Design, Characterization, and Dynamic Simulation of the MAHI Open Exoskeleton Upper Limb Robot

Nathan Dunkelberger, *Graduate Student Member, IEEE*, Jeffrey Berning, Kevin J. Dix, *Student Member, IEEE*, Samuel A. Ramirez, *Student Member, IEEE*, and Marcia K. O'Malley, *Fellow, IEEE*

***Abstract*—Stroke and spinal cord injury are becoming ever more prevalent in the United States. Recent research has shown that rehabilitation robots have the potential to positively impact the rehabilitation process by providing a platform for repetition-based movement therapy. To advance the field, future research directions for robotic rehabilitation are focused on advanced model-based control algorithms, and the combination of robotics with cutting-edge neuromodulation technologies. These approaches necessitate devices that not only feature kinematic and dynamic properties well-suited for model-based control, but also require devices that allow for easy placement of sensors and electrodes on the limb when inserted in the robot. With these design goals in mind, we present the MAHI Open Exoskeleton, a four-degree-of-freedom robot with an open mechanical structure and simplified dynamics, combined with open-source software, that together lay the groundwork for advanced model-based control. The dynamic properties of each joint were characterized and compared against other recently developed rehabilitation robots. Open-source software was developed for the robot, which provides users with both low-level and application-level interfaces to implement a variety of control strategies. Dynamic equations were developed and implemented into a real-time simulation with a visualization, including a seamless interface to the developed software library. Impedance control and model predictive control were implemented and compared to the simulation, proving the value of the new designs.**

***Index Terms*—Design, mechanisms, medical and rehabilitation robotics, modeling and control, real-time and hardware-in-the-loop simulation, system identification.**

I. Introduction

RESEARCH has predicted that by 2030, 3.88% of the population older than 18 will have had a stroke [1]. In that same time frame, annual stroke-related medical costs will rise to $184.13 billion in the United States, up from $71.55 billion in 2012. Additionally, there are 291 000 people living with spinal cord injuries in the U.S. [2], with many of them requiring assistance when performing activities of daily living (ADLs). The pervasiveness of stroke and spinal cord injury (SCI), along with other conditions resulting in motor impairment, has spurred research into rehabilitative and assistive technologies that can be used to regain or supplement motor function following such injuries.

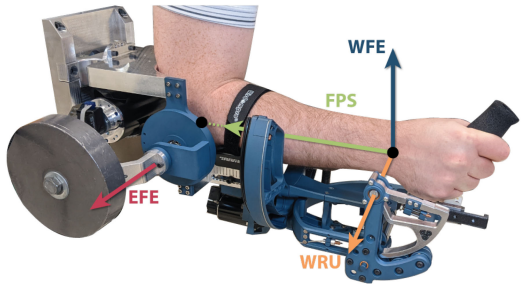

Fig. 1. MOE is shown with a user's arm placed for rehabilitation or assistive applications. The overlaid coordinate frames show the axes of rotation for each joint, with elbow flexion/Extension (EFE) in purple, forearm pronation/supination (FPS) in green, wrist flexion/extension (WFE) in blue, and wrist radial/ulnar deviation (WRU) in orange. The black dot on the elbow indicates where the FPS joint meets the EFE joint, and the black dot on the wrist indicates the interaction of the axes of rotation of the FPS, WFE, and WRU joints.

Using rehabilitation robots to provide repetition-based therapy has shown to be effective at restoring some upper limb motor function following a stroke or spinal cord injury [3], [4]. Robots have properties that make them ideal for the rehabilitation process, such as being able to perform repetitive motions in a very accurate and precise manner. Still, clinical outcomes of robot-mediated upper limb rehabilitation have not significantly surpassed what is achievable with traditional therapeutic interventions for individuals with stroke [5]. Further, few large-scale studies have been conducted for individuals with SCI, though robot-assisted interventions have been shown to be safe and feasible for this population [6].

Research has shown that it is important for the patient to be actively involved in rehabilitation activities, rather than being

Manuscript received January 21, 2022; revised March 11, 2022; accepted May 2, 2022. Recommended by Technical Editor R. Carloni and Senior Editor X. Chen. *(Corresponding author: Nathan Dunkelberger.)*

The authors are with the Mechatronics and Haptic Interfaces Lab, Department of Mechanical Engineering, Rice University, Houston, TX 77005 USA (e-mail: nbd2@rice.edu; jb139@rice.edu; kjd2@rice.edu; chromium.ramirez@gmail.com; omalleym@rice.edu).

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TMECH.2022.3175507.

Digital Object Identifier 10.1109/TMECH.2022.3175507

passively carried through motions by the robot [3], [7]. There have been several techniques implemented to promote participant engagement, including gamification of rehabilitation [8] and assist-as-needed control algorithms [9]. In these scenarios, it is useful, and sometimes necessary, to have an accurate dynamic model of the effort contributions of both the robot and the patient. Consider the approach of using neuromusculoskeletal modeling with myoelectrical sensors to understand how much effort the patient can provide, and to supplement the rest of the required effort through the robot [10]. In order to do this effectively, a model of both the human and the robot are necessary to truly benefit the rehabilitation process. Having a model of the robot is also useful for functional assistance by allowing for collaboration between the robot and other actuators, such as surface functional electrical stimulation so that the load can be efficiently shared [11].

Another promising future direction is to combine robotic rehabilitation and other interventions, such as neuromodulation [12]–[14]. Combination therapies by their very nature require the physical integration of hardware such as myoelectric sensors [15], neuromuscular electrical stimulation (NMES) electrodes [16], and brain machine interface hardware components [17], within the work volume of the robot. To be able to combine these technologies effectively, the robot should be designed such that the same areas of the body that are being rehabilitated are also accessible for the placement of other sensors, electrodes, and devices necessary for such combinatorial approaches to rehabilitation.

