawbacks such as limited real-time performance [24], dropped commands [17], and instabilities [18].

The combination of ROS and Gazebo for control and simulation is one that has been used on other hardware platforms such as mobile robots and manipulator arms [25, 26]. For manipulators, ROS and Gazebo have been used to perform trajectory planning, manipulation and grasping control, and for mobile robots to simulate autonomous navigation and 3D-mapping, as well as enable experiments in a physical environment. This combination allows for the ease of prototyping, validation and testing by allowing for the consolidation of simulation and hardware control software, and would be a useful tool to also apply to the development of exoskeletons, which has not yet been explored. The use of exoskeleton devices presents challenges related to the interaction of a patient and the device, where safety and comfort are primary concerns, and simulations are vital to validating an appropriate gait pattern on-the-fly in a physiotherapy session.

In this paper, we present a novel centralized control architecture that allows a desired gait pattern to be controlled and adjusted on-the-fly both in simulation and on a real exoskeleton. This will allow for customization and validation directly during gait rehabilitation sessions, thereby optimizing the time spent in rehabilitation. We achieve this by using ROS and a Gazebo simulator.

2. CONTROL ARCHITECTURE

The overall control architecture we have developed is presented in Fig. 1. The input to the control architecture is the desired gait pattern, provided by OpenSim's 2354 gait model [27], Fig. 2. The architecture consists of the following main modules: 1) the Joint Publisher Node, which is responsible for sending joint position commands, 2) the Simulation Platform, which models and controls the exoskeleton in a simulated environment, 3) the Communication Interface, which consists of the development board used to communicate joint angle messages to the exoskeleton, and 4) the Exoskeleton module, which executes control commands on the physical device. These modules are discussed in more detail below. In this work, we utilize the H2 exoskeleton for implementation and testing of our control architecture.

2.1 Joint Publisher Node

This Joint Publisher Node is responsible for providing the control commands to the entire architecture. This module contains the gait pattern from OpenSim's 2354 gait model [27], which is represented as a series of joint angles. The Joint Publisher Node reads the joint angles and establishes each of the joint position topics. It then publishes the joint position

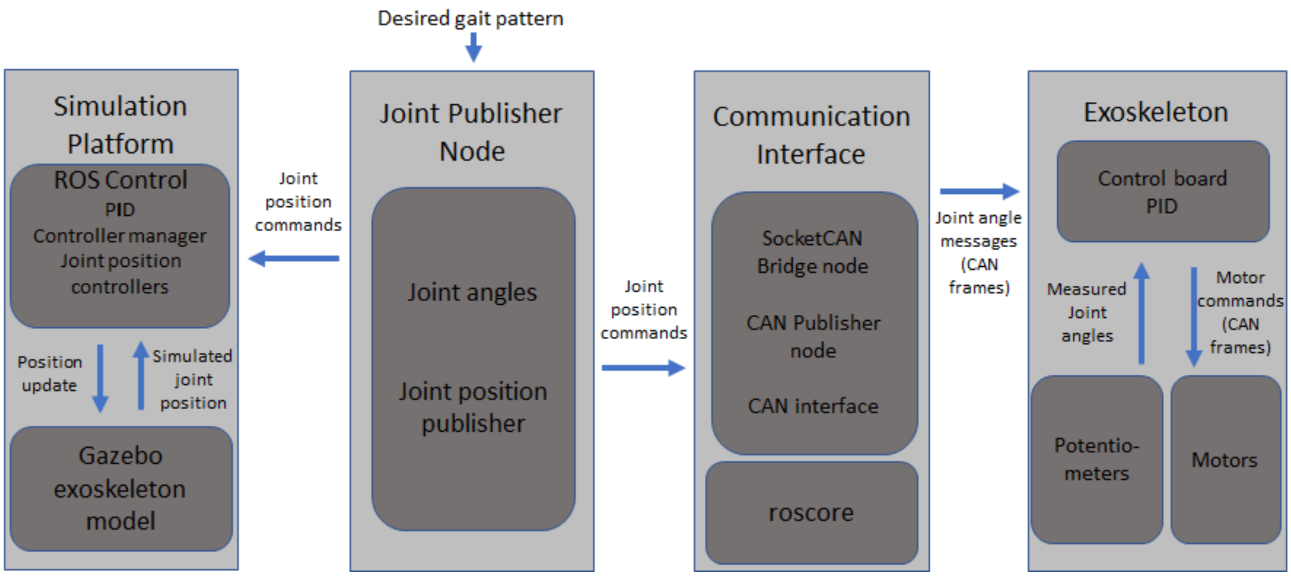

FIGURE 1: THE OVERALL PROPOSED ARCHITECTURE FOR THE CONTROL OF THE SIMULATION AND HARDWARE DEVICES (EXOSKELETON, COMMUNICATION INTERFACE).

