

Deep Learning Enables Exoboot Control to Augment Variable-Speed Walking

Max K. Shepherd¹, *Member, IEEE*, Dean D. Molinaro², *Student Member, IEEE*, Gregory S. Sawicki², *Member, IEEE*, and Aaron J. Young², *Member, IEEE*

Abstract—Ankle exoskeletons have the potential to improve mobility, but common controllers are often inflexible to variations in tasks, such as changes in walking speed. To enable effective variable-speed exoboot control, we developed and validated a two-headed convolutional neural network trained to (1) classify stance versus swing and (2) predict the phase during stance, which was then mapped to a desired exoboot torque. This Machine Learning Estimator (MLE) was trained from nine participants walking at three speeds and four exoboot assistance levels. A Time-Based Estimator (TBE) that predicted gait phase from the two previous stride durations was used to apply realistic torques during MLE training and served as a within-participant control condition. The MLE was validated online with three new participants walking at a range of speeds and torques, both interpolating within and extrapolating outside the training set. Online validation accuracy (RMSE) across tested speeds and torque levels was 3.9%. On a simple walking task in which treadmill speed was varied sinusoidally between 1.1 and 1.6 m/s with a 30 s period, the three participants exhibited a mean 5.2% decrease in metabolic expenditure with the MLE compared to no-exo (boots only), but exhibited a 5.4% increase when walking with the TBE. The MLE more accurately predicted heel strike and toe off events (heel strike Mean Absolute Error: 9.6 ms; toe off MAE: 13.2 ms) than the TBE (heel strike MAE: 19.1 ms; toe off MAE: 34 ms). These positive results validated the potential of using a deep learning model for gait state estimation to effectively control an ankle exoskeleton across variable walking speeds.

Index Terms—Exoskeletons, Machine Learning

I. INTRODUCTION

Powered ankle exoskeletons have successfully augmented human energetics during walking [1]–[3]. Recent studies have shown that these energetic benefits are highly sensitive to assistance timing, favoring assistance strategies that provide positive net joint work during stance (i.e., commanding plantarflexion torque during late stance) [4]–[6]. Additionally, failure to promptly disengage plantarflexion assistance after toe off (i.e., as the leg enters swing) can lead to increased antagonistic muscle activation and/or tripping. These considerations can be accounted for via accurate stance phase

estimation. Stance phase can be defined using a continuous gait variable that monotonically increases from 0% at heel strike to 100% at toe off, and may be a function of time or a kinematic phase variable [7]. Using a real-time estimate of stance phase, ankle exoskeleton controllers can then compute desired plantarflexion assistance using a predefined phase-based torque trajectory.

Most simply, stance phase can be estimated in a feed-forward manner using a Time-Based Estimator (TBE), which stores the timing of heel strike and toe off events from previous strides in memory. Heel strike and toe off events are measured from onboard contact sensors, such as force sensitive resistors (FSRs) [2], [8], [9], ground reaction forces [2], or IMU-based heuristics [1], [10]. Using the data from previous strides, the TBE computes an expected stance duration, often by implementing a moving average filter over the previous measured stance durations. Finally, stance phase is computed by dividing the time since last heel strike by the expected stance duration. Using a TBE, previous researchers have reduced the metabolic cost of walking with ankle exoskeletons [1], [2], [10]; however, these studies have been limited to steady-state ambulation (e.g., constant speed treadmill walking), since the TBE assumes steady, periodic ambulation.

Alternatively, machine learning has become a popular approach for estimating gait variables used in exoskeleton controllers [11]–[15]. We previously investigated the efficacy of various types of deep neural networks for estimating gait phase using sensors onboard a robotic hip exoskeleton [11], [12]. We found that using a Convolutional Neural Network (CNN) significantly outperformed a TBE during overground ambulation. Specifically, the CNN-based gait phase estimator was able to instantaneously adapt gait phase estimates with natural stride-to-stride changes, while the expected stride duration of the TBE lagged, due to its reliance on previous stride data. Though the principles of gait phase estimation promise to be applicable for ankle exoskeletons, it remains unclear if a machine learning stance phase estimator would have similar benefits. Distal sensors may contain more

Manuscript received: 9, Sept, 2021; Revise Dec 10, 2021; Accepted Jan 16, 2022. This paper was recommended for publication by Editor Jee-Hwan Ryu upon evaluation of the Associate Editor and Reviewers' comments. Research supported in part by the National Science Foundation Graduate Research Fellowship under Grant No. DGE-1650044 and the NSF NRI Award No. 1830215.

¹M. K. Shepherd is with the College of Engineering and Bouvé College of Health Sciences, Northeastern University, Boston, MA, USA. He was

previously with the George W. Woodruff School of Mechanical Engineering, Georgia Institute of Technology, Atlanta, GA, USA, and (e-mail: m.shepherd@northeastern.edu).

²D. D. Molinaro, G. S. Sawicki and A. J. Young are with the George W. Woodruff School of Mechanical Engineering, Georgia Institute of Technology, Atlanta, GA, USA and the Institute for Robotics and Intelligent Machines (IRIM), Georgia Institute of Technology, Atlanta, GA, USA.