When developing new robotic hardware, dynamic models, and control algorithms, particularly during a global pandemic, computer simulations can be a valuable tool to enable research progress. Not only do simulation environments allow designers to develop and test experiment code and novel control algorithms remotely, they also provide a safe environment to develop models and algorithms without endangering the robot or a user. When designed well, the code and algorithms developed in the simulation environment can easily and quickly be ported to experimental hardware, streamlining the development process. Such processes also emphasize safety, since tuning control algorithms often involves significant trial and error to adjust gains and reconfigure target parameters to achieve the desired result.

In this article, we present a new kinematic design for an upper limb robotic exoskeleton, the MAHI open exoskeleton (MOE), designed to deliver rehabilitation following neurological injury. The robot's serial mechanism, described in Section II, is well-suited for therapeutic interventions that require additional sensors and electrodes to be affixed to the limb, given the ease of insertion of the limb in the robot. In Section III-A, we describe the methods and results of experimental characterization of MOE compared with other upper limb rehabilitation robots. Section III-B presents the explicit dynamic equations of MOE, along with a description of the real-time dynamic simulation environment that was created to facilitate our development process. The software implementation is presented in Section IV. An example control implementation showing standard impedance control and model predictive control (MPC) is described in Section V. We discuss the contributions of this article and future directions in Section VI. Finally, Section VII concludes this article.

II. MAHI Open Exoskeleton (MOE)

We present the design of MOE, a four-degree-of-freedom (DoF) exoskeleton robot for the upper limb. This device was developed for robotic rehabilitation interventions that require model-based control methods, or approaches that require the robot to work in combination with neuromodulation systems. Based on our experiences with prior robot designs like the MAHI Exo-II and OpenWrist, we identified the following design criteria to drive our development of MOE.

1) Mechanism presents minimal barriers when donning and doffing for users with motor impairment, and facilitates easy placement of sensors and electrodes necessary for combinatorial interventions.
2) Kinematic design enables computationally simple derivation of closed-form dynamic equations that can be used to develop a dynamic simulation environment to safely design and test novel control algorithms

A. Mechanical Design

MOE, shown in Fig. 1, is a serial 4-DoF upper limb exoskeleton robot consisting of four revolute joints that correspond to the four anatomical joints spanning the elbow through the wrist. The structure of MOE is open, allowing the robot to be donned simply by lowering the arm into the robot, rather than inserting the limb through circular components found in some exoskeletons [21]. All four joints are driven by capstan-cable drives, allowing for backdrivability while minimizing backlash. Range of motion and torque output are presented for MOE and other relevant robots in Table I.

The most proximal joint moves the elbow through the flexion and extension motion. The elbow of the user is placed in the middle of the joint with the capstan located on one side and a counterweight located on the other. The counterweight offsets the weight of the other joints of the exoskeleton and the arm of the user, easing the load on the elbow flexion/extension motor. The location of the counterweight can be adjusted with a discrete sliding mechanism. The second joint is attached to the elbow joint underneath the joint axis allowing for the arm to be lowered into the robot without having to maneuver around robot components. There is a discrete sliding mechanism between the first and second joints that accounts for varying forearm lengths. When the screws are loose, the joint slides to adjust for arm length and locks back into place by tightening the screws. A set of three screws lock the passive joint in place. The discrete locations of the screws that can be tightened allows for modeling of MOE without needing to accurately measure the location of the slider every time it is changed. The design of MOE's elbow joint is based heavily on that of the MAHI Exo-II [21].

The forearm pronation/supination joint rotates along a curvilinear rail. The user can easily place their arm into MOE through the opening in the rail, a significant improvement over our prior design [21] that required the user to maneuver the limb through a fully enclosed circular bearing. The axis of the forearm pronation/supination joint intersects the elbow flexion extension axis as well as the two wrist joint axes (see Fig. 1). The curvilinear rail design is based on the pronation/supination joint present in the OpenWrist [23].

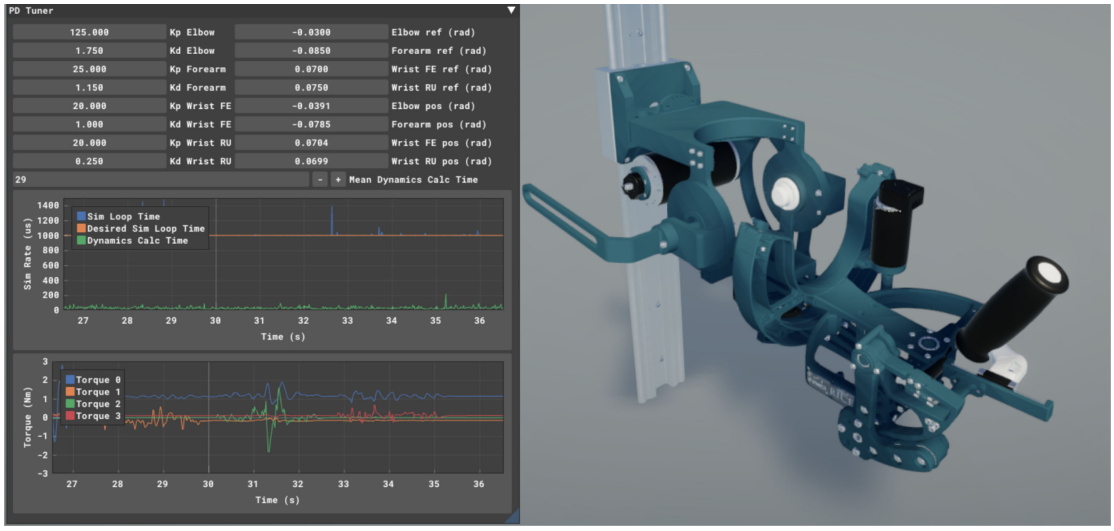

Fig. 2. MOE simulation provides an implementation of the dynamic equations, along with a user interface to visualize the output of the robot. In this use case, a GUI is being used to tune the PD gains for impedance control on each of the joints while tracking setpoints input to the GUI.

