Annals of Biomedical Engineering
https://doi.org/10.1007/s10439-023-03391-y

REVIEW

User- and Speed-Independent Slope Estimation for Lower-Extremity Wearable Robots

Jairo Y. Maldonado-Contreras[1,2] · Krishan Bhakta[1] · Jonathan Camargo[1] · Pratik Kunapuli[3] · Aaron J. Young[1,2]

Received: 16 January 2023 / Accepted: 17 October 2023

Abstract

Wearable robots can help users traverse unstructured slopes by providing mode-specific hip, knee, and ankle joint assistance. However, generalizing the same assistance pattern across different slopes is not optimal. Control strategies that scale assistance based on slope are expected to improve the feel of the device and improve outcome measures such as decreasing metabolic cost. Prior numerical methods for slope estimation struggled to estimate slopes at variable walking speeds or were limited to a single estimation per gait cycle. This study overcomes these limitations by developing machine-learning methods that yield continuous, user- and speed-independent slope estimators for a variety of wearable robot applications using an able-bodied wearable sensor dataset. In a leave-one-subject-out cross-validation ($N=9$), four-phase XGBoost regression models were trained on static-slope (fixed-slope) data and evaluated on a novel subject's static-slope and dynamic-slope (variable-slope) data. Using all available sensors, we achieved an average error of 0.88° and 1.73° mean absolute error (MAE) on static and dynamic slopes, respectively. Ankle prosthesis, knee-ankle prosthesis, and hip exoskeleton sensor suites yielded average errors under 2° MAE on static and dynamic slopes, except for the ankle prosthesis and hip exoskeleton cases on dynamic slopes which yielded an average error of 2.2° and 3.2° MAE, respectively. We found that the thigh inertial measurement unit contributed the most to a reduction in average error. Our findings suggest that reliable slope estimators can be trained using only static-slope data regardless of the type of lower-extremity wearable robot.

Keywords Exoskeleton · Prosthesis · Machine learning · Slope estimation · Intention recognition

Introduction

Analyses of able-bodied slope ambulation reveal that human biomechanics vary with slope. Specifically, range-of-motion, peak flexion, and peak extension angles for ankle, knee, and hip joints vary across slopes [1–4]. On increasing positive slopes, across different walking speeds, able-bodied biomechanics shows there is an increase in average individual leg total positive work—primarily generated by the ankle. Additionally, on decreasing negative slopes, across different walking speeds, there is an increase in average individual leg total negative work—primarily generated by the knee [4].

Slope-specific biomechanics are also observed in users of wearable robots [5–8]. Seo et al. provided able-bodied hip exoskeleton users with equal levels of output torque as they walked on slopes of 0, + 5, and + 10% grade. Metabolic costs were reduced by 13.5, 15.5, and 9.8% at 0, + 5, and + 10% grade, respectively, relative to a no-exoskeleton condition. Although output torque was fixed, hip joint kinematics were affected by the slope which led to an increase in exoskeleton power with increasing slopes [5]. Franks et al. sought to optimally assist able-bodied exoskeleton users on slopes using human-in-the-loop optimization algorithms. For all slopes (+ 5°, + 10°, and + 15°), metabolic costs were reduced by at least 50% relative to a no-assistance condition. Exoskeleton hip and knee extension torque were shown to

Associate Editor Sean S. Kohles oversaw the review of this article.

✉ Jairo Y. Maldonado-Contreras
jym3@gatech.edu

[1] Woodruff School of Mechanical Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA

[2] Institute for Robotics and Intelligent Machines, Georgia Institute of Technology, Atlanta, GA 30332, USA

[3] General Robotics Automation Sensing and Perception Laboratory, University of Pennsylvania, Philadelphia, PA 19104, USA

Published online: 06 November 2023

increase with steeper inclines [6]. Maclean et al. found that able-bodied knee exoskeleton users achieved a 4.2% reduction in metabolic cost when walking on a + 15° incline with a backpack load. Exoskeleton assistance was tuned based on user preference and no joint torque information was provided [7]. Sup et al. also chose to tune knee-ankle prosthesis assistance based on user preference. Knee and ankle joint torque were shown to increase with steeper inclines (0°, + 5°, and + 10°) for a transfemoral amputee user [8]. These studies reveal that users of wearable robots benefit from or prefer slope-specific assistance. Therefore, methods of slope estimation are needed to appropriately scale assistance during wearable robot usage.

Ambulatory slope estimation has primarily been attempted using tilt estimation [9–11], direct integration [12–15], kinematic modeling [16–18], extended Kalman filters (EKF) [19], and machine learning [20–22] methods. In tilt estimation, slope angle is determined by calculating the inverse tangent of the ratio between the normal and tangential linear acceleration obtained from an accelerometer placed at the foot. This computation takes place during midstance when the foot is flat on the ground [9]. Direct integration methods integrate foot or shank IMU signals to track linear and angular displacements from which slope can be computed. To avoid drift, integration initial values are reset during midstance when the foot and velocities are assumed to be zero [12–15]. Kinematic modeling methods use joint angle information and kinematic relationships to calculate slope during foot-to-ground contact [16, 17]. Some studies involving direct integration or kinematic modeling reported that slope estimation performance was negatively impacted by walking speed [13, 18]. Medrano et al. introduced an EKF method of slope estimation that used sensor measurements obtained from an ankle exoskeleton to continuously track slopes [19]. Unlike regular Kalman filters [23], EKFs can track the state of a system (slope) even if non-linearities are present in system dynamics and sensor-state relationships. In this study, we assume linear system dynamics and utilize a Kalman filter to smooth slope estimates output by machine learning models. Machine learning methods use regression models trained with wearable sensor information to continuously estimate slope [20–22, 24, 25]. Slope estimation methods that are constrained to only estimate slope during midstance cannot keep up with changes in slope that occur outside of midstance. It is necessary to do so since able-bodied individuals show biomechanical adaptations to changes in slope during swing. During swing, hip and knee flexion angle increase as slope increases to help the individual clear the ramp angle [1]. Thus, continuous methods of estimation are preferred here. Machine-learning methods were chosen over EKF methods since machine learning models can learn inter-subject variability, interactions between speed and slope, and device-specific dynamics that impose sensor noise—processes that impede EKF implementation [19].

This paper describes machine-learning methods for continuous slope estimation and an expansion of those methods to multiple lower-limb wearable robot applications. Sensor sets were chosen based on sensors intrinsic to lower-limb wearable robots. We improve upon prior user-dependent machine-learning methods [20, 21, 24] by providing user- and speed-independent slope estimation solutions. User-independent (IND) estimators are trained with multi-subject datasets and evaluated with novel subject data [21, 24, 26] while user-dependent (DEP) estimators are trained and evaluated with subject-specific data [20, 24, 27]. IND estimators are a more practical solution for slope estimation than DEP estimators since they can generalize well to novel subjects without additional data collection. In addition, given that able-bodied biomechanics vary with walking speed [28], we varied walking speeds during data collections so that resulting slope estimators may be robust against changes in walking speed. Machine-learning architectures, feature extraction methods, Kalman filter parameters, and feature sets are optimized for the following device cases: ALL (includes all sensors), knee-ankle prosthesis (KA), ankle prosthesis (AN), and hip exoskeleton (HE) (Fig. 1).