Digital Object Identifier (DOI): see top of this page.

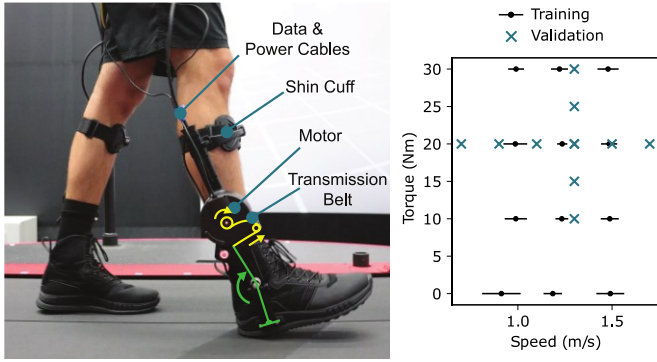


Fig. 1. (Left) Picture of the Dephy Exoboos. (Right) The speeds and torques collected as training data from nine participants (black error bars), and the torque and speed levels tested during validation (blue x's).

information about foot contact than proximal sensors, which would be advantageous in detecting stance/swing transitions. However, many ankle exoskeletons utilize unidirectional cable-based transmissions, which enable light and efficient torque amplification, but which necessitate mid-level state machines capable of managing slack during swing [2], [16]–[18]. Because of this, it is critical to accurately detect heel strike and toe off events in tandem with stance phase estimation to achieve effective ankle exoskeleton assistance.

In this paper, we propose and validate a user-independent Machine Learning Estimator (MLE) that simultaneously (1) detects stance/swing transitions (binary classification) and (2) estimates stance phase (regression) using a two-headed, deep convolutional neural network. We validated this system by quantifying stance phase and stance/swing transition accuracy while using the MLE to control the Dephy ExoBoots (Dephy Inc., Maynard, MA, USA), a commercially available bilateral ankle exoskeleton (Fig. 1). To test the real-world efficacy of our system, we evaluated its performance on three novel users during constant-speed walking conditions within and outside of the training set distribution and during variable-speed walking. Additionally, we hypothesized that the benefits of machine learning gait state estimation would lead to a reduction in metabolic cost during variable speed walking compared to using a TBE for exoskeleton control. Given the positive results of our proposed framework, our study solves the problem of stance phase estimation during real-world gait and demonstrates the benefits of this framework during transient ambulation. To reduce the barrier-to-entry for using our MLE on the commercially available Dephy Inc. exoboos, we have also released the trained parameters of our network, available here: <https://github.com/maxshep/exoboot-ml-gait-state-estimator>.

II. POWERED EXOBOOTS

A. Exoboot Hardware

The Dephy ExoBoots (Model 504; Firmware v7.1, Dephy Inc., Maynard, MA, USA) are powered boots, capable of applying ~ 30 Nm of peak plantarflexion torque through shin cuffs and carbon fiber keels embedded in the boots' midsoles (Fig. 1). A shank-mounted motor transmits torque to the ankle

through a nonlinear, unidirectional belt-driven transmission. Additionally, a waist-mounted pack houses a Raspberry Pi 4B microprocessor (Raspberry Pi Foundation, Cambridge, UK) used to run the main control loop, a 5 V battery that powers the microprocessor, and two 22.2 V lithium polymer batteries wired in series to power the exoboot actuators. The exoboos have built-in shank-mounted IMUs, and a 14-bit absolute encoder at the ankle joint. The left IMU was transformed to follow left-hand rule, and the absolute encoders were zeroed based on kinematic hard stops (i.e., were not recalibrated between participants). These adjustments mirrored sensor data across the sagittal plane, and left/right data was effectively indistinguishable. Each exoboot had mass 1.4 kg, and the waist-mounted pack had a total mass of 1.9 kg.

B. Exoboot Controller

The main control loop ran sequentially on the Raspberry Pi at 200 Hz using Python v3.7. To accommodate the change in ankle dynamics between stance and swing, the controller was implemented using a four-state finite-state machine. During leg swing (state 1), the exoboos quickly decoupled the user's ankle joint from the motor's reflected inertia by slacking (reeling-out) the belt. Slacking the belt prevented any detrimental resistance to swing-phase dorsiflexion, allowing for adequate ground clearance of the toes. During this state, the actuators were commanded using position control to minimize excessive slack in the system, which could lead to delayed assistance onset. At heel strike, the controller softly reeled-in the belt by commanding voltage to mitigate rapid torque onset to the user at heel strike (state 2). After reel-in, exoboot assistance torque was commanded to the actuators using open-loop torque control (state 3). The desired assistance torque was computed using a predefined piecewise cubic hermite interpolating polynomial, which was a function of the estimated stance phase. The nodes of the assistance polynomial were tuned based on those optimized by Zhang *et al.* [2]. The resulting assistance polynomial maintained a bias torque of 3 Nm after reel-in until 33% of stance phase, then increased assistance until a peak torque at 84% of stance.