TABLE I
MOE CAPABILITIES COMPARED WITH ADL REQUIREMENTS AND OTHER REHABILITATION DEVICES (MIT-MANUS [18], IIT WRIST ROBOT [19], WRIST GIMBAL [20], MAHI EXO-II [21], RICEWRIST-S [22], AND OPENWRIST [23])

	Range of Motion [deg]								**Max continuous torque** [Nm]							
Joint	ADL	MIT	IIT	WG	ME-II	RW-S	OW	**MOE**	ADL	MIT	IIT	WG	ME-II	RW-S	OW	**MOE**
EFE	150	-	-	-	90	-	-	**102**	3.5	-	-	-	7.35	-	-	**6.59**
FPS	150	140	160	180	180	180	170	**180**	0.06	1.85	2.77	2.87	2.75	1.69	3.50	**3.55**
WFE	115	120	144	180	65	130	135	**131**	0.35	1.43	1.53	1.77	1.45	3.37	3.60	**3.63**
WRU	70	75	72	60	63	75	75	**74**	0.35	1.43	1.63	1.77	1.45	2.11	2.30	**2.54**

The wrist flexion/extension joint rotates about an axis perpendicular to the forearm pronation/supination axis. The capstan for the wrist flexion/extension joint lies below the arm of the user so the wrist can be lowered onto the joint without having to navigate around any hardware components. The parameter corresponding to the forearm length can be adjusted and locked to ensure that the joint axis is aligned with the anatomical wrist flexion/extension axis of the user.

In order to maximize the range of motion of the wrist flexion/extension joint, the wrist radial/ulnar deviation joint rotates along a set of three pulleys about which the driving cable is wrapped. The cable is wrapped around the motor shaft similar to the other DoFs, but is wrapped around three pulleys before the capstan to increase the output transmission ratio. This allows for a capstan with a smaller arc angle that does not interfere with the rotation of the wrist flexion/extension joint, while still maintaining high torque output.

In addition to the parameter to adjust forearm length, MOE features two additional adjustable parameters to aid in alignment between the robot and the user. The handle for MOE is attached to the wrist radial/ulnar deviation joint with a linear rail and bearing. When MOE is in use, this passive DoF allows for small amounts of motion to account for any discrepancies between anatomical and robotic joint axes in the wrist. The last adjustable parameter is located above the elbow flexion/extension joint that allows for changes in shoulder abduction/adduction angle. Prior to use, this DoF can be set in 15° increments, and locked with a set of two screws. These screws are solely for locking purposes, as a large bolt attaches MOE to the cart and bears the weight of MOE.

B. Electronics Design

The electronics subsystem consists of a power supply, a printed circuit board (PCB) with mounted motor controllers and connecting interfaces, and an enclosure that houses these components while providing the user with access to its components and safety features.

A four-layer board is used for the PCB, with the two internal layers for grounding and two external layers for routing. The high power motor signals are routed on the top, while the rest of the signals are routed on the bottom. The PCB contains four isolated areas with separate common grounds: motor control, encoder pass through, digital signals, and analog signals. All grounding layers are tied together via connections to a thick metal plate to which the PCB mounts. These design choices reduce the cross talk between signals within the PCB and noise radiated from the PCB.

The layout of the PCB is shown in Fig. 3. The power supply output is fed and filtered through region 4 of the PCB, which is used to supply power to the motor drivers as seen in region 5. The motor controllers receive input from the data acquisition device

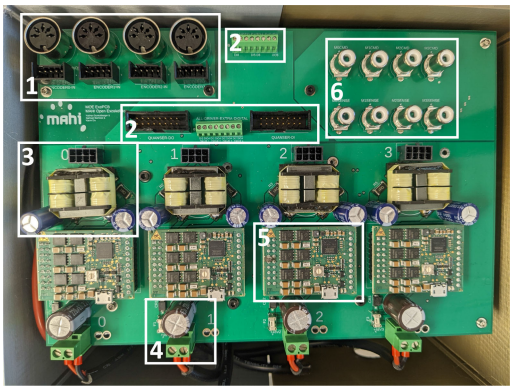

Fig. 3. MOE PCB, with functional regions labeled. 1) Encoder pass through. 2) Digital inputs and outputs to the chosen DAQ used for signal processing. 3) Signal conditioning circuit between the motor controller and motor output. 4) Electronic power stage. 5) ESCON motor controller. 6) Analog inputs and outputs.

(DAQ) in region 2 and run their output current through a signal conditioning circuit in region 3, where it is fed to the motors. Region 6 is used for analog inputs and outputs, which include current command and current sensing. Region 1 is designed as a simple encoder pass through. This layout allows all DAQ connections to run directly to the PCB rather than the robot, allowing the user to choose a new DAQ without having to change any wiring to the robot.