We hypothesize that slope estimators trained with only static-slope data can estimate both static- and dynamic-slope data belonging to novel subjects with similar performance to existing methods–which would eliminate the need to collect dynamic-slope data in future implementations. In line with sensors used in direct integration methods, we anticipate distal sensors to be of higher importance to slope estimators as they provide richer foot-to-ground contact information. Lastly, we hypothesized that nonlinear models (i.e., Neural Network and XGBoost models) would outperform linear models (i.e., Linear Regression models) due to the complexity of estimating slopes at varying walking speeds—a limitation encountered when implementing numerical slope estimation methods [13, 18]. The novel contribution of this work is the development of continuous user-independent machine learning algorithms that estimate slope independent of speed for multiple wearable robot applications.

Materials and Methods

Our study consisted of nine able-bodied subjects (6 males and 3 females) with an average (mean ± 1SD) age of 21.6 ± 2.8 years, height of 1.76 ± 0.10 m, and body weight of 72.0 ± 12.4 kg. This experiment was approved by the Georgia Institute of Technology IRB.

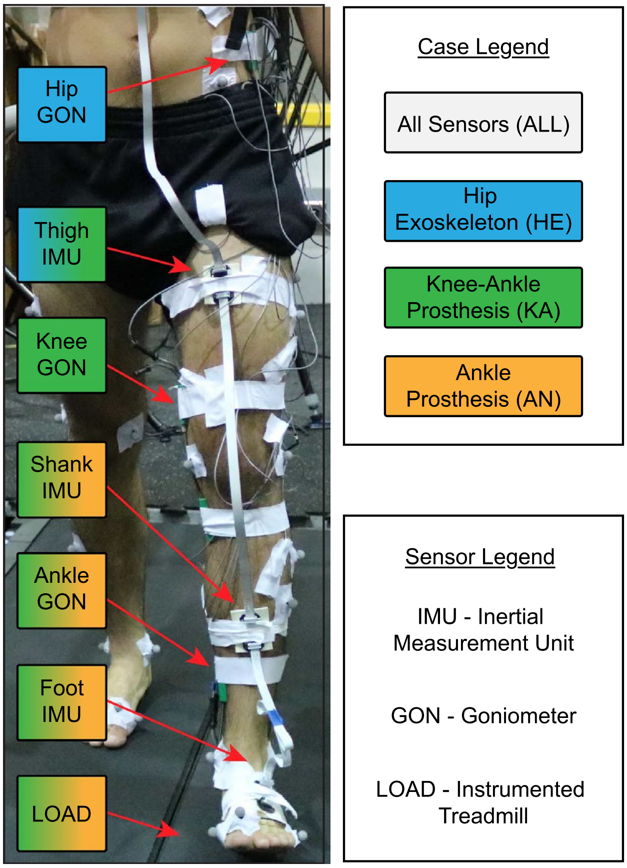

Fig. 1 Experimental setup in which an able-bodied subject is completing a trial. All sensors were placed unilaterally. Sensors are labeled above based on the abbreviations described in the sensor legend. The case legend describes the color assignments for each device case studied. The coloring assigned to each sensor shows the categorization of sensors by device case

Sensors

Subjects were unilaterally instrumented (4 right and 5 left) with three 6-axis YOST 3-Space LX Embedded inertial measurement units (IMUs) and three two-degree-of-freedom (2-DOF) Biometrics electrogoniometers (GONs). By maintaining a fair split of right- and left-instrumented subjects, we ensured that our estimators were trained to perform regardless of leg dominance or unilateral set up. Each IMU was of 16-bit resolution and ± 8 g and ± 1000°/s full scale ranges. IMUs were placed on the foot, shank, and thigh. Each GON was of ± 0.1° resolution and ± 180° full scale range. GONs were placed at the ankle (frontal and sagittal), knee (sagittal), and hip (frontal and sagittal). Ground reaction forces and moments (LOAD) were measured using a Bertec Instrumented Treadmill (Bertec Corp., Columbus, OH, USA). The location of each sensor is shown in Fig. 1.

Data Collection

Each subject was instrumented with six sensors (3 IMUs and 3 GONs) and completed 17 walking trials: 13 static-slope and 4 dynamic-slope trials. Static-slope trials involved walking on fixed slopes at varying walking speeds. Dynamic-slope trials involved walking on variable slopes at fixed walking speeds. Slopes and walking speeds were modulated with a Bertec instrumented treadmill. Static-slope trials were collected for each slope between − 15° and + 15°, incremented by 2.5°. Slopes were fixed while walking speeds were imposed in the following order: 0.6, 0.8, 1.0, 1.2, 1.4, 1.3, 1.1, 0.9, and 0.7 m/s. Each walking speed was held constant for 10 seconds then modulated to the next speed at 0.2 m/s^2. The total length of each static-slope trial was 108 s. Four dynamic-slope trials were collected—one trial for each unique combination of walking speed (1.0 or 1.2 m/s) and slope direction (positive or negative). For each trial, walking speed was held constant while slopes were imposed in the following order: 0°, ± 1°, ± 3°, ± 5°, ± 7°, ± 9°, ± 10°, ± 8°, ± 6°, ± 4°, and ± 2°. Each slope was held constant for 10 s then modulated to the next slope at 0.1°/s. The total length of each dynamic-slope trial was 310 s. Positive and negative slope trials were collected separately due to the mechanical limitations of the Bertec treadmill.

Data Processing

IMU and GON data were collected with a Rasberry Pi 3 Model B+ at 200 Hz and 1000 Hz, respectively. Slopes, walking speeds, and six-axis ground reaction forces and moments (LOAD) were recorded at 1000 Hz using the Bertec Instrumented Treadmill. IMU and GON data were collected on the Rasberry Pi 3 using the Robot Operating System (ROS). Bertec Treadmill signals were collected on a local desktop. Signals were synced using a shared sync pin on the Rasberry Pi 3. During post-processing, for convenience to keep all data sizes the same, IMU data were up-sampled to 1000 Hz to match the sampling rate of Bertec Treadmill signals. Angular velocities of each joint were also calculated during post-processing. GON and LOAD data were filtered with a fifth-order low-pass Butterworth filter with a 20 Hz cutoff frequency. Signals from left-instrumented subjects were inverted to match the signals of right-instrumented subjects.

Data were segmented into gait cycles by thresholding the vertical ground reaction force to obtain heel contact events. Then, gait cycles were evenly divided into four phases to later generate phase-specific slope estimators [29].

Sensors were divided into the three device cases (KA, AN, and HE) based on sensors suites commonly referenced in literature (Fig. 1). A fourth case incorporating all sensors was included as the ALL case. The number of sensor channels per case totaled 34, 30, 22, and 10 for the ALL, KA, AN, and HE cases, respectively. More information regarding cases, sensors, and sensor channels is available in Tables IV and V in the Supplemental Material.

Feature Extraction

Five features were extracted per channel: mean, standard deviation, minimum, maximum, and final value [30] (Fig. 2). Features were extracted from a window of signal data every 50 ms (20 Hz). Initial features and window sizes were selected based on their demonstrated capacity to train robust mode classifiers for lower limb devices [24, 26, 26, 27, 31–33]. Features extracted from a single window served as one sample for slope estimators. Each sample was assigned a label that was the average slope of the window. Also, each sample was assigned a phase label (not used as a model input) based on the gait phase during which the last value of the window occurred.