C. Time-Based Estimation of Gait Phase (TBE)

To collect data for our CNN-based gait state estimator, we developed a baseline controller using a set of heuristics to estimate gait phase, heel strike, and toe off suitable for constant speed walking. Heel strike was detected as a peak in the onboard sagittal plane angular rate gyro, with a 50 ms added delay to better align with true heel strike; this delay was determined from high-speed video ($n=1$). A Time-Based Estimator (TBE) then used a moving average to track the two previous stride durations and project forward to the next stride. Notably, our TBE was more conservative (i.e., used fewer previous strides) than most other TBEs; we wanted our TBE to be highly adaptive to natural changes in walking speed during our over-ground trials. Toe off was hardcoded at 62% of the predicted stride duration, based also on high-speed video as well as pilot participant feedback ($n=1$), and stance phase was calculated from a linear interpolation in time between heel

strike and toe off. Importantly, this method is open-loop; the only external sensing is the heel strike detection via the gyroscope, and stance phase and toe off were based entirely off the previous two stride durations.

III. CNN-BASED GAIT STATE ESTIMATOR DEVELOPMENT

A. Training Data and Labeling

Nine participants (5 male, 4 female; height: 171 ± 10 cm; body mass: 74 ± 9 kg) participated in the training data collections. All participants in this study provided informed consent to participate, and this study was approved by the Georgia Institute of Technology Institutional Review Board. Training data trials consisted of a grid of walking speeds (0.9 m/s to 1.5 m/s) and exoboot torque levels (0, 10, 20, and 30 Nm) (Fig. 1), with torque applied based on the TBE. The purpose of applying torque during the training data collections was to create more realistic sensor input for the model to be trained on; subjects often change lower-limb kinematics when walking due to added assistance [19], and the interface displaces due to soft tissue compliance as well as flexing in the exoboos. Thus, it was important to capture these effects in the training data by mimicking the desired controller. Trials took place overground in an 81 m hallway, and an experimenter set the pace (after timed practice runs) while walking in front of the subject with a stopwatch and target trial completion times.

Force-Sensing Resistors (FSRs) (Tekscan, Boston, Mass, USA) were taped underneath the insoles, and approximately under the heel and the head of the first metatarsal joint. FSR data were debounced with a 40-sample median filter and visually inspected and manually corrected for missed strides. The stance/swing label was determined from the rising edge of the heel FSR and the falling edge of the toe FSR, and the stance phase label was calculated as a temporally-interpolated percentage between heel strike (0% stance) and toe off (100% stance).

B. Machine Learning Estimator of Gait State (MLE)

The purpose of the Machine Learning Estimator (MLE) was

to predict stance/swing (binary classification) and the percent stance (regression) from a sliding window of unilateral exoboot sensor data (Fig. 2). Based on our previous work, we implemented a deep convolutional neural network (CNN) [11]. Inputs to the network included 3-axis linear accelerations and 3-axis angular velocities from the built-in, shank-mounted IMU, ankle angle from an absolute magnetic encoder, and ankle velocity, which was calculated via first-order finite differencing and filtered with a causal 2nd order 10 Hz Butterworth filter. IMUs follow right-hand-rule on the right exoboot and are mirrored to follow left-hand-rule on the left exoboot.

The neural network was trained in Tensorflow v2.3.0. The 1D CNN portion of the network consisted of three convolutional layers (30 feature maps each, and kernel sizes of 20, 20, and 6), with ReLU activation functions. The input consisted of a window of the 44 most recent samples (~220 ms). Kernel sizes were chosen for the CNN to reduce the window to a single neuron depth (by the 30 filters in the final layer). The first and second convolutional layers were followed by batch normalization. The network then split into two fully-connected layers with 20 neurons each. One head predicted percent stance, with a MSE loss function that ignored swing phase. The other head detected stance/swing using a binary cross entropy loss function. Losses were combined with a weighted average (80% for the stance phase regression, 20% for the stance/swing classification).

Training data consisted of 128-sample sequences. The model was trained using data from both left and right exoboos simultaneously. The model was optimized with Adam, an adaptive learning rate optimization algorithm, and stopped after 12 epochs (determined from average performance on leave-one-subject-out validation).

C. Offline Validation

Prior to online testing, we assessed sensitivity of offline leave-one-participant-out validation results to various components of model architecture (e.g., CNN and DNN (Deep Neural Network) width and depth, kernel sizes, normalization