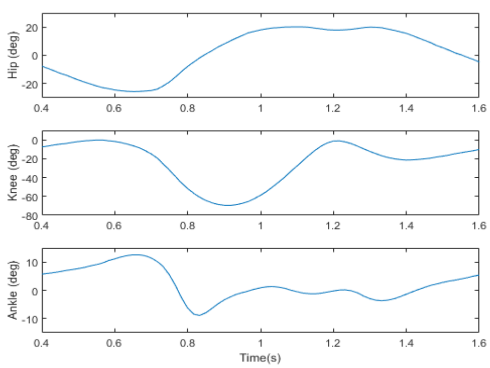

FIGURE 2: DESIRED GAIT PATTERN BASED ON OPENSIM'S 2354 MODEL [27].

commands for each joint to the Simulation Platform and the Communication Interface modules.

2.2 Simulation Platform

To simulate the exoskeleton, a combination of a ROS Control [28] node and Gazebo [29] is used. ROS Control contains a controller manager to subscribe to each joint position topic, a joint position interface to store the simulated joint positions, and also contains a PID controller. We have designed the PID controller based on the H2 exoskeleton's built-in controller [32] with matching PID gains. The controller is used to obtain and hold joint positions.

The joint position commands published from the Joint Publisher Node are sent to the position controller of each joint in Gazebo, to simulate the motion of the exoskeleton. The simulated joint position data can be saved with the ROS command, *rosbag,* for analysis and comparison.

As a first step in simulating the dynamics of the exoskeleton, the March IV model in URDF format [30] was extended to represent the physical configuration, dimensions and specifications of the H2, Table I. The 3D simulated model of the exoskeleton is presented in Fig. 3.

2.3 Communication Interface

To enable control on the exoskeleton, a CAN (Controller Area Network) Bus interface is necessary to send the joint position commands from the Joint Publisher Node to the on-board microcontroller, in the form of joint angle messages (CAN frames). The CAN bus is a message-based protocol that allows messages to be sent and received to and from all nodes in a system [31]. The control board on the H2 has the capability to receive CAN frames from an external source and send them to the motors. The CAN protocol and message types that can be sent and received were developed by Technaid S.L. and discussed in [32].

Table I: SPECIFICATIONS OF THE SIMULATED EXOSKELETON.

Link	**Thigh**	**Shank**	**Foot**
Length [m]	0.40	0.40	0.23
Mass [kg]	1.00	1.00	0.5
I_{xx} [kg m^2]*	0.034	0.034	0.013
I_{yy} [kg m^2]	0.034	0.034	0.013
I_{zz} [kg m^2]	0.042	0.042	0.021
Joints Angles	**Hip**	**Knee**	**Ankle**
Maximum [°]	105	105	20
Minimum [°]	-15	-15	-15

*The I_{xx}, I_{yy}, and I_{zz}, are the diagonal terms of the moment of inertia tensor, with the off-diagonal terms equaling to zero.

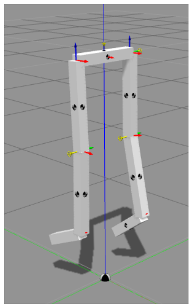

FIGURE 3: SIMULATED 3D MODEL OF THE EXOSKELETON.

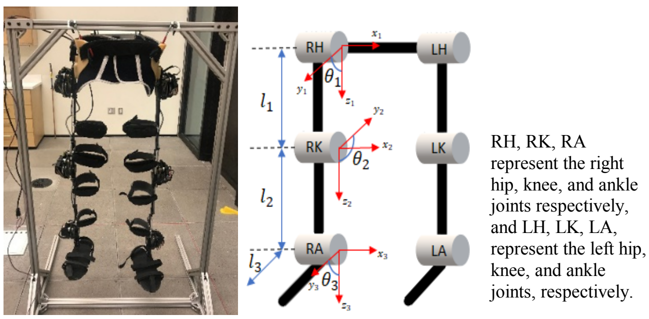

FIGURE 4: THE H2 EXOSKELETON AND ITS ASSOCIATED DYNAMICS MODEL.