Open-Source Dataset

The filtered data is available for public access here: https://www.epic.gatech.edu/open-source-able-bodied-slope-data/. The dataset contains 51 sensor channels. Ankle, knee, and hip joint velocities were calculated during postprocessing. Header, speed, and slope columns are also provided. Trunk IMU and electromyography data were not investigated in this study but are included in this open-source dataset.

Machine Learning Algorithms

The slope estimation capabilities of three regression models were compared: linear regression, feed-forward neural network, and XGBoost models. Linear regression is a comparatively less complex approach that fits linear models to the data. XGBoost and neural network models were selected given their previous success in the field [20, 22, 24, 34]. Neural networks are biologically inspired models that modify inputs using linear combinations of trained weights and map their sum with internodal activation functions to obtain an output of specific amplitude [35]. Neural networks have seen a large history of use in gait tasks, especially in slope estimation [20, 25]. XGBoost is a decision-tree-based ensemble algorithm that employs optimized gradient boosting techniques to sequentially build higher-performing decision trees [36]. XGBoost models have recently emerged as a leading classification algorithm for gait tasks involving fall detection [37], gait decoding [38], locomotion phase [39], and mode identification [24, 33]. This comparison between linear and non-linear machine learning models will help detail the level of complexity associated with estimating slopes at varying walking speeds.

Kalman Filter

Walking on slopes at different walking speeds introduces non-steady-state gait that can negatively impact the performance of regression-based slope estimators. Kalman filters can help offset the issue by fusing uncertain measurements with prior state information to produce better slope estimates. Kalman filter calculations were divided into prediction (Eqs. 1 and 2) and update stages (Eqs. 3 and 4).

In the prediction stage, a predefined model of the system (Eqs. 1 and 2) was used to compute the next slope estimate and slope estimate uncertainty ($p_{n+1,n}$). We chose to model the environment as a static system due to the relatively small changes in slope that occur between timesteps during static-slope and dynamic-slope trials. State (Eq. 1) and Covariance (Eq. 2) Extrapolation equations are defined as follows:

$$\hat{x}_{n+1,n} = \hat{x}_{n,n} \tag{1}$$

$$\hat{p}_{n+1,n} = \hat{p}_{n,n} + q_n \tag{2}$$

where $\hat{x}_{n+1,n}$ is the predicted slope estimate, $\hat{x}_{n,n}$ is the current slope estimate, $\hat{p}_{n+1,n}$ is the predicted slope estimate uncertainty, and $\hat{p}_{n,n}$ is the current slope estimate uncertainty, and q_n is the current process noise. A process noise is included in Eq. 2 to account for uncertainties in the system

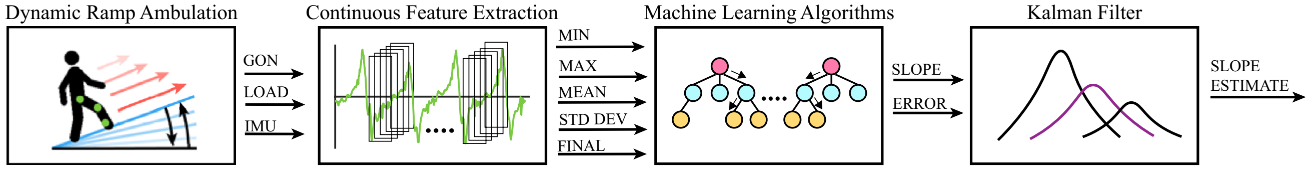

Fig. 2 Slope estimation pipeline used to continuously estimate slope during slope ambulation (offline). Signals included linear accelerations, and rotational velocities from three IMUs, joint angles from three goniometers, and ground-reaction forces and moments from treadmill force plates. Data was segmented into four phases—roughly early stance, late stance, swing flexion, swing extension. Features (minimum, maximum, mean, standard deviation, and end value) are extracted from a rolling buffer of data for a fixed window size. A new row of data is added to the buffer at 20 Hz. Extracted features were fed into slope and error models to obtain slope and uncertainty estimates. Lastly, a Kalman filter smooths final slope estimates given model outputs

dynamic model and produce estimation errors. Here, process noise is unitless because it models multiple error-inducing factors such as modeling errors, unmodeled dynamics, and external disturbances.

In the update stage, current and previous state information is used to update current Kalman variables (Eqs. 3, 4, 5). The Kalman Gain, State, and Covariance Update equations are defined as follows:

$$K_n = \frac{\widehat{p}_{n,n-1}}{\widehat{p}_{n,n-1} + r_n} \tag{3}$$

$$\widehat{x}_{n,n} = \widehat{x}_{n,n-1} + K_n\left(z_n - \widehat{x}_{n,n-1}\right) \tag{4}$$

$$\widehat{p}_{n,n} = (1 - K_n)\widehat{p}_{n,n-1} \tag{5}$$

where K_n is the current Kalman Gain, $\widehat{x}_{n,n}$ is the current slope estimate, $\widehat{x}_{n,n-1}$ is the previous slope estimate, z_n is the current slope measurement, r_n is the current slope measurement uncertainty, $\widehat{p}_{n,n}$ is the current slope estimate uncertainty, and $\widehat{p}_{n,n-1}$ is the previous slope estimate uncertainty. Measurements, estimates, and uncertainties were measured in degrees.

Analyses

The following analyses were completed for each case using a leave-one-subject-out cross-validation: (1) Model Optimization and (2) Window Size optimization, (3) Kalman Filter Optimization, (4) Sensor Selection, and (5) Slope Estimation. Only static-slope data were used during steps 1-3. Static-slope and dynamic-slope trial data were used to evaluate the optimized pipeline during steps 4 and 5. Models were evaluated using mean absolute error (MAE) ± standard error of the mean (SEM).

Model Optimization

The slope estimation performances of the following three regression models were compared using a window size of 250 ms: linear estimator (LNR) from the Sklearn library, feed-forward neural network (NN) from the Keras library, and extreme gradient boosting decision tree (XGB) from the XGboost library (Fig. 3).

A total of four machine-learning models (one model for each of the four gait phases) were optimized and evaluated for each model type (e.g., XGB). Phase-specific errors were averaged to attain a comprehensive error for each trial. For