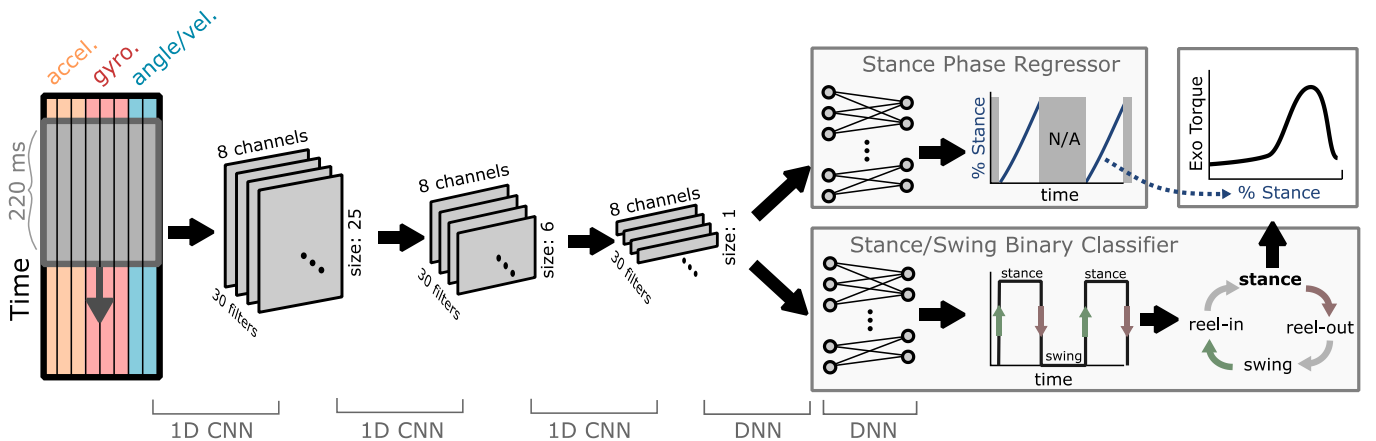


Fig. 2. Machine Learning Estimator (MLE) of gait state model architecture. A sliding 220 ms window of sensor data is input to the first convolutional layer (kernel size: 20). Intermediate convolutional layers further compress until the network is flattened and split into two heads; one (bottom inset) concludes in a binary classifier predicting stance/swing, and the other (top inset) estimates stance phase. The stance/swing classifier informs the four-state state machine, which produces plantarflexion torque during stance based on the stance phase regressor.

layer locations). Generally, increasing the number of CNN or DNN layers did not lead to improvements in performance, nor did low-pass filtering input data. Average offline Root Mean Square Error (RMSE) for stance phase across participants was 3.8%.

IV. ONLINE VALIDATION

A. Online MLE Implementation

The MLE was implemented on an NVIDIA Jetson Nano (NVIDIA, Santa Clara, CA, USA) for online inference using Python v3.6.9, which accounted for 0.4 kg of the waist-mounted system. Before running the network online, it was converted to a TensorRT runtime engine to maximize inference speed using TensorRT v7.1.3. The Jetson Nano (server) communicated with the Raspberry Pi (client) via an ethernet cable using TCP/IP. During operation, the Raspberry Pi streamed exoboot sensor data of size $\mathbb{R}^9$ to the Jetson Nano, containing IMU and encoder data of a single exoboot. Each packet also contained a flag indicating the corresponding exoboot (left or right). The Jetson Nano maintained a first-in-first-out buffer of exoboot sensor data with size $\mathbb{R}^{44 \times 8}$ for each exoboot, which was used as the input to the MLE model. The MLE was sampled with each incoming data packet from the Raspberry Pi. After inference, the Jetson Nano streamed the MLE estimates along with the side flag back to the Raspberry Pi with an approximate latency of 6 ms from the time the corresponding data packet was sent by the Raspberry Pi.

B. Online MLE Validation Protocol

Three participants (2 male, 1 female; height: 179 ± 10 cm; body mass: 66 ± 11 kg), who did not participate in the training data collection protocol, participated in each stage of the online MLE validation protocol: 1) MLE accuracy testing at multiple peak assistance torques; 2) MLE accuracy testing at multiple constant walking speeds; 3) MLE and TBE accuracy testing and human metabolic cost validation during variable-speed treadmill walking. During Stage 1 of the protocol, each participant walked for 1 min/trial on a Motek CAREN treadmill at 1.2 m/s while the MLE-informed exoboos assisted the user with each validation peak torque shown in Fig. 1 (10, 15, 20, 25, 30 Nm). During Stage 2 of the validation protocol, each participant walked for 1 min/trial on the treadmill at each validation speed shown in Fig. 1 (0.7, 0.9, 1.2, 1.5, 1.7 m/s) while the MLE-informed exoboos assisted the user with a peak torque of 20 Nm.

For Stage 3 of the validation protocol, each participant was outfitted with a COSMED K5 metabolic system (COSMED USA, Inc., Concord, CA, USA). A sinusoidally varying treadmill speed profile was then prescribed, ranging from 1.1 to 1.6 m/s with a 30 s period. Subjects walked twice each in three different conditions: 1) NO EXO (boots only with all exoboot hardware removed); 2) Time-Based Estimator (TBE), in which the exoboos provided assistance with a peak torque of 30 Nm using a two-stride moving average TBE as described in Section II.C; and 3) Machine Learning Estimator (MLE), in which the exoboos provided assistance with a peak torque of 30 Nm

using our MLE as described in Section III.B. Trials for the three conditions were repeated in a within-participant counterbalanced design (i.e., ABC-CBA). Participants were blinded to the condition, and the trial order was pseudo-randomized, with the NO EXO condition either in the A or C position to reduce exoboot don/doff time during the experiment. The participants walked at each condition for 6 minutes and metabolic cost was computed using a modified Brockway equation [2], [20], which was a function of the VO_2 and VCO_2 data. Six minutes of quiet standing followed the walking trials, to determine basal metabolic rate. The resulting steady-state metabolic cost for each condition was computed as the average metabolic cost computed over the last three minutes of each trial, minus the metabolic cost of quiet standing.