To enable CAN communication, a BeagleBone Black (BBB) wireless development board [33] is used. The BBB has 512MB DDR3 RAM, 4GB 8-bit eMMC on-board flash storage with expandable micro SD support, 802.11b/g/n and Bluetooth 4.1 plus BLE. The BBB also hosts the roscore, which executes the master program used for communication between the ROS nodes in Fig. 1.

To create the CAN interface necessary for sending messages to the exoskeleton control board, a package called SocketCAN [34] is used. SocketCAN is an open source collection of drivers for Linux, used for establishing a CAN socket. This software allows a CAN port to be enabled by creating a socket similar to the TCP/IP protocol, and binds the socket to an interface in the software [34]. To send CAN frames over the network, a CAN transceiver is needed to interface between the socket and the exoskeleton CAN input [31]. An add-on board for the BBB from Waveshare containing a CAN transceiver [35] was used for this purpose.

To send the published joint angles to the CAN network, the CAN Publisher node is established. This node receives the joint position commands from the Joint Publisher Node and translates it into a CAN frame message format. On the BBB, a ROS node, SocketCAN Bridge [36], is used to establish a socket and ROS message format. The CAN Publisher node then sends joint angle messages (in the form of CAN frames) to the exoskeleton control board.

2.4 Exoskeleton

The H2 exoskeleton, Fig. 4, is a lower-limb 6 degree of freedom (DOF) device, for use in over-ground gait training with post-stroke patients [6]. This device has active hip, knee, and ankle actuation controlled by brushless DC motors with attached harmonic drive gear boxes. The device also has sensors for joint position (potentiometers), joint interaction torque, and foot/ground contact (heel and toe) [32].

FIGURE 5: THE EXPERIMENTAL SETUP FOR THE EXOSKELETON TRIAL WITH A USER WEARING THE DEVICE.

The associated dynamics model for the exoskeleton is also presented in Fig. 4. Each leg is composed of 3 revolute joints, linked by equally sized linkages. In its current configuration these have a length l_1, l_2, of 0.4 m and represent the thigh and shank linkages. The foot linkage has a length l_3 equal to 0.23 m. The three joint angles, i.e., $\theta_1, \theta_2, \theta_3$, are the active hip, knee, and ankle joints, respectively; where θ_1 is the angle from y_1 to z_1, θ_2, is the angle from y_2 to z_2, and θ_3 is the angle from y_3 to z_3. The zero-degree state for these motors is aligned with the *z*-plane for the thigh and shank, and aligned with the *y*-plane for the foot.

The on-board microcontroller on the H2 receives the CAN frames and sends them to the motors as motor commands for execution. The board also contains a PID controller to regulate the position control on the exoskeleton. The H2 includes onboard potentiometers for each motor to measure joint angles. As the measured joint angle data is being sent over the CAN network, it is read by the CAN transceiver on the BBB and is stored with SocketCAN's *candump* function.

3. EXPERIMENTS

We conducted experiments utilizing our control architecture to obtain and compare joint angles for both the simulated and physical exoskeleton. A participant wearing the exoskeleton, as seen in Fig. 5, repeatedly walked along a straight line for at least 5 gait cycles. The gait cycles were recorded to evaluate the real-time performance of the exoskeleton within the control architecture. Then, a new gait pattern provided by a physiotherapist was implemented on-the-fly during the experiment, and gait cycles were again recorded.

The joint position data (with timestamps) from the potentiometers on the exoskeleton (captured using the *candump* command) was compared to the input joint position commands and the simulated joint angles. The latter are obtained from Gazebo using the *rosbag* function.