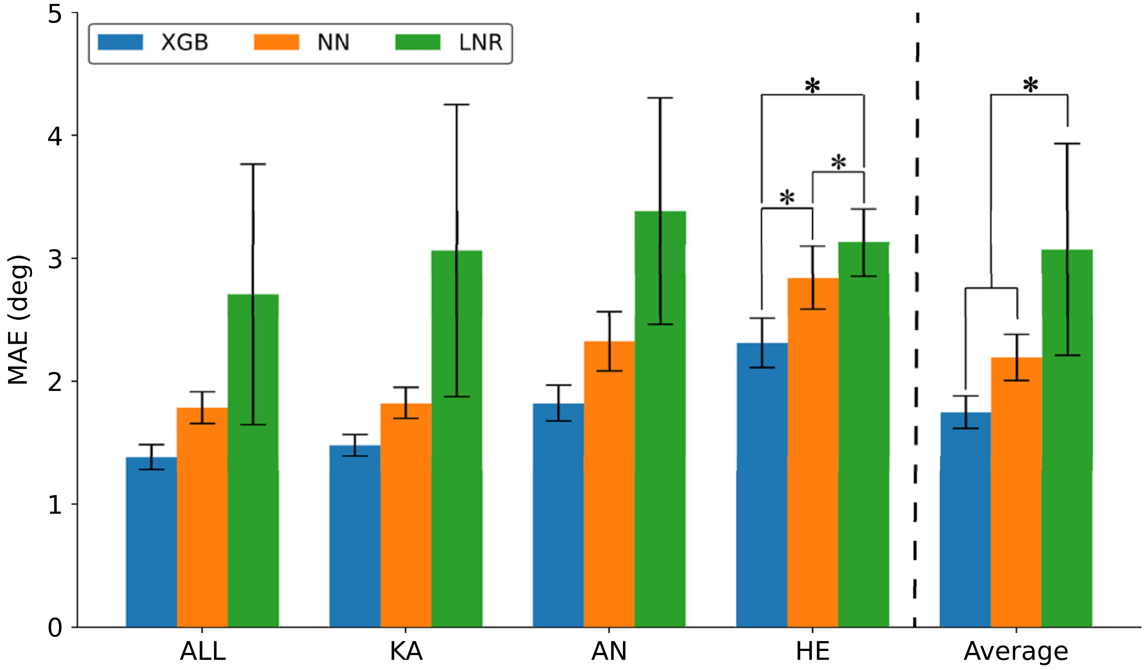

Fig. 3 Cross-subject user-independent performance comparison of optimized XGB, optimized NN, and LNR models on static-slope estimation for each. Hyperparameters that yielded the lowest average MAE errors (across all four phases) were selected as the optimal set. A window size of 250 ms and increment size of 50 ms was used during this analysis. When averaged across all cases, slope estimation errors of 1.75° ± 0.13°, 2.19° ± 0.18°, and 3.07° ± 0.86° MAE were achieved for XGB, NN, and LNR, respectively. Given that the XGB model was shown to outperform all other models, the XGB model and its optimized hyperparameters were used in later analysis. Hyperparameters of the optimal XGB model for each case listed in Table I in the Supplemental Material. Error bars represent the ± SEM. Asterisks indicate statistical significance ($p < 0.05$)

XGB and NN, each combination of hyperparameters was explored. LNR models do not have hyperparameters to optimize. Models were trained with only static-slope data. Static-slope data of each training subject ($N = 8$) were split into training (80%) and validation (20%) sets. Then, these subject-specific sets were combined to form a multi-subject training and validation set. The validation set was used to implement an Early Stopping criterion of 5 rounds while training slope estimators. Slope estimators were evaluated using all the static-slope data of the testing subject ($N = 1$). The resulting optimized XGB model hyperparameters can be found in Table I in the Supplemental Material.

Window Size Optimization

The optimal window size was determined by comparing the performances of optimized XGB models (obtained from the Model Optimization analysis) across 50, 100, 150, 200, 250, 300, 350, 400, 450, and 500 ms window sizes. Identical training and testing methods to those mentioned in Model Optimization were used. The only difference was that this analysis was conducted using only XGB models and window size was not fixed. Optimized window sizes are listed in Table II in the Supplemental Material.

Kalman Filter Optimization

Optimized 4-phase XGB models and window sizes were used during the optimization of Kalman filter process noise. Process noise is an uncertainty measurement of our modeled system dynamics. Process noise can significantly impact the effectiveness of a Kalman filter. To determine the process noise of our system dynamics we chose to tune process noise to minimize estimation error. We ran a leave-one-subject-our cross-validation across multiple values of process noise to determine the process noise that best smooths our slope estimates. Process noise values ranging from 0.1 to 1e−10 (incremented by one order of magnitude) were tested.

Given that the studied slope estimators do not output uncertainty estimates and the Kalman filter requires a measure of measurement uncertainty, error estimators were trained to output uncertainty estimates (absolute error) using the same model inputs as slope estimators. An error estimator was trained for each slope estimator (one per gait phase) using identical model architectures and window sizes.

Static-slope data of each training subject were split into four sets: a slope training set (80%), slope validation set (10%), error training set (5%), and an error validation set (5%). Subject-specific sets were combined to form multi-subject sets. The resulting sets were used to train slope and error estimators with an Early Stopping criterion of 5 rounds. An overview of slope and error model training is shown in Fig. A1 in the Supplemental Material.

Upon training a total of 8 estimators (4 slope and 4 error), a Kalman filter was evaluated on the testing subject's static-slope data using a selected value for process noise. Prior slope estimate and variance were initialized to 0 and 1, respectively. As test data was looped through, depending on the phase, a switch between phase-specific slope and error estimator pairs would occur. As a result of this switching logic, a single error was computed instead of averaging phase-specific errors. Updated Kalman filter parameters were maintained between gait phases. This process was repeated for each process noise tested. Optimized process noises are listed in Table II in the Supplemental Material.

Slope Estimation

For each case, using all case-specific sensors, the optimized estimation pipeline was evaluated through a leave-one-subject-out cross-validation. Like the Sensor Selection analyses, models were trained on only static-slope data belonging to training subjects and evaluated on static-slope and dynamic-slope data belonging to the testing subject. Results are shown in Fig. 4.

An additional analysis was conducted to evaluate the same models on more realistic test profiles. In addition to testing on static-slope trials (Static) and dynamic-slope trials (Dynamic #1), a third test set named Dynamic #2 was created by concatenating static-slope trials with sudden jumps in slope. The Static test set consisted of all 13 static-slope trials concatenated from smallest (− 15°) to largest slope (+ 15°). Each static-slope trial was evaluated individually, then the results were concatenated in Fig. 5A. The Dynamic #1 test set consisted of all 4 dynamic-slope trials concatenated from slowest (positive and negative slopes at 1.0 m/s) to largest speed (positive and negative slopes at 1.2 m/s). Each dynamic-slope trial was evaluated individually, then the results were concatenated in Fig. 5B. The Dynamic #2 test set consisted of all 13 static-slope trials concatenated and ordered in such a way that slopes were staggered by 5°, which more accurately mimicked real-world slope ambulation scenarios (Fig. 5C). Unlike Static and Dynamic #1 where each trial was individually evaluated, Dynamic #2 was treated as one large trial during evaluations. This meant that Kalman filter parameters (i.e., prior slope estimate and variance) were not reinitialized between slopes/trials. The impact of adding dynamic-slope data in the training set was also investigated (see Fig. A5 in the Supplemental Material).

Sensor Selection

A forward sensor selection analysis was conducted to determine the most critical sensors for slope estimation. In its first iteration, the forward sensor selection algorithm trains and tests a slope estimator (i.e., an optimized 4-phase XGB