V. RESULTS

The three validation participants completed all trials. One participant completed the constant-speed trials and the varying-speed trials on separate days, and one of the participants had an FSR misplacement, and their left exoboot data were discarded.

Stance phase Root Mean Square Error (RMSE) across all tested constant speeds and torque levels was 3.9% for the MLE, which was close to the RMSE found in our offline leave-one-participant-out analysis (3.8%) (Fig. 3). There was a large participant-specific effect on stance phase RMSE, which may have been due to either participant-specific gait patterns or FSR placement.

During the variable-speed trials, commanded torque RMSE relative to the FSR-based retrospective ground truth was 2.6 ± 0.2 Nm and 3.3 ± 0.2 Nm for the MLE and TBE, respectively (Fig. 4a). Similarly, the MLE more accurately estimated peak assistance timing compared to the TBE, resulting in an average peak assistance timing Mean Absolute Error (MAE) of 19 ± 8 ms and 27 ± 2 ms, respectively (Fig. 4b). Additionally, there were large disparities in heel strike and toe off accuracy and consistency (Fig. 4d and 4e). On average, the TBE was early in detecting heel strike (14 ± 19 ms early) and late in detecting toe off (24 ± 33 ms late), with larger between-stride variability. The MLE, by contrast, was late but more consistent in detecting both heel strike (9.3 ± 7.1 ms), and toe off (7.7 ± 15 ms). There was also a one-cycle delay in communication between the Raspberry Pi and Jetson Nano, which would account for 5 ms of the MLE delay on average. The MAE for the MLE's heel strike and toe off detector were 9.6 ms and 13.2 ms, respectively, compared to the TBE's heel strike (MAE: 19.2 ms) and toe off detector (MAE: 34.0 ms).

Finally, all three participants reduced their metabolic cost while wearing the exoboos when using the MLE compared to NO EXO (mean: 5.2% decrease; [-7.2%, -3.3%, -5.1%]; Fig. 4c). The TBE that was used to collect realistic training data did not show an improvement in metabolic cost compared to the NO EXO condition (mean: 5.4% increase; [+8.7%, +6.7, -

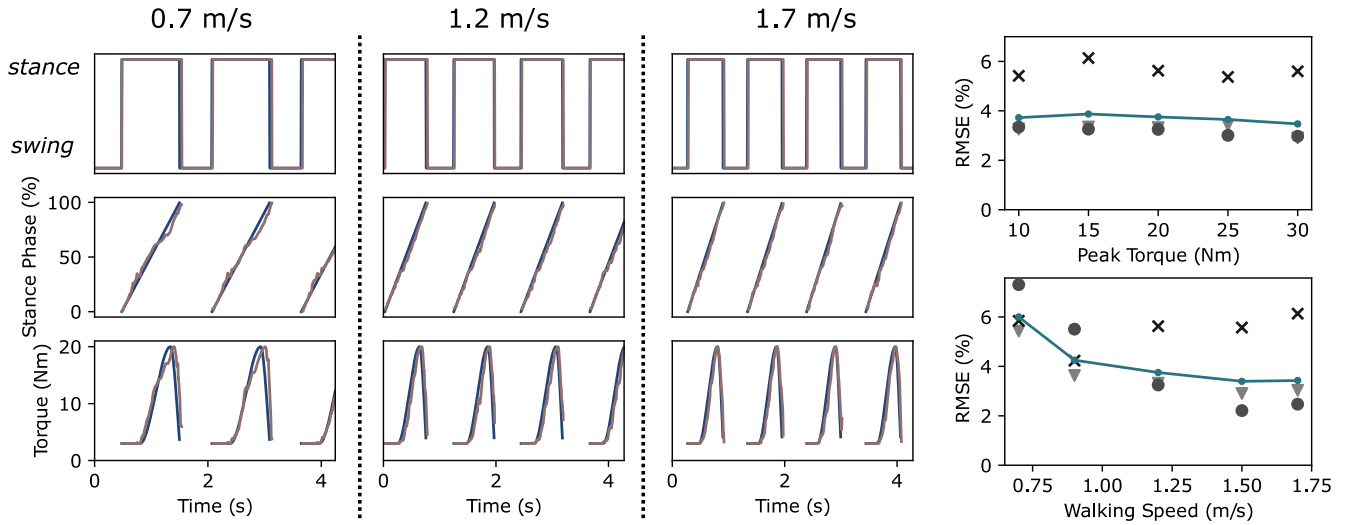


Fig 3. (Left) Data from the MLE stance/swing classifier (top), the stance phase estimator (middle), and the resulting commanded torque profile (bottom) are shown for a representative participant walking at multiple speeds. Ground truth profiles (blue) are derived from FSR labeling. (Right) RMSE vs. peak torque (top), and RMSE vs. walking speed (bottom) for the three validation participants. Markers denote individual participant means. Note: FSR data from the left exoboot of the participant denoted with 'x' was discarded due to sensor failure, so their data had a lesser effect on the mean.