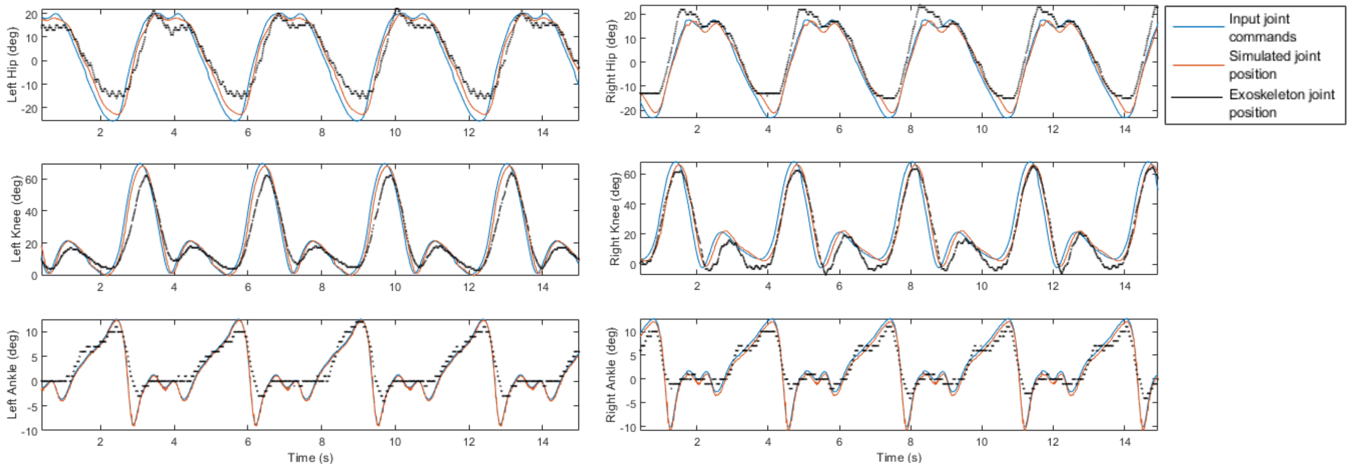

FIGURE 6: RESULTS FOR THE EXPERIMENTS WITH THE EXOSKELETON WORN BY A USER WITH THE GAIT PATTERN DERIVED FROM OPENSIM.

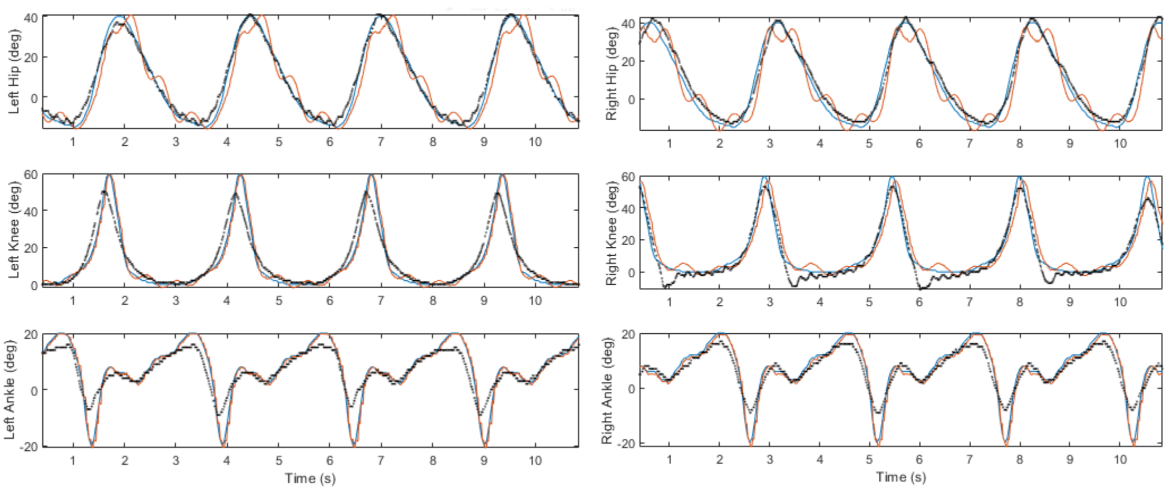

FIGURE 7: RESULTS FOR THE EXPERIMENTS WITH THE EXOSKELETON WORN BY A USER WITH AN ADJUSTED GAIT PATTERN RECOMMENDED BY A PHYSIOTHERAPIST.