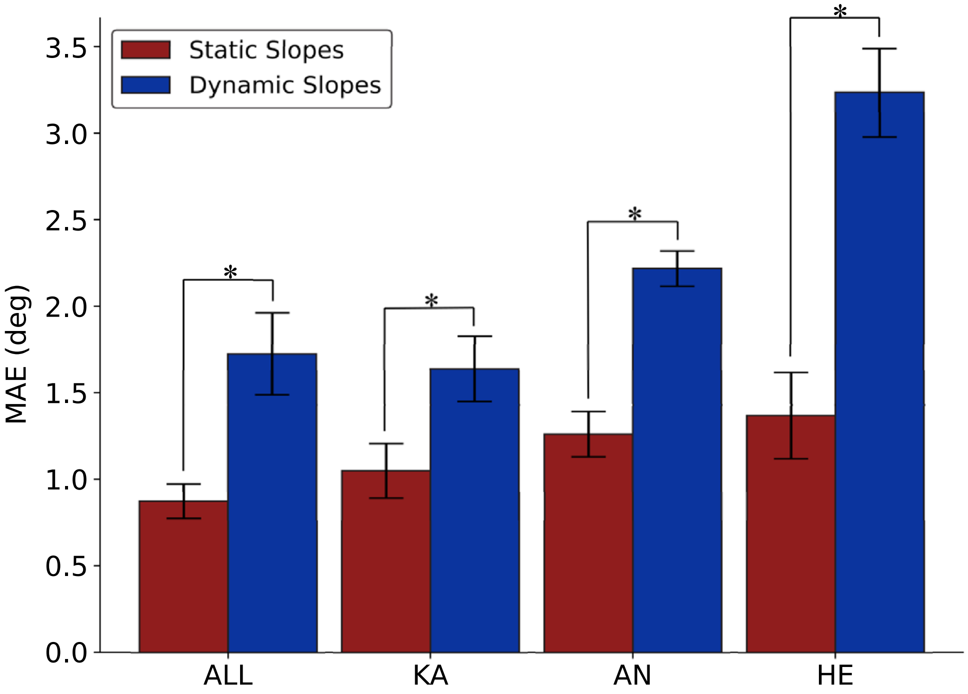

Fig. 4 Cross-subject and user-independent performance of optimized XGB models on static- and dynamic-slopes across cases. Each subject's static-slope and dynamic-slope data was evaluated by optimized XGB models and a Kalman filter trained with static-slope data belonging to all other subjects. Mean-absolute errors of slope estimates were averaged across subjects and plotted above for each case. Slope estimation errors of 0.88° ± 0.10°, 1.05° ± 0.16°, 1.26° ± 0.13°, and 1.37° ± 0.25° MAE were achieved for static slopes. Errors of 1.73° ± 0.24°, 1.64° ± 0.19°, 2.22° ± 0.10°, and 3.24° ± 0.26° MAE were achieved for dynamic slopes. Error bars represent the SEM. Asterisks indicate statistical significance ($p < 0.05$)

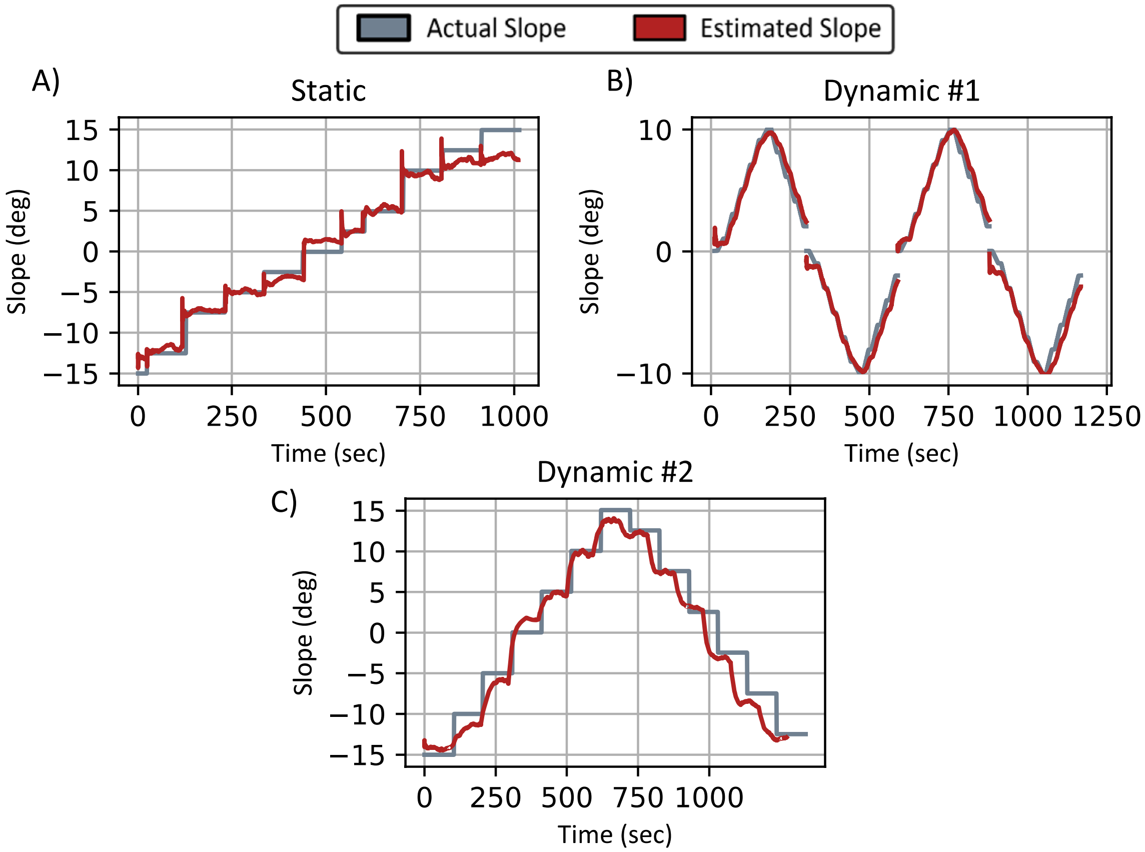

Fig. 5 Slope tracking of the Static (**a**), Dynamic #1 (**b**), and Dynamic #2 (**c**) test sets. Dynamic #2 is comprised of reordered static-slope data. Slopes are ordered as follows: − 15°, − 10°, − 5°, 0°, + 5°, + 10°, + 15°, + 12.5°, + 7.5°, + 2.5°, − 2.5°, − 7.5°, and − 12.5°. This ordering of static-slope trials is meant to simulated real-world ramp ambulation where large changes of slope occur instantaneously. A Kalman filter is used to filter estimates produced by optimized, user-independent, phase-specific slope estimators

model paired with a Kalman filter) using only one sensor's features. This process is repeated for all available sensors. The sensor that yields lowest error is considered the "best" sensor and is used in the next iteration. Successive iterations test combinations of the "best" sensors with each remaining sensor until all combinations are exhausted. Sensors were selected based on the lowest average of cross-subject error. Identical training and testing methods to those mentioned in Model Optimization were used with the addition of dynamic-slope test data. Sensor selection results are shown in Fig. 6.

An additional analysis was conducted to assess the impact of individual sensors and sensor channels. This was a shorter analysis in which models were trained and evaluated on only static-slope data. Results are available in Fig. A4 in the Supplemental Material.

Statistical Analysis

A two-way repeated-measures ANOVA ($p < 0.05$) was used to determine the effect of optimized regression model choice on static-slope estimation for each case. In this analysis, case, model type, and subject were defined as independent variables, and MAE was the dependent variable. For each case, a one-way repeated measures ANOVAs ($p < 0.05$) was used to compare model performance across static-slope and dynamic-slope trial types. Trial type and subject were defined as independent variables and MAE error was the dependent variable. Similarly, multiple one-way repeated measures ANOVAs ($p < 0.05$) were used to determine the significance of error drops due to adding sensors to the sensor suite. Sensor and subject were defined as independent variables.