The motor controllers used in the PCB are ESCON Module 50/5 Servo Controllers (Maxon Motor), which are rated for 15 A max and 5-A continuous current at a nominal voltage of 50 Vdc. The controller operates using pulsewidth modulation (PWM) output at a rate of 53.6 kHz. To prevent the high-frequency noise typically present in PWM signals from being radiated by the motor cables, cable shielding, and a passive noise filter are employed. The cable shielding is routed to the motor controller ground plane to prevent ground plane cross talk with other signals. The noise filter chosen is a two-pole *LC* low-pass circuit where the inductive element is a common-/differential-mode choke that couples both motor outputs. The passbands for the common mode and differential mode were chosen to be 1 and 5 kHz, respectively, to remove the common-mode noise and the high-frequency differential-mode noise while leaving the low-frequency differential-mode signal alone.

III. Dynamic Characterization and Simulation

Experiments were performed to determine the inertia, viscous damping, kinetic friction, and static friction of each of the joints of MOE to support the development of a dynamic model, and to allow for comparison with prior upper limb robot designs. Closed-loop bandwidth was also determined. With the results from these experiments, dynamic equations were developed in MATLAB and implemented in C++, and a graphical user interface (GUI) was created to provide a visual representation of robot motion and behavior. This environment allows for expedited development of advanced control algorithms by allowing parameters to be adjusted and refined before implementing on hardware.

A. System Characterization

In each test, the joint of interest was aligned such that the joint axis was parallel to the direction of gravity. Methods were employed as reported in prior related work for similar robot designs [21]–[23]. Results of these characterization experiments are presented in Fig. 4 and Table II.

For an underdamped system, the step response can be used to obtain the dynamic properties via the logarithmic decrement method. We extracted the inertia, viscous damping, and kinetic friction components from the peaks and valleys of the step response [24]. Each joint was commanded to a position step of ten degrees centered about an angle chosen in the center of the workspace. A proportional controller was used and assumed to be the only source of stiffness in the system. No derivative control was used to ensure that there was no software damping present in the system. For each trial, three step responses were recorded and the results over each trial were averaged.

Static friction was measured as a function of joint position. For each joint, the commanded position consisted of no motion for 2 s followed by a ramp to five degrees over 2 s. A soft proportional-derivative (PD) controller was used so that the precise torque that initiated movement could be identified. During the ramp input, the joint velocity was measured and the commanded torque corresponding to the time step immediately prior to nonzero velocity was defined as the static friction torque at this position.

Since position-based control is a common control strategy for robotic exoskeletons, closed-loop position bandwidth was measured as an additional performance metric for MOE. A chirp input was used as the commanded position with a magnitude of ten degrees for each joint. The input provided a frequency sweep and the corresponding output magnitude was used to determine the bandwidth. The controller used for this was a critically damped PD controller.

B. Dynamic Simulation

Dynamic equations were generated using the iterative Newton–Euler method after determining the appropriate Denavit-Hartenberg (DH) parameters for MOE using the method presented by Craig [25], and are given in the following equation:

$$\tau = M(\theta)\ddot{\theta} + V(\theta, \dot{\theta}) + G(\theta). \quad (1)$$

The relevant distance parameters and mass properties needed for the dynamics were calculated using SOLIDWORKS for parts that were machined, and recorded from datasheets for parts that were not machined. In addition to the inertias from the machined parts, the reflected rotor inertias were added. The dynamic equations were generated symbolically using MATLAB, and then, converted to C++ so that they could be implemented for real-time simulation. The generated dynamic simulation solves the dynamic equations at an average rate of < 40 μs on a Windows 10 system with a 3.4-GHz, 6th generation, Intel i7 processor. This lends well for the model to be used in high-speed model-based control that requires dynamic equation evaluations. Because the dynamic equations are generated systematically

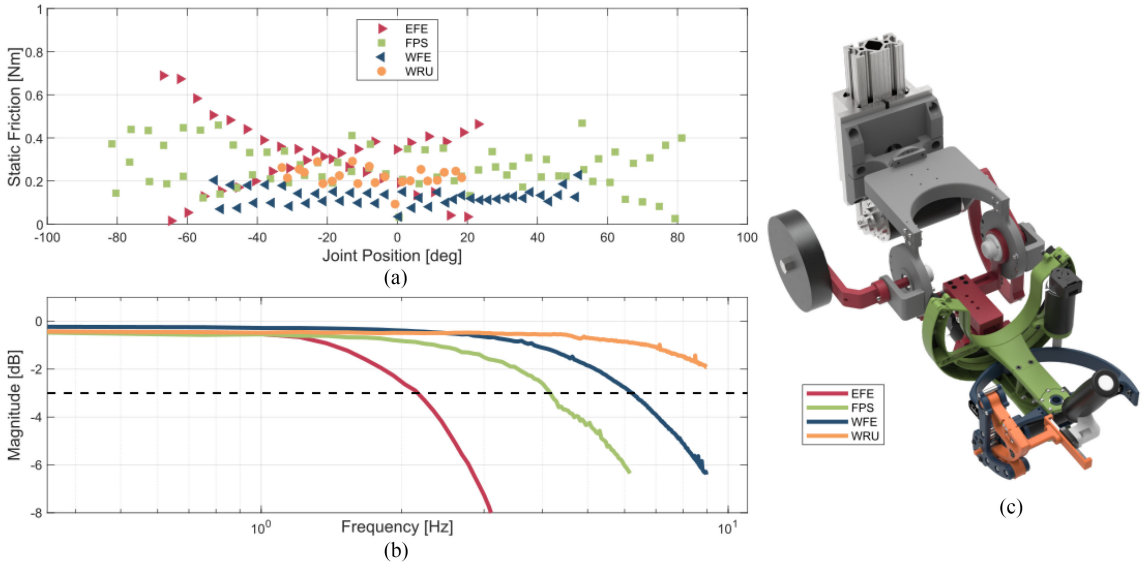

Fig. 4. (a) Static friction values across the workspace for all four joints of MOE. (b) Frequency response plots (3-dB attenuation cutoff shown). (c) MOE, with each joint colored according to the plot legend.