3.1 Results and Discussion

The results from the exoskeleton test with a participant wearing the exoskeleton for both the left and right joints are presented in Figs. 6 and 7. When compared to the input commands, both the simulated joint angles and exoskeleton joint angles follow the desired gait pattern closely. This is especially true with the hip and knee joints. However, the exoskeleton did not reach the minimum joint angles for both the ankle and hip for the gait pattern in Fig. 6 and the gait pattern in Fig. 7. This is due to the physical joint limitations that are inherent in the design of the exoskeleton hardware. Some discrepancies observed (i.e., Fig. 7 right knee) are due to noisy potentiometer readings and slight differences in the simulation and hardware execution speeds. A video of the simulated and experimentally implemented gait patterns utilizing the proposed architecture are presented on our YouTube channel here.

4. CONCLUSION

In this work, a centralized control architecture was developed utilizing Gazebo and ROS for lower body exoskeleton simulation and physical hardware control. The control architecture consists of only a single software structure to customize gait patterns for patients. The simulation model can be visualized in real-time, allowing for the modification of gait during rehabilitation. Namely, a new gait pattern can be easily and quickly simulated, verified and controlled on the exoskeleton. This is an important feature for physiotherapists in clinical sessions as it will optimize time spent on rehabilitation. Experiments showed that our overall control architecture was able to both model and control a physical exoskeleton, while following desired gait patterns. Future work will build upon the developed control architecture by designing predictive controllers and individualized gait patterns with the aim of improving adaptability and training with the exoskeleton for persons post-stroke.

ACKNOWLEDGEMENTS

This research was funded through the EMHSeed funding from the University of Toronto and the Canada Research Chairs Program. The authors would like to thank Guillermo Asín-Prieto, José L. Pons, Juan C. Moreno, Javier O. Roa Romero and Technaid S.L. for their assistance in supplying and providing technical support for the H2. They would also like to thank Dina Brooks for her assistance in consideration of the clinical outcomes.

REFERENCES

[1] Louie, Dennis R., and Eng, Janice J. "Powered Robotic Exoskeletons in Post-Stroke Rehabilitation of Gait: A Scoping Review." *Journal of NeuroEngineering and Rehabilitation* Vol. 13 No. 1 (2016): pp. 53. DOI 10.1186/s12984-016-0162-5.

[2] Pazzaglia, Mariella, and Molinari, Marco. "The Embodiment of Assistive Devices-from Wheelchair to Exoskeleton." *Physics of Life Reviews* Vol. 16 (2016): pp. 163-175. DOI 10.1016/j.plrev.2015.11.006.

[3] López-Larraz, Eduardo, Trincado-Alonso, Fernando, Rajasekaran, Vijaykumar, Pérez-Nombela, Soraya, del-Ama, Antonio J., Aranda, Joan, Minguez, Javier, Gil-Agudo, Angel, and Montesano, Luis. "Control of an Ambulatory Exoskeleton with a Brain-Machine Interface for Spinal Cord Injury Gait Rehabilitation." *Frontiers in Neuroscience* Vol. 10 (2016): pp. 359. DOI 10.3389/fnins.2016.00359.

[4] Contreras-Vidal, Jose L., Bhagat, Nikunj A., Brantley, Justin, Cruz-Garza, Jesus G., He, Yongtian, Manley, Quinn, Nakagome, Sho, Nathan, Kevin, Tan, Su H., Zhu, Fangshi, and Pons, Jose L. "Powered Exoskeletons for Bipedal Locomotion after Spinal Cord Injury." *Journal of Neural Engineering* Vol. 13 No. 3 (2016). DOI 10.1088/1741-2560/13/3/031001.

[5] Murray, Spencer A., Ha, Kevin H., Hartigan, Clare, and Goldfarb, Michael. "An Assistive Control Approach for a Lower-Limb Exoskeleton to Facilitate Recovery of Walking Following Stroke." *IEEE Transactions on Neural Systems and Rehabilitation Engineering* Vol. 23 No. 3 (2015): pp. 441–449. DOI 10.1109/TNSRE.2014.2346193.

[6] Bortole, Magdo, Venkatakrishnan, Anusha, Zhu, Fangshi, Moreno, Juan C, Francisco, Gerard E, Pons, Jose L, and Contreras-Vidal, Jose L. "The H2 Robotic Exoskeleton for Gait Rehabilitation after Stroke: Early Findings from a Clinical Study." *Journal of NeuroEngineering and Rehabilitation* Vol. 12 No. 1 (2015): pp. 54. DOI 10.1186/s12984-015-0048-y.