2.9%]; Fig 4c). Average basal metabolic rate was 1.16 ± 0.21 W/kg.

VI. DISCUSSION

This study introduced and validated an end-to-end user-independent Machine Learning Estimator (MLE) that instantaneously detected stance/swing transitions and estimated stance phase during variable-speed walking for ankle exoboot control. Our approach used a two-headed convolutional neural network capable of estimating gait state using onboard exoboot sensors without the need for user-specific calibration. To validate our system, the MLE was evaluated online as novel users walked with the MLE-informed exoboot controller during both constant-speed and variable-speed walking. In general, the MLE performed well, leading to a mean 5.2% reduction in metabolic cost during variable-speed walking compared to the NO EXO condition ($n=3$: -7.2%, -3.3%, -5.1%, Fig. 4c). This result was accompanied with an average commanded torque RMSE of 2.6 Nm (peak assistance torque was 30 Nm) and an average peak assistance timing MAE of 19 ms. Thus, our proposed method generalized well to speeds and accelerations representative of natural gait [21].

Using the MLE to control the exoboos reduced the metabolic cost of walking by a total of 10.6% compared to using the TBE during the variable-speed trial ($n=3$: 15%, 9.7%, 2.3%; Fig. 4c). This was expected since the TBE relied on data from the previous two strides to estimate the expected duration of the current stride. As the walking speed changed, the expected stride duration of the TBE consistently lagged the correct value. Surprisingly, this discrepancy completely removed the energetic benefit of the exoboos, as the MLE condition reduced the metabolic cost of walking compared to not wearing the exoboos while the TBE condition increased it. Further, all three validation participants strongly preferred the MLE compared to the TBE during these tests, despite being blinded to the

controller conditions.

During the variable-speed trials, the TBE misidentified toe off timing more frequently, with toe off error distributions of 24 ± 33 ms compared to the MLE: 7.7 ± 15 ms (Fig. 4e). Similarly, the TBE had larger error in peak torque timing than the MLE (Fig. 4b). These results reinforce the findings from previous ankle exoskeleton studies [2], [8] that late stance phase estimation accuracy is critical for effective ankle exoskeleton control. Additionally, we found that the MLE error distribution about heel strike was smaller than that about toe off. It is likely that the impact of heel strike provided richer information in the exoskeleton sensor data compared to during toe off. To further prioritize accurate late stance phase estimation and toe off detection, it is possible to customize the loss function to weight the relative importance of estimation accuracy throughout stance and between heel strike and toe off detection.

During the online validation trials, the MLE generalized well to changes in walking speed and peak exoboot assistance. This validation included walking speeds both within and outside of the training set, as well as a wide range of peak assistance torques including the maximum torque capable of the exoboos (30 Nm), suggesting that the MLE generalized well to a variety of conditions experienced during level ground walking. Further, the stance phase RMSE and stance/swing error distributions of the MLE were similar when comparing the results of the constant speed validation trials (Fig. 3) to the variable-speed trials (Fig. 4), in which the participant was continuously accelerating or decelerating their walking speed. Of these conditions, the MLE performed worst during the slowest walking speed (0.7 m/s). This may be caused by increased stride-to-stride variability in walking at low speeds [22], or the decrease in signal to noise ratio of the kinematic exoboot sensors during this condition compared to the others. More generally, phase-based approaches will require a high-level controller to govern behavior during non-cyclic or quasi-cyclic tasks like shuffling, and may need to switch to other mid-level

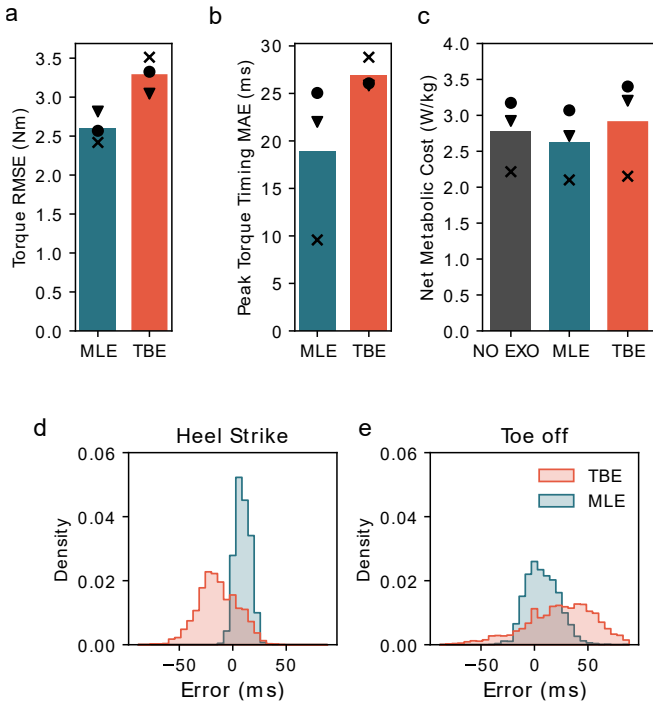


Fig 4. Performance comparison between the Time-Based Estimator (TBE) and Machine Learning Estimator (MLE) during a treadmill task with sinusoidally varying speed. (a) The MLE had lower torque RMSE than the TBE and (b) lower MAE of peak torque timing. Note: individual participant averages are denoted with unique markers. (c) The resulting metabolic cost while walking in the exoboots using the MLE and TBE during the variable-speed trial is shown, along with the resulting metabolic cost of walking without wearing the exoboots (NO EXO). (d) The TBE also had larger errors in heel strike estimation and (e) toe off estimation compared to the MLE.

control methods not requiring defined phase, such as biological torque estimation [23], [24].

It's likely that the MLE could be further improved with more participants in the training data, more spline timing parameters in the training data (e.g. the stance percentage mapped to peak torque), and the introduction of data from the contralateral limb. While the overground training likely added useful natural variability to the model, the training protocol could also be designed to include accelerations/decelerations, or explicit variations in step length/step width. Metabolic effects of the exoboots may depend on training and adaptation; a user's ability to synergistically activate their plantarflexors with the device may improve over time, and required training may be highly user-specific [25], [26]. Understanding how simple time-based gait state estimation versus learned kinematic-based gait state estimation affect user adaptation is an important direction for future research.

A major limitation in this study is that the TBE parameters were not perfectly tuned or optimized, and the TBE is an imperfect control condition (i.e., there is no perfect ground truth). The TBE gyro-based heel strike detector was tuned based on high-speed video but was biased to be earlier than the FSRs indicated. Similarly, the TBE toe off was hardcoded at 62% of the stride rather than the ~60% that we found from the FSRs, which likely explains the delayed average toe off. The delayed toe off, in combination with the TBE's expected over-

prediction of stance duration during treadmill accelerations, caused peak torque and reel-out to occur exceptionally late. Anecdotally, this induced visible compensatory mechanisms to prevent toe scuffing. We wanted both controller conditions to be user-independent, but the heel strike delay parameter and toe off percentage would likely have been improved had they been tuned on an individual-basis. Also, the use of two previous strides in the TBE, compared to the greater number used in other (albeit steady-state) studies, allowed quicker controller adaptation to gait speed, but may not have adequately filtered natural stride-to-stride variability.

A large portion of the total error was likely due to noise in the FSR signals, stemming from placement inconsistencies and sensor degradation. Future experiments could use a force-instrumented treadmill to obtain more reliable estimates of heel strike and toe off events, though we noted substantial cross-stepping during our validation tests. With only three validation participants, it is unknown whether the metabolic trends would apply to a larger sample of users, and we do not have statistical power to test this trend. Finally, in approximately 10% of the training and validation trials, the exoboot firmware had critical errors that caused the exoboots to go unstable; these experiences may have prevented subjects from fully trusting the devices, and they may have altered their gait to ensure stability.

Though the participants in the validation trials were on average 8 kg lighter and 8 cm taller than in the training group (with one validation participant being the tallest and another being the lightest of all participants), the model successfully extrapolated to their anthropometrics. However, it's unknown how well the model would perform with people further outside of these distributions in terms of height, weight, age, or ambulatory ability. In particular, we expect user-dependent models would need to be trained for individuals with asymmetric gait impairments, and using the same model for both the left and right sides may fail.

In future studies, we plan to add ambulation modes such as ramps and stairs, as well as a high-level state machine capable of predicting activity mode. The MLE we introduced here is also suitable for user-specific optimization of metabolic cost and/or preferred walking speed.

VII. ACKNOWLEDGMENT

The authors thank Inseung Kang and Heejoo Jin for their contributions analyzing pilot data and high-level discussions of the deep learning model, as well as Katherine Summers for help with data collection and mechanical characterization of the exoboots.

VIII. REFERENCES

- [1] L. M. Mooney, E. J. Rouse, and H. M. Herr, "Autonomous exoskeleton reduces metabolic cost of human walking," vol. 11, no. 1, pp. 1–5, 2014.
- [2] J. Zhang, P. Fiers, K. A. Witte, R. W. Jackson, K. L. Poggensee, C. G. Atkeson, and S. H. Collins, "Human-in-the-loop optimization of exoskeleton assistance during walking," *Science (80-.)*, vol. 356, no. 6344, pp. 1280–1283, 2017.