TABLE II
DEVICE CHARACTERISTICS OF MOE COMPARED WITH THE MAHI EXO-II [21] AND THE OPENWRIST [23]

	Inertia $[\mathrm{kg}\cdot\mathrm{m}^2]$			**Viscous coeff.** $[\frac{\mathrm{Nm}\cdot\mathrm{s}}{\mathrm{rad}}]$			**Kinetic friction** [Nm]			**Static friction** [Nm]			**Bandwidth** [Hz]		
Joint	ME-II	OW	MOE	ME-II	OW	MOE	ME-II	OW	MOE	ME-II	OW	MOE	ME-II	OW	MOE
EFE	0.2713	-	0.2061	0.1215	-	0.0393	-	-	0.1838	0.949	-	0.307	2.8	-	2.14
FPS	0.139	0.0305	0.0271	0.0167	0.0252	0.0691	-	0.1891	0.1572	0.139	0.2250	0.263	4.2	4.6	4.13
WFE	0.002	0.0119	0.0118	0.0283	0.0019	0.0068	-	0.0541	0.0996	0.109	0.0720	0.127	13.3	7.0	6.23
RU	0.0033	0.0038	0.0048	0.0225	0.0029	0.0025	-	0.1339	0.1685	0.112	0.1180	0.222	10.6	9.8	-

with values that only require the denavit-hartenberg (DH) table, mass properties, and part datasheets, this process can be easily adjusted for any robot design for which these details are known.

In the C++ simulation, the derivatives of position and velocity were calculated at each time step, then stepped forward in time using trapezoidal integration. This was implemented to run at a 1-kHz simulation rate, and an API was added that allowed a separate program to start, stop, or reset the simulation, as well as read the position and velocity, or set torque values. The simulation was implemented using the Eigen library [26] in C++, and a dynamic linking library was created to load them into the Unity3D game engine.

To aid in the understanding of the dynamic simulation, a visualization was created using the Unity3D game engine. The physical models were exported from SOLIDWORKS and imported into Unity3D so that they could be manipulated based on the simulation results. The extra functionality of stopping, starting, and resetting the simulation was made accessible in Unity so that the end user could set the simulation to the desired state.

Additionally, because all of the adjustable components on the robot are only able to be changed to discrete values, this finite set is available to select in the simulation, allowing users to accurately adjust the dynamics with simple sliders. This means that a user can estimate the required torques and resulting movements of the robot for any implementation intended for the real robot, regardless of desired configuration.

All that is needed to run the simulation is a Windows computer and the executable for the simulation, along with some configuration files. This extends the reach of MOE, making it possible for anyone to safely test a control implementation for MOE without needing the real robot. One example of this is shown in Fig. 2, where the gains for impedance control were tuned in the simulation before moving to the physical robot, ensuring that the robot would safely operate when used.

IV. SOFTWARE IMPLEMENTATION

An important part of MOE ecosystem is the open source software (github.com/mahilab/MOE) that enables ease of use at multiple interface levels. The software works with the physical robot and the simulation in an identical manner, without any change to source code, allowing for easy prototyping of control methods without requiring access to the physical robot. In a significant change from previous robots developed by our group, the software is now provided in C++ instead of MATLAB/Simulink. The interface enables the user to write code at an application level or a low-level control interface.

The choice of C++ as the development language means that the software is broadly available given access to a C++ compiler.

This also enables access to a wide variety of heavily tooled ecosystems built for use with C++, including several interfaces mentioned in Sections V-B and III-B. The predecessors to this robot [21], [22] were developed using MATLAB's Simulink environment, which limited the use to only those that have access to MATLAB's proprietary software, and made it difficult to use with some third party software.

This software framework has a built-in interface to both the physical robot, as well as the simulation of MOE. Both low-level and high-level interfaces are implemented so the user can be sure that they know what to expect when moving to the real robot, whether they are implementing a full experimental protocol, or a specific piece of a novel control algorithm. With this feature, software to test a control algorithm or experimental setup can be written once, and can be used in the exact same way for the simulation and the actual robot. The user just needs to have the simulation running and to implement a flag at runtime, and the simulation will be properly interfaced. When doing this, the simulation, shown in Fig. 2, immediately starts interacting with the executable as if it were the physical system.

The C++ library is built to enable access to low-level signals critical to the operation of the robot. The software interface allows read access of the position, velocity, and sensed torque output for each joint. If there is no hardware velocity estimation available, software velocity estimation is implemented using joint position data. Write access is provided for enabling each joint, and for writing torque outputs to each of the independent joints. With this low level control, it is possible to create many kinds of specialized control schemes.

If the user of the software is more interested in using the robot at an application level to provide general movements and control, the user also has access to higher level interfaces. Pretuned PD controllers are available for general use that will provide smooth and safe impedance control for use in rehabilitation or assistive applications. There are also features that allow the robot to be backdriven so that it can be used as a passive data-collection device.

While the robot has many redundant safeties built into each of its subsystems, the software also provides an additional level of safety. Position limits, velocity limits, and torque limits are available on each of the joints, as well as a DAQ watchdog check with the currently implemented DAQ. These redundant safeties are ideal for use in human–robot interaction cases where safety is critical.