Results

The comparison between optimized XGB, NN, and LNR models revealed that non-linear models (XGB and NN) yielded significantly ($p < 0.05$) lower cross-case average errors (XGB; 1.75° ± 0.13° and NN; 2.19° ± 0.18° MAE) than linear models (LNR; 3.07° ± 0.86° MAE) (Fig. 3). This result falls in line with our hypothesis that non-linear models would outperform linear models. Only the HE case yielded the same result. Across all cases, XGB models achieved lower errors than NN and LNR models. As a result, the XGB model was selected for subsequent analyses. XGB hyperparameter optimization results are shown in Tables I in the Supplemental Material.

An optimization routine of window size was conducted. All cases yielded an optimal window size of 450 ms. A Kalman filter optimization across cases resulted in process noise values of 1e−5, 1e−5, 1e−6, and 1e−6 for the ALL, KA, AN, and HE cases, respectively. Optimal window sizes and process noises are listed in Table II in the Supplemental Material.

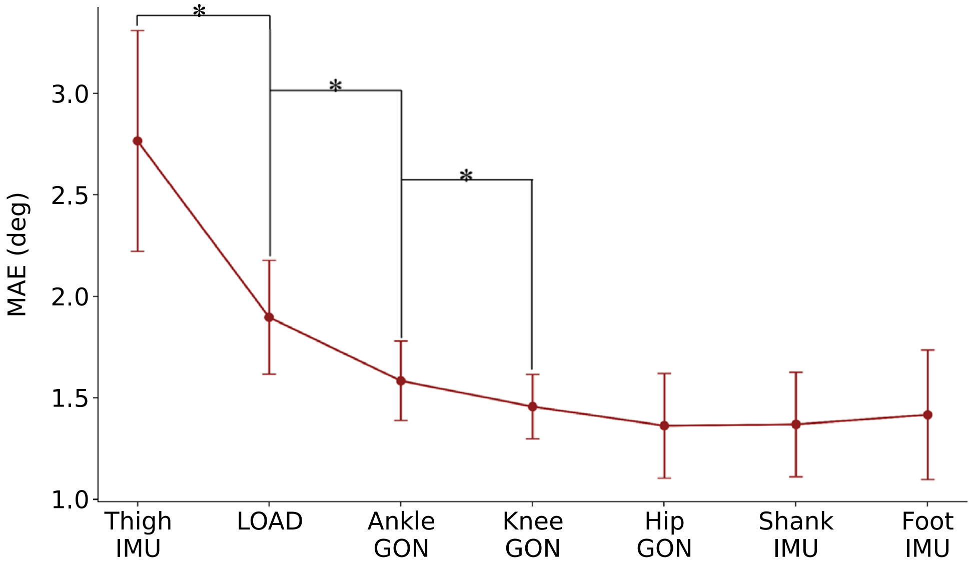

Fig. 6 Cross-subject and user-independent sequential forward sensor selection for the ALL case using an optimized XGB model. Sensors were selected based on the average MAE obtained from testing each model on a novel subject's static-slope and dynamic-slope data. Sensors are listed (left to right) in the order they were selected by the forward sensor selection algorithm (i.e., Thigh IMU, Thigh IMU + LOAD, Thigh IMU + LOAD + Ankle GON). Slope estimation errors from left to right were 2.77° ± 0.54°, 1.90° ± 0.28°, 1.58° ± 0.20°, 1.46° ± 0.16°, 1.36° ± 0.26°, 1.37° ± 0.26°, and 1.42° ± 0.32° MAE. Error bars represent the ± SEM. Asterisks indicate statistical significance ($p < 0.05$)

Optimal XGB models, window sizes, and process noises were used to evaluate slope estimation performance across static and dynamic slopes. For static slopes, slope estimation for ALL, KA, AN, and HE cases respectively achieved cross-subject error rates of 0.88° ± 0.10°, 1.05° ± 0.16°, 1.26° ± 0.13°, and 1.37° ± 0.25° MAE. For dynamics slopes, cross-subject error rates were 1.73° ± 0.24°, 1.64° ± 0.19°, 2.22° ± 0.10°, and 3.24° ± 0.26° MAE. A bar plot of these results is shown in Fig. 4. A statistical comparison between static-slope and dynamic-slope estimation performance revealed that for each case estimating static slopes yielded a significantly ($p < 0.05$) lower error than dynamic slopes. A representative plot for static- and dynamic-slope tracking is shown in Fig. 5. Most of the errors experienced in the ALL case occurred on the extreme slopes (− 12.5° and + 12.5°) and between − 2.5° and 0° (Fig. A2 in the Supplemental Material). With respect to walking speed, higher errors were found at the lower speeds of 0.6–0.8 m/s (Fig. A3 in the Supplemental Material).

The forward sensor selection for the ALL case is shown in Fig. 6 and reveals a decline in estimation errors when specific sensors are added to the sensor pool. The Thigh IMU alone achieved an error of 2.77° ± 0.54° MAE. Sequentially, adding LOAD (− 0.87° MAE), Ankle GON (− 0.32° MAE), and Knee GON (− 0.12° MAE) sensors resulted in a significant reduction of error. After Knee GON, additional sensors did not significantly affect error.

Three additional analyses were conducted to (1) investigate the performance of XGB models on more realistic slope profiles, (2) assess the benefit of adding dynamic-slope data to the training set, and (3) examine the contribution of sensor channels. The results of analysis 1 and 2 are shown in Fig. A5 in the Supplemental Material. Results for analysis 3 are shown in Fig. A4 in the Supplemental Material.

Discussion

The results indicate that strong performance (less than 2° of error for static-slope estimation) can be achieved in a user- and speed-independent manner regardless of the sensor configuration for the 3 different wearable robot test cases (Fig. 4). In prior work, continuous slope estimation achieved error rates of less than 1° using direct integration [15], 2.4° RMSE using EKF [19], and 1.25° [22], 1.3° [20], and 1.5° [21] RMSE using machine-learning. Notably, our methods achieved comparable slope estimation errors under difficult circumstances—across an expansive set of static and dynamic slopes, at variable walking speeds, with novel subjects. We found that on average, across all cases and subjects, XGBoost and neural network models significantly outperformed linear regression models. Perhaps, linear models trained on only static-slope data could not cope with the non-linearity and variability in data presented by variable walking speeds, dynamic slopes, and novel users. Across all model comparisons, the XGBoost model performed the best, with the lowest errors occurring for cases containing the most sensors (Fig. 3). Evaluating models on static slopes yielded significantly lower error than evaluating on dynamic slopes (Fig. 4). This is likely due to not including any dynamic-slope data in the training set. A separate analysis on the impact of adding dynamic-slope data to the training set resulted in a 49% reduction in error when evaluating on Dynamic #1 (Fig. A5). This drop in error is reasonable as Dynamic #1 is a concatenation of the testing subject's dynamic-slope data—data with the same slope profile as the data added to the training set. Additionally, errors increased across the board when evaluating on more realistic slope profiles (i.e., Dynamic #2). This may be due to the sudden jumps in slope within the profiles. The Kalman filter was not modeled to handle large changes in slope. Our system model assumes that slope does not change between time steps (50 ms). This holds true for individual static-slope trials and is safe to assume for dynamic-slopes that vary at 0.1 °/s. This static assumption may not hold up for Dynamic #2 profiles shown in Fig. 5C.