[7] Federici, Stefano, Meloni, Fabio, Bracalenti, Marco, and Filippis, Maria Laura De."The Effectiveness of Powered, Active Lower Limb Exoskeletons in Neurorehabilitation: A Systematic Review." *NeuroRehabilitation* Vol. 37 No. 3 (2015): pp. 321-340. DOI 10.3233/NRE-151265.

[8] Young, Aaron J., and Ferris, Daniel P. "State of the Art and Future Directions for Lower Limb Robotic Exoskeletons." *IEEE Transactions on Neural Systems and Rehabilitation Engineering* Vol. 25 No. 2 (2017): pp. 171–182. DOI 10.1109/TNSRE.2016.2521160.

[9] Blaya, Joaquin A., and Herr, Hugh. "Adaptive Control of a Variable-Impedance Ankle-Foot Orthosis to Assist Drop-Foot Gait." *IEEE Transactions on Neural Systems and Rehabilitation Engineering* Vol. 12 No. 1 (2004): pp. 24–31. DOI 10.1109/TNSRE.2003.823266.

[10] "Exo H2," Technaid S.L. Accessed 2019, www.technaid.com/products/robotic-exoskeleton-exo-exoesqueleto/.

[11] Zhang, Xinyi, Wang, Haoping, Tian, Yang, Peyrodie, Laurent, and Wang, Xikun. "Model-Free Based Neural Network Control with Time-Delay Estimation for Lower Extremity Exoskeleton." *Neurocomputing* Vol. 272 (2018): pp. 178–188. DOI 10.1016/j.neucom.2017.06.055.

[12] Aftab, Zohaib, and Ali, Asad. "Simulating a Wearable Lower-Body Exoskeleton Device for Torque and Power Estimation." *Proceedings of the IEEE-RAS International Conference on Humanoid Robots.* pp. 412–417. Birmingham, UK, November 15-17, 2017. DOI 10.1109/HUMANOIDS.2017.8246906.

[13] Ferrati, Francesco, Bortoletto, Roberto, and Pagello, Enrico. "Virtual Modelling of a Real Exoskeleton Constrained to a Human Musculoskeletal Model." *Lecture Notes in Computer Science* Vol. 8064 (2013): pp. 96–107. DOI 10.1007/978-3-642-39802-5-9.

[14] Fournier, Brandon N., Lemaire, Edward D., Smith, Andrew J.J., and Doumit, Marc. "Modeling and Simulation of a Lower Extremity Powered Exoskeleton." *IEEE Transactions on Neural Systems and Rehabilitation Engineering* Vol. 26 No. 8 (2018): pp. 1596–1603. DOI 10.1109/TNSRE.2018.2854605.

[15] Asl, Hamed Jabbari, Narikiyo, Tatsuo, and Kawanishi, Michihiro. "Neural Network Velocity Field Control of Robotic Exoskeletons with Bounded Input." *Proceedings of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics, AIM*. pp.1363–1368. Munich, Germany, July 3-7, 2017. DOI 10.1109/AIM.2017.8014208.

[16] Santos, Danilo G, and Siqueira, Adriano A. G. "ADAMS / Matlab Co-Simulation of an Exoskeleton for Lower Limbs." *Proceedings of the 20th International Congress of Mechanical Engineering*. Gramado, Brazil, November 15-20, 2009.

[17] "Simulink and ROS Interaction." MathWorks. Accessed 2019, www.mathworks.com/help/ros/ug/simulink-and-ros-interaction.html.

[18] Budai, Csaba and Kovacs, Laszlo L. "Limitations Caused by Sampling and Quantization in Position Control of a Single Axis Robot," *XV International PhD Workshop*. OWD 2013, 19–22 October 2013.

[19] Torricelli, Diego, Cortés, Camilo, Lete, Nerea, Bertelsen, Álvaro, Gonzalez-Vargas, Jose E., del-Ama, Antonio J., Dimbwadyo, Iris, Moreno, Juan C., Florez, Julian, and Pons, Jose L. "A Subject-Specific Kinematic Model to Predict Human Motion in Exoskeleton-Assisted Gait." *Frontiers in Neurorobotics* Vol. 12 (2018): pp. 18. DOI 10.3389/fnbot.2018.00018.

[20] Copilusi, Cristian, Ceccarelli, Marco, Dumitru, N., and Carbone, Giuseppe. "Design and Simulation of a Leg Exoskeleton Linkage for a Human Rehabilitation System." *Mechanisms and Machine Science* Vol 17 (2014): pp. 117–125. DOI 10.1007/978-3-319-01845-4_12.