V. Control Implementation

Impedance control is particularly relevant for rehabilitation applications, due to its ability to allow cooperative interaction between a user and the robot. MPC is also of interest, since model-based control algorithms are valuable when combining robots with other forms of actuation, such as functional electrical stimulation that can be modeled. We implemented both impedance control and MPC on MOE, in simulation and on the physical robot, to demonstrate how our simulation pipeline can be used to develop control algorithms. While only these two specific controllers are demonstrated in this article, the groundwork laid out here provides a significant set of resources for extending the abilities of MOE to use more advanced control algorithms.

A. Impedance Control

Impedance control was implemented with independent PD controllers on each joint. To tune the PD controllers, the simulation environment shown in Fig. 2 was used to find initial gains, then to refine the gains by following various trajectories. Using these constant gains, the resulting position and torque values for MOE following a set of test sinusoidal waves are shown in Fig. 5(a) and (b) respectively. The joint positions are overlapping, showing the similar output behavior; however, the joint torques do vary, especially for the elbow joint when it is flexed. These results demonstrate how a researcher can effectively tune the robot in the simulation and achieve similar output results when porting the control algorithms to the physical robot.

B. Model Predictive Control (MPC)

The state variables for MOE were defined as all position and velocity variables. The formulation of the MPC problem was defined with the following inputs, x, and outputs, u:

$$x = [q_1, q_2, q_3, q_4, \dot{q}_1, \dot{q}_2, \dot{q}_3, \dot{q}_4]^T \tag{2}$$

$$u = [\tau_1, \tau_2, \tau_3, \tau_4]^T \tag{3}$$

where q_1–q_4 represent the rotation of each DoF, in the order of the serial connection of the robot, $\dot{q}_1$–$\dot{q}_4$ represent the rotational velocities of these degrees of freedom, and τ_1–τ_4 represent the input torques of these DoFs.

The cost function that represents the optimal control problem is defined as

$$J_k = \sum_{i=1}^{N-1} (y_{k+i} - r_{k+i})^T Q (y_{k+i} - r_{k+i}) + \Delta u_{k+i}^T R \Delta u_{k+i} \tag{4}$$

where k represents the current time step, N represents the prediction horizon, r_{k+i} is the reference input at time step $k+i$, y_{k+i} is the predicted robot state at time step $k+i$, and Δu_{k+i} is the change in input between time $k+i$ and time $k+i-1$. $Q \in \mathbb{R}^{8\times 8}$ and $R \in \mathbb{R}^{4\times 4}$ are positive definite and diagonal weighting matrices that remain constant. The predicted robot state at each of the time step, y_{k+i}, was calculated by linearizing the equations of motion about the current state, and numerically integrating them using the forward Euler method. The resulting outputs from one step of an optimization $u_k, u_{k+1}, \ldots, u_{k+N}$ were used as control inputs to the robot until another optimization step finished.

A multiple shooting optimal control problem was created from the dynamics and cost function using the C++ optimal control framework, CasADi [27]. With the optimal control problem formulated, a file was generated that was able to be compiled and run at high speeds. This enabled us to use the CasADi framework along with the IPOPT nonlinear solver [28] to solve this optimal control problem in real time.

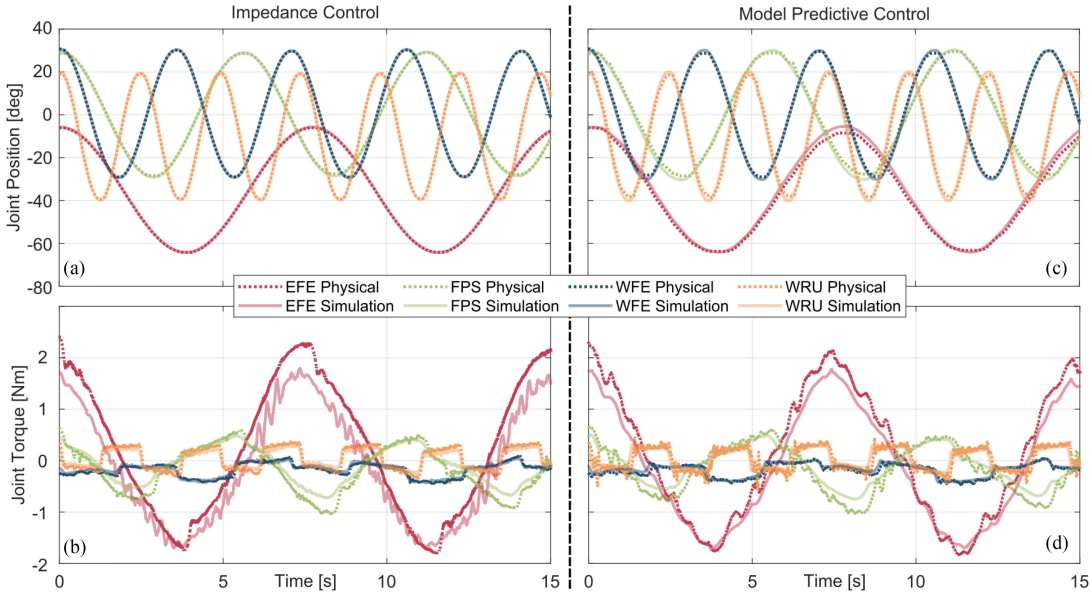

Fig. 5. Position and torque results for multi-DoF impedance control and MPC following a trajectory of mixed-frequency sinusoids. (a) and (b) Joint positions and corresponding joint torques resulting from impedance control. (c) and (d) Joint positions and corresponding joint torques resulting from MPC.

