1. Title Page

Artificial Neural Network-Based Activities Classification, Gait Phase Estimation, and Prediction

Shuangyue Yu[1,2], Jianfu Yang[1], Tzu-Hao Huang[