[21] Winder, Samuel B., and Esposito, Joel M. "Modeling and Control of an Upper-Body Exoskeleton." *Proceedings of the Annual Southeastern Symposium on System Theory*. pp. 263–68. New Orleans, LA, March 16-18, 2008. DOI 10.1109/SSST.2008.4480234.

[22] Dinh, Binh Khanh, Cappello, Leonardo, and Masia, Lorenzo. "Localized Extreme Learning Machine for Online Inverse Dynamic Model Estimation in Soft Wearable Exoskeleton." *Proceedings of the IEEE RAS and EMBS International Conference on Biomedical Robotics and Biomechatronics*. pp. 580–87. UTown, Singapore, June 26-29, 2016. DOI 10.1109/BIOROB.2016.7523688.

[23] Ceseracciu, Elena, Mantoan, Alice, Bassa, Marco Matteo, Moreno, Juan C., Pons, Jose L., Prieto, Guillermo Asin, del-Ama, Antonio J., Marquez-Sanchez, Ester, Gil-Agudo, Angel, Pizzolato, Claudio, Lloyd, David G., and Reggiani, Monica. "A Flexible Architecture to Enhance Wearable Robots: Integration of EMG-Informed Models." *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*. pp. 4368–4374. Hamburg, Germany, Sept 28 - Oct 2, 2015. DOI 10.1109/IROS.2015.7353997.

[24] "QUARC Win64 Target." Quanser, 2019. Accessed 2019, http://quanser-update.azurewebsites.net/quarc/documentation/quarc_win64_target.html.

[25] Takaya, Kenta, Asai, Toshinori, Kroumov, Valeri, and Smarandache, Florentin. "Simulation Environment for Mobile Robots Testing Using ROS and Gazebo." *Proceedings of the 20th International Conference on System Theory, Control and Computing*. pp. 96–101. Sinaia, Romania, October 13-15, 2016. DOI 10.1109/ICSTCC.2016.7790647.

[26] Qian, Wei, Xia, Zeyang, Xiong, Jing, Gan, Yangzhou, Guo, Yangchao, Weng, Shaokui, Deng, Hao, Hu, Ying, and Zhang, Jianwei. "Manipulation Task Simulation Using ROS and Gazebo." *Proceedings of the 2014 IEEE International Conference on Robotics and Biomimetics*. pp. 2594–98. Bali, Indonesia, December 5-10, 2014. DOI 10.1109/ROBIO.2014.7090732.

[27] Seth, Ajay, Anderson, Frank C., and Delp, Scott L. "Gait 2392 and 2354 Models." Accessed 2019, https://simtk-confluence.stanford.edu/display/OpenSim/Gait+2392+and+2354+Models

[28] "ROS Control." Open Source Robotics Foundation, 2014. Accessed 2019, http://gazebosim.org/tutorials/?tut=ros_control.

[29] Coleman, Dave. "Gazebo ROS Demos," 2017. Accessed 2019, https://github.com/ros-simulation/gazebo_ros_demos.

[30] "MARCH IV Exoskeleton." Project MARCH, TU Delft, 2019. Accessed 2019, https://github.com/project-march/march-iv.

[31] Corrigan, Steve. "Introduction to the Controller Area Network (CAN)," 2016. Accessed 2019, http://www.ti.com/lit/an/sloa101b/sloa101b.pdf.

[32] Bassa, Marco Matteo. "Development Of The Communication System For A Lower Limb Human Hexoskeleton Using The Ros Middleware." Master's Thesis. University of Padua, Padua, Italy. 2015.

[33] "BeagleBone Black Wireless." Beagleboard, 2019. Accessed 2019, https://beagleboard.org/black-wireless.

[34] Hartkopp, Oliver and Kleine-Budde, Marc. "Linux-CAN / SocketCAN user space applications," 2019. Accessed 2019, https://github.com/linux-can.

[35] "RS485 CAN CAPE," Waveshare, 2016. Accessed 2019, https://www.waveshare.com/wiki/RS485_CAN_CAPE.

[36] "socketcan_bridge," Open Source Robotics Foundation, 2016. Accessed 2019, http://wiki.ros.org/socketcan_bridge.