For this implementation, the same trajectory was used as for the impedance control implementation, although the algorithm is abstracted to be used with any desired trajectory. The gains for MPC were tuned in simulation, and then, the same gains were used to run the physical robot. The output trajectories and resultant torques are shown in Fig. 5(c) and (d), respectively, for the simulation and the physical robot. The results show that the joint positions match reasonably well, while the torques deviate some at the minimum and maximum positions.

VI. Discussion

Rehabilitation robots have shown promise as a tool to provide repetition-based upper limb therapy following a stroke or spinal cord injury, yet clinical studies using rehabilitation robots have not produced clinical outcomes that are significantly greater than those realized with standard of care interventions [5]. Researchers are now looking toward advanced rehabilitation intervention methods that rely on the integration of sensors and electrodes to enable neuromodulation in conjunction with robotic interventions [12]–[14]. Other advanced approaches require precise dynamic models of the robot to facilitate accurate estimates of both robot and human contributions to movements [10]. In this article, we present the kinematic design and dynamic modeling of MOE, a four DoF upper limb exoskeleton robot for rehabilitation of the upper limb following neurological injury. To facilitate the development of advanced control algorithms to use with MOE, a simulation of the robot was developed to provide a platform to prototype and refine novel implementations, enabled by the serial kinematic design. Explicit dynamic equations were generated that were solved quickly in real time, allowing for integration into model-based computational algorithms. This simulation helped our team to quickly develop gains for impedance control, and weighting parameters for MPC, which could be immediately run on the physical robot for further fine tuning. The results showed similar position accuracy between the simulation and the physical robot when the algorithms were developed on the simulation, then transferred to the robot. This method has already proven to be valuable to develop control strategies, and allows for further development into new control areas.

Toward the goal of enabling MOE to collaborate with other novel technologies, several modifications were made to previous robot designs. The range of motion was increased relative to the MAHI Exo-II and OpenWrist [21], [23], allowing the robot to be more relevant for assistance or rehabilitation applications simulating ADLs. The design was also changed to facilitate easy donning and doffing, making MOE well suited to be paired with other forms of physiological sensors and actuators, such as electromyography (EMG) [29] or surface functional electrical stimulation (FES) [11].

Impedance control and MPC were implemented and tested, and each showcases a different use case for the robot. Impedance control acts independently on each joint, and is simple to use, allowing it to be easily implemented in many different scenarios. However, in more complex control scenarios between several actuation devices, intelligent sharing cannot be achieved due to the simplicity. On the other hand, MPC uses knowledge of the entire dynamic system, and can integrate dynamic models from other novel technologies to realize a truly coordinated control scheme. However, it is difficult to create accurate cooperative dynamic models, and takes significant computational power to implement.

For both control strategies, the simulation proved effective at prototyping control gains and enabling similar position accuracy between the sim and the physical robot; however, the torque varied between the simulation and physical robot to achieve the same position. Several factors could explain the discrepancies

between the model and the physical system, such as unmodeled dynamics attributed to electrical cables or simplification of complex components (e.g., motors and encoders are currently modeled with uniform density).

If the user needs to understand the torque that will be required very precisely, future work could improve the dynamic characterization of the robot to make it more closely match the simulation. While friction and inertia were characterized for each individual joint, the mass properties of each physical component were estimated using SOLIDWORKS based on built-in material properties. To fine tune these values, components could be physically measured to achieve a better agreement between the simulation and physical robot.

Because the dynamic equations of the robot were explicitly developed, other model-based robot features could also be easily implemented. Gravity compensation could be implemented by providing a portion of the $G(\theta)$ vector, as shown in (1), as a feed-forward input to the robot. Likewise, friction compensation could be implemented utilizing the characterized friction parameters from Section III-A. Both of these features would be useful in assistive and rehabilitation applications, making assistance more consistent throughout the workspace, and in passive, data-collection modes, where it can allow the user to more easily backdrive the robot.

VII. Conclusion

MOE is a 4-DoF upper limb robot that was developed to address specific areas of emphasis for future robotic rehabilitation paradigms. The serial kinematic design simplifies the computational burden for model-based control design. In parallel with the improved hardware design, we developed a dynamic simulation and software interface environment to facilitate the development of model-based control algorithms. An added feature of the new robot design is ease of integration with hardware components like sensors and electrodes necessary for combined robotic and neuromodulation interventions.

VII. Acknowledgment

The authors would like to thank E. Pezent, A. Lettenberger, and E. Mahan for the contribution to the development of MOE.

References

[1] B. Ovbiagele *et al.*, "Forecasting the future of stroke in the United States: A policy statement from the American Heart Association and American Stroke Association," *Stroke*, vol. 44, no. 8, pp. 2361–2375, Aug. 2013.

[2] National Spinal Cord Injury Statistical Center, "Spinal cord injury facts and figures at a glance," National Spinal Cord Injury Statistical Center, Birmingham, AL, USA, 2019.

[3] A. A. Blank, J. A. French, A. U. Pehlivan, and M. K. O'Malley, "Current trends in robot-assisted upper-limb stroke rehabilitation: Promoting patient engagement in therapy," *Curr. Phys. Med. Rehabil. Rep.*, vol. 2, no. 3, pp. 184–195, 2014.

[4] M. Mekki, A. D. Delgado, A. Fry, D. Putrino, and V. Huang, "Robotic rehabilitation and spinal cord injury: A narrative review," *Neurotherapeutics*, vol. 15, no. 3, pp. 604–617, 2018.

[5] A. C. Lo *et al.*, "Robot-assisted therapy for long-term upper-limb impairment after stroke," *New England J. Med.*, vol. 362, no. 19, pp. 1772–1783, 2010.