A forward sensor selection on the ALL case revealed that Thigh IMU, Ankle GON, and LOAD sensors contributed the most to a reduction in slope estimation error. The inclusion of the Ankle GON and LOAD sensors in this top 3 list supports our hypothesis that distal sensors would provide the most important information to a slope regression model. The importance of the Thigh IMU could be explained by the fact that walking across steeper slopes requires more hip flexion movement [1–4] which is represented within Thigh IMU data. A deeper look into the contributions of LOAD channels showed that training on force data yielded a lower error than training on moment data or training on both force and moment data. For GON sensors, a combination of ankle, knee, hip GONs yielded a lower error than any individual GON sensor. For IMU sensors, the Thigh IMU yielded a lower error than any other individual IMU sensor or combination of all IMUs (Fig. A4). Across all selection analyses, the lowest error still occurred with the combination of Thigh IMU, Ankle GON, and LOAD. These results can help inform the selection of sensors during wearable robot development. The key result of our analysis is that to have reasonable performance, more than a single sensor is needed in wearable robotic applications for user-independent slope prediction, as any single sensor has relatively poor performance.

One limitation of this study is the offline nature of the analyses. Real-time wearable robot usage can experience drops in computing performance or data communication which can result in higher errors. It is also important to mention that real-time sensor sampling rates are typically lower than those presented in this study. Though, it may be the case that feature extraction, which compresses higher frequency information, can help mitigate differences in sampling rates. Additionally, the assistance delivered by wearable robots can physically deform the device itself and augment a user's gait in such a way that it introduces sensor noise—noise that is not experienced in this study. Furthermore, wearable robot users of different age, height, weight, or level of impaired gait may present variability in data that is not represented in our datasets. Differences in sensor placement/locations between subjects and wearable robots may also affect slope estimation performance. Also, we assumed that electrogoniometers and force plates were adequate substitutes for encoders and loadcells used in wearable robots. Unlike electrogoniometers, encoders face issues with deflection and backlash that can have a detrimental effect on the accuracy of readings. Loadcells are likely less accurate than a force plate and are commonly located higher up on prostheses. Though, a comparison between an iPecs 6-DOF load cell and floor-mounted force plates found comparable results between measured forces (3.4% RMSE) and moments (5.4% RMSE) while walking on a prosthesis [40]. Lastly, our tests were limited to a treadmill and did not include "real-world" walking data which may represent the best-case scenario. We attempted to mitigate this issue by evaluating our models on more realistic slope profiles. It is important to mention that the treadmill also presented limitations in that varying slopes required the treadmill to be unclamped and allowed slight oscillation on each heel-strike which could affect true slope readings and introduce sensor noise.

Future work will aim to verify our assumptions by implementing and testing our findings on lower-limb devices in real-time using the open-source able-bodied dataset detailed in this paper. If these methods are validated for wearable robot usage, the open-source dataset would enable researchers to train and implement device-specific slope estimators without the need to collect additional data. The implementation of slope estimators during device usage can help achieve optimal assistance on slopes. As wearable robots become more aware of the surrounding environment, scalable assistance will lead to an improved quality of life in users. Machine learning models that account for variations in walking speed paired with Kalman Filters have shown great promise in estimating static and dynamic slopes while being trained with a simple dataset. Our findings provide baseline architectures for accurately estimating slopes in lower-limb prosthesis and exoskeleton applications.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s10439-023-03391-y.

Acknowledgments This work was funded in part by NSF National Robotics Initiative (Award #1830215) to A.J.Y., U.S. Army Natick Soldier Research, Development and Engineering Center (W911QY18C0140) to G.S.S and by the Department of Defense Congressionally Directed Medical Research Programs (DoD CDMRP) Award No. W81XWH-17-1-0031. This study was also supported by graduate fellowships from the Ford Foundation, NSF Accessibility, Rehabilitation, and Movement Science Traineeship Program, and the National GEM Consortium. The content is solely the responsibility of the authors and does not necessarily represent the official views of the funding agencies listed.

Declarations

Conflict of interest There is no conflict of interest reported by the authors.

References

1. Camargo, J., A. Ramanathan, W. Flanagan, and A. Young. A comprehensive, open-source dataset of lower limb biomechanics in multiple conditions of stairs, ramps, and level-ground ambulation and transitions. *J. Biomech.*119:110320, 2021. https://doi.org/10.1016/j.jbiomech.2021.110320.
2. Lay, A. N., C. J. Hass, and R. J. Gregor. The effects of sloped surfaces on locomotion: a kinematic and kinetic analysis. *J. Biomech.* 39(9):1621–1628, 2006. https://doi.org/10.1016/j.jbiomech.2005.05.005.
3. Lay, A. N., C. J. Hass, T. Richard Nichols, and R. J. Gregor. The effects of sloped surfaces on locomotion: an electromyographic analysis. *J. Biomech.* 40(6):1276–1285, 2007. https://doi.org/10.1016/j.jbiomech.2006.05.023.
4. Montgomery, J. R., and A. M. Grabowski. The contributions of ankle, knee and hip joint work to individual leg work change during uphill and downhill walking over a range of speeds. *R. Soc. Open Sci.*5(8):180550, 2018. https://doi.org/10.1098/rsos.180550.
5. Seo, K., J. Lee, and Y. J. Park. Autonomous hip exoskeleton saves metabolic cost of walking uphill. In: 2017 International Conference on Rehabilitation Robotics (ICORR). 2017, pp. 246–251. https://doi.org/10.1109/ICORR.2017.8009254.
6. Franks, P. W., G. M. Bryan, R. Reyes, M. P. O'Donovan, K. N. Gregorczyk, and S. H. Collins. The effects of incline level on optimized lower-limb exoskeleton assistance. 2021. 2021.09.13.460170. https://doi.org/10.1101/2021.09.13.460170.
7. MacLean, M. K., and D. P. Ferris. Energetics of walking with a robotic knee exoskeleton. *J. Appl. Biomech.* 35(5):320–326, 2019. https://doi.org/10.1123/jab.2018-0384.
8. Sup, F., H. A. Varol, and M. Goldfarb. Upslope walking with a powered knee and ankle prosthesis: initial results with an amputee subject. *IEEE Trans. Neural Syst. Rehabil. Eng.* 19(1):71–78, 2011. https://doi.org/10.1109/TNSRE.2010.2087360.
9. Park, S. K., and Y. S. Suh. Height compensation using ground inclination estimation in inertial sensor-based pedestrian navigation. *Sensors.* 11(8):8045–8059, 2011. https://doi.org/10.3390/s110808045.
10. Ulf Holmberg, W. S. An autonomous control system for a prosthetic foot ankle. *IFAC Proc. Vol.* 39(16):856–861, 2006. https://doi.org/10.3182/20060912-3-DE-2911.00147.
11. Best, T. K., C. G. Welker, E. J. Rouse, and R. D. Gregg. Data-driven variable impedance control of a powered knee-ankle prosthesis for adaptive speed and incline walking. *IEEE Trans. Robot.* 2023. https://doi.org/10.1109/TRO.2022.3226887.
12. López, A. M., D. Álvarez, R. C. González, and J. C. Álvarez. Slope estimation during normal walking using a shank-mounted inertial sensor. *Sensors.* 12(9):11910–11921, 2012. https://doi.org/10.3390/s120911910.