- [3] G. S. Sawicki, O. N. Beck, I. Kang, and A. J. Young, "The exoskeleton expansion: Improving walking and running economy," *J. Neuroeng. Rehabil.*, vol. 17, no. 1, pp. 1–9, 2020.
- [4] R. W. Jackson and S. H. Collins, "An experimental comparison of the relative benefits of work and torque assistance in ankle exoskeletons," *J. Appl. Physiol.*, vol. 119, no. 5, pp. 541–557, 2015.
- [5] P. Malcolm, W. Derave, S. Galle, and D. De Clercq, "A Simple Exoskeleton That Assists Plantarflexion Can Reduce the Metabolic Cost of Human Walking," *PLoS One*, vol. 8, no. 2, pp. 1–7, 2013.
- [6] E. Schaffalitzky, P. Gallagher, M. MacLachlan, and S. T. Wegener, "Developing consensus on important factors associated with lower limb prosthetic prescription and use," *Disabil. Rehabil.*, vol. 34, no. 24, pp. 2085–2094, 2012.
- [7] R. D. Gregg, T. Lenzi, L. J. Hargrove, and J. W. Sensinger, "Virtual constraint control of a powered prosthetic leg: From simulation to experiments with transfemoral amputees," *IEEE Trans. Robot.*, vol. 30, no. 6, pp. 1455–1471, 2014.
- [8] S. Galle, P. Malcolm, S. H. Collins, and D. De Clercq, "Reducing the metabolic cost of walking with an ankle exoskeleton: interaction between actuation timing and power," *J. Neuroeng. Rehabil.*, vol. 14, no. 1, pp. 1–16, 2017.
- [9] C. L. Lewis and D. P. Ferris, "Invariant hip moment pattern while walking with a robotic hip exoskeleton," *J. Biomech.*, vol. 44, no. 5, pp. 789–793, 2011.
- [10] B. T. Quinlivan, S. Lee, P. Malcolm, D. M. Rossi, M. Grimmer, C. Sivi, N. Karavas, D. Wagner, A. Asbeck, I. Galiana, and C. J. Walsh, "Assistance magnitude versus metabolic cost reductions for a tethered multiarticular soft exosuit," *Sci. Robot.*, vol. 4, no. January, 2017.
- [11] I. Kang, P. Kunapuli, and A. J. Young, "Real-Time Neural Network-Based Gait Phase Estimation Using a Robotic Hip Exoskeleton," *IEEE Trans. Med. Robot. Bionics*, vol. 2, no. 1, pp. 28–37, 2019.
- [12] I. Kang, D. D. Molinaro, S. Duggal, Y. Chen, P. Kunapuli, and A. J. Young, "Real-Time Gait Phase Estimation for Robotic Hip Exoskeleton Control during Multimodal Locomotion," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 3491–3497, 2021.
- [13] X. Tu, M. Li, M. Liu, J. Si, He, and Huang, "A Data-Driven Reinforcement Learning Solution Framework for Optimal and Adaptive Personalization of a Hip Exoskeleton," *arXiv*, 2020.
- [14] D. Lee, I. Kang, D. D. Molinaro, A. Yu, and A. J. Young, "Real-time user-independent slope prediction using deep learning for modulation of robotic knee exoskeleton assistance," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 3995–4000, 2021.
- [15] W. Van Dijk, C. Meijneke, and H. Van Der Kooij, "Evaluation of the achilles ankle exoskeleton," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 25, no. 2, pp. 151–160, 2017.
- [16] J. M. Caputo and S. H. Collins, "A universal ankle-foot prosthesis emulator for human locomotion experiments," *J. Biomech. Eng.*, vol. 136, no. 3, p. 035002, 2014.
- [17] C. Walsh, "Human-in-the-loop development of soft wearable robots," *Nat. Rev. Mater.*, vol. 3, no. 6, pp. 78–80, 2018.
- [18] L. M. Mooney, E. J. Rouse, and H. M. Herr, "Autonomous exoskeleton reduces metabolic cost of walking during load carriage," *J. Neural Eng. Rehabil.*, 2014.
- [19] K. E. Gordon and D. P. Ferris, "Learning to walk with a robotic ankle exoskeleton," *J. Biomech.*, vol. 40, no. 12, pp. 2636–2644, 2007.
- [20] J. M. Brockway, "Derivation of formulae used to calculate energy expenditure in man," *Hum. Nutr. Clin. Nutr.*, pp. 463–471, 1987.
- [21] M. S. Orendurff, J. A. Schoen, G. C. Bernatz, A. D. Segal, and G. K. Klute, "How humans walk: Bout duration, steps per bout, and rest duration," *J. Rehabil. Res. Dev.*, vol. 45, no. 7, pp. 1077–1090, 2008.
- [22] H. G. Kang and J. B. Dingwell, "Separating the effects of age and walking speed on gait variability," *Gait Posture*, vol. 27, no. 4, pp. 572–577, 2008.
- [23] D. D. Molinaro, I. Kang, J. Camargo, and A. J. Young, "Biological Hip Torque Estimation using a Robotic Hip Exoskeleton," *Proc. IEEE RAS EMBS Int. Conf. Biomed. Robot. Biomechatronics*, vol. 2020–Novem, pp. 791–796, 2020.
- [24] H. C. Siu, J. Sloboda, R. J. McKindles, and L. A. Stirling, "A Neural Network Estimation of Ankle Torques From Electromyography and Accelerometry," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 29, pp. 1624–1633, 2021.
- [25] Y. Acosta-Sojo and L. Stirling, "Individuals differ in muscle activation patterns during early adaptation to a powered ankle exoskeleton," *Appl. Ergon.*, vol. 98, no. August 2021, p. 103593, 2022.
- [26] K. L. Poggensee and S. H. Collins, "How adaptation, training, and customization contribute to benefits from exoskeleton assistance," *Sci. Robot.*, vol. 6, no. 58, 2021.