[6] H. Singh *et al.*, "Robot-assisted upper extremity rehabilitation for cervical spinal cord injuries: A systematic scoping review," *Disabil. Rehabil., Assist. Technol.*, vol. 13, no. 7, pp. 704–715, 2018.

[7] A. H. Lequerica and K. Kortte, "Therapeutic engagement: A proposed model of engagement in medical rehabilitation," *Amer. J. Phys. Med. Rehabil.*, vol. 89, no. 5, pp. 415–422, May 2010.

[8] O. Mubin, F. Alnajjar, N. Jishtu, B. Alsinglawi, and A. Al Mahmud, "Exoskeletons with virtual reality, augmented reality, and gamification for stroke patients' rehabilitation: Systematic review," *J. Med. Internet Res. Rehabil. Assist. Technol.*, vol. 6, no. 2, Sep. 2019, Art. no. e12010.

[9] A. U. Pehlivan, D. P. Losey, and M. K. O'Malley, "Minimal assist-as-needed controller for upper limb robotic rehabilitation," *IEEE Trans. Robot.*, vol. 32, no. 1, pp. 113–124, Feb. 2015.

[10] J. Berning, G. E. Francisco, S.-H. Chang, B. J. Fregly, and M. K. O'Malley, "Myoelectric control and neuromusculoskeletal modeling: Complementary technologies for rehabilitation robotics," *Curr. Opin. Biomed. Eng.*, vol. 19, 2021, Art. no. 100313.

[11] N. Dunkelberger, E. M. Schearer, and M. K. O'Malley, "A review of methods for achieving upper limb movement following spinal cord injury through hybrid muscle stimulation and robotic assistance," *Exp. Neurol.*, vol. 328, 2020, Art. no. 113274.

[12] F. Resquín *et al.*, "Hybrid robotic systems for upper limb rehabilitation after stroke: A review," *Med. Eng. Phys.*, vol. 38, no. 11, pp. 1279–1288, 2016.

[13] D. Simonetti *et al.*, "Literature review on the effects of tDCS coupled with robotic therapy in post stroke upper limb rehabilitation," *Front. Hum. Neurosci.*, vol. 11, 2017, Art. no. 268.

[14] P. E. Dupont *et al.*, "A decade retrospective of medical robotics research from2010 to 2020," *Sci. Robot.*, vol. 6, no. 60, 2021, Art. no. eabi8017.

[15] A. Nasr, B. Laschowski, and J. McPhee, *Myoelectric Control of Robotic Leg Prostheses and Exoskeletons: A Rev.*, (International Design Engineering Technical Conferences and Computers and Information in Engineering Conference), Virtual, vol. 8A:, Aug. 2021.

[16] K. Takeda, G. Tanino, and H. Miyasaka, "Review of devices used in neuromuscular electrical stimulation for stroke rehabilitation," *Med. Devices*, vol. 10, pp. 207–213, Aug. 2017.

[17] A. Venkatakrishnan, G. E. Francisco, and J. L. Contreras-Vidal, "Applications of brain–machine interface systems in stroke recovery and rehabilitation," *Curr. Phys. Med. Rehabil. Rep.*, vol. 2, no. 2, pp. 93–105, Jun. 2014.

[18] H. I. Krebs *et al.*, "Robot-aided neurorehabilitation: A robot for wrist rehabilitation," *IEEE Trans. Neural Syst. Rehab. Eng.*, vol. 15, no. 3, pp. 327–335, Sep. 2007.

[19] L. Cappello *et al.*, "Evaluation of wrist joint proprioception by means of a robotic device," in *Proc. 11th Int. Conf. Ubiquitous Robots Ambient Intell.*, Nov. 2014, pp. 531–534.

[20] J. A. Martinez *et al.*, "Design of wrist gimbal: A forearm and wrist exoskeleton for stroke rehabilitation," in *Proc. IEEE Int. Conf. Rehab. Robot.*, Jun. 2013, pp. 1–6.

[21] J. A. French *et al.*, "System characterization of MAHI Exo-II: A robotic exoskeleton for upper extremity rehabiliation," in *Proc. Amer. Soc. Mech. Engineers Dyn. Syst. Controls Conf.*, 2014, pp. 1–5.

[22] A. U. Pehlivan *et al.*, "Design and validation of the RiceWrist-S exoskeleton for robotic rehabilitation after incomplete spinal cord injury," *Robotica*, vol. 32, no. 8, pp. 1415–1431, Dec. 2014.

[23] E. Pezent, C. G. Rose, A. D. Deshpande, and M. K. O'Malley, "Design and characterization of the openwrist: A robotic wrist exoskeleton for coordinated hand-wrist rehabilitation," in *Proc. Int. Conf. Rehabil. Robot.*, 2017, pp. 720–725.

[24] J. W. Liang and B. F. Feeny, "Identifying coulomb and viscous friction from free-vibration decrements," *Nonlinear Dyn.*, vol. 16, no. 4, pp. 337–347, 1998.

[25] J. J. Craig, *Introduction to Robotics: Mechanics and Control*, 3rd ed. New York City, New York: Pearson Education, 2009.

[26] G. Guennebaud *et al.*, "Eigen v3," 2010. [Online]. Available: http://eigen.tuxfamily.org

[27] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi—A software framework for nonlinear optimization and optimal control," *Math. Program. Comput.*, vol. 11, no. 1, pp. 1–36, 2019.

[28] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Math. Program.*, vol. 106, no. 1, pp. 25–57, Mar. 2006.

[29] C. G. McDonald, J. L. Sullivan, T. A. Dennis, and M. K. O'Malley, "A myoelectric control interface for upper-limb robotic rehabilitation following spinal cord injury," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 28, no. 4, pp. 978–987, Apr. 2020.