13. Li, Q., M. Young, V. Naing, and J. M. Donelan. Walking speed and slope estimation using shank-mounted inertial measurement units. In: 2009 IEEE International Conference on Rehabilitation Robotics, 2009, pp. 839–844. https://doi.org/10.1109/ICORR.2009.5209598.
14. Sabatini, A. M., C. Martelloni, S. Scapellato, and F. Cavallo. Assessment of walking features from foot inertial sensing. *IEEE Trans. Biomed. Eng.* 52(3):486–494, 2005. https://doi.org/10.1109/TBME.2004.840727.
15. Svensson, W., and U. Holmberg. Foot and ground measurement using portable sensors. In: 9th International Conference on Rehabilitation Robotics, 2005. ICORR, 2005, pp. 448–451. https://doi.org/10.1109/ICORR.2005.1501139.
16. Shim, M., J. I. Han, H. S. Choi, S. M. Ha, J. H. Kim, and Y. S. Baek. Terrain feature estimation method for a lower limb exoskeleton using kinematic analysis and center of pressure. *Sensors*. 19(20):4418, 2019. https://doi.org/10.3390/s19204418.
17. Zou, C., R. Huang, J. Qiu, Q. Chen, and H. Cheng. Slope gradient adaptive gait planning for walking assistance lower limb exoskeletons. *IEEE Trans. Autom. Sci. Eng.* 18(2):405–413, 2021. https://doi.org/10.1109/TASE.2020.3037973.
18. Kim, J., S. J. Kim, and J. Choi. Real-time gait phase detection and estimation of gait speed and ground slope for a robotic knee orthosis. In: 2015 IEEE International Conference on Rehabilitation Robotics (ICORR), 2015, pp. 392–397. https://doi.org/10.1109/ICORR.2015.7281231.
19. Medrano, R. L., G. C. Thomas, C. G. Keais, E. J. Rouse, and R. D. Gregg. Real-time gait phase and task estimation for controlling a powered ankle exoskeleton on extremely uneven terrain. 2022. https://doi.org/10.48550/arXiv.2205.00155.
20. Kang, I., P. Kunapuli, H. Hsu, and A. J. Young. Electromyography (EMG) signal contributions in speed and slope estimation using robotic exoskeletons. In: 2019 IEEE 16th International Conference on Rehabilitation Robotics (ICORR), 2019, pp. 548–553. https://doi.org/10.1109/ICORR.2019.8779433.
21. Lee, D., 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.* 6(2):3995–4000, 2021. https://doi.org/10.1109/LRA.2021.3066973.
22. Camargo, J., W. Flanagan, N. Csomay-Shanklin, B. Kanwar, and A. Young. A machine learning strategy for locomotion classification and parameter estimation using fusion of wearable sensors. *IEEE Trans. Biomed. Eng.* 68(5):1569–1578, 2021. https://doi.org/10.1109/TBME.2021.3065809.
23. Kalman, R. E. A new approach to linear filtering and prediction problems. *J. Basic Eng.* 82(1):35–45, 1960. https://doi.org/10.1115/1.3662552.
24. Bhakta, K., J. Camargo, L. Donovan, K. Herrin, and A. Young. Machine learning model comparisons of user independent & dependent intent recognition systems for powered prostheses. *IEEE Robot. Autom. Lett.* 5(4):5393–5400, 2020. https://doi.org/10.1109/LRA.2020.3007480.
25. Aminian, K., P. Robert, E. Jequier, and Y. Schutz. Estimation of speed and incline of walking using neural network. *IEEE Trans. Instrum. Meas.* 44(3):743–746, 1995. https://doi.org/10.1109/19.387322.
26. Young, A. J., and L. J. Hargrove. A classification method for user-independent intent recognition for transfemoral amputees using powered lower limb prostheses. *IEEE Trans. Neural Syst. Rehabil. Eng.* 24(2):217–225, 2016. https://doi.org/10.1109/TNSRE.2015.2412461.
27. Young, A. J., A. M. Simon, N. P. Fey, and L. J. Hargrove. Intent recognition in apowered lower limb prosthesis using time history information. *Ann. Biomed. Eng.* 42(3):631–641, 2013. https://doi.org/10.1007/s10439-013-0909-0.
28. Fukuchi, C. A., R. K. Fukuchi, and M. Duarte. Effects of walking speed on gait biomechanics in healthy participants: a systematic review and meta-analysis. *Syst. Rev.* 8(1):153, 2019. https://doi.org/10.1186/s13643-019-1063-z.
29. Huang, H., T. A. Kuiken, and R. D. Lipschutz. A strategy for identifying locomotion modes using surface electromyography. *IEEE Trans. Biomed. Eng.* 56(1):65–73, 2009. https://doi.org/10.1109/TBME.2008.2003293.
30. Farrell, M. T., and H. Herr. A method to determine the optimal features for control of a powered lower-limb prostheses. In: 2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 2011, pp. 6041–6046. https://doi.org/10.1109/IEMBS.2011.6091493.
31. Varol, H. A., F. Sup, and M. Goldfarb. Multiclass real-time intent recognition of a powered lower limb prosthesis. *IEEE Trans. Biomed. Eng.* 57(3):542–551, 2010. https://doi.org/10.1109/TBME.2009.2034734.
32. Huang, H., F. Zhang, L. J. Hargrove, Z. Dou, D. R. Rogers, and K. B. Englehart. Continuous locomotion-mode identification for prosthetic legs based on neuromuscular—mechanical fusion. *IEEE Trans. Biomed. Eng.* 58(10):2867–2875, 2011. https://doi.org/10.1109/TBME.2011.2161671.
33. Wang, C., X. Wu, Y. Ma, G. Wu, and Y. Luo. A flexible lower extremity exoskeleton robot with deep locomotion mode identification. *Complexity*.2018:e5712108, 2018. https://doi.org/10.1155/2018/5712108.
34. Bhakta, K., J. Camargo, W. Compton, K. Herrin, and A. Young. Evaluation of continuous walking speed determination algorithms and embedded sensors for a powered knee & ankle prosthesis. *IEEE Robot. Autom. Lett.* 6(3):4820–4826, 2021. https://doi.org/10.1109/LRA.2021.3068711.
35. Kleene, S. C. Representation of Events in Nerve Nets and Finite Automata. RAND Corporation, 1951. https://www.rand.org/pubs/research_memoranda/RM704.html. Accessed 15 July 2023.
36. Chen, T., and C. Guestrin. XGBoost: a scalable tree boosting system. *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Min*. KDD '16. Association for Computing Machinery, 2016, pp. 785–794.
37. Shawen, N., L. Lonini, C. K. Mummidisetty, et al. Addendum of: fall detection in individuals with lower limb amputations using mobile phones: machine learning enhances robustness for real-world applications. *JMIR mHealth uHealth*.5(12):e167, 2017. https://doi.org/10.2196/mhealth.9177.
38. Nakagome, S., T. P. Luu, Y. He, A. S. Ravindran, and J. L. Contreras-Vidal. An empirical comparison of neural networks and machine learning algorithms for EEG gait decoding. *Sci. Rep.* 10(1):4372, 2020. https://doi.org/10.1038/s41598-020-60932-4.
39. Qin, P., and X. Shi. Evaluation of feature extraction and classification for lower limb motion based on sEMG signal. *Entropy (Basel)*. 22(8):852, 2020. https://doi.org/10.3390/e22080852.
40. Koehler, S. R., Y. Y. Dhaher, and A. H. Hansen. Cross-validation of a portable, six-degree-of-freedom load cell for use in lower-limb prosthetics research. *J. Biomech.* 47(6):1542–1547, 2014. https://doi.org/10.1016/j.jbiomech.2014.01.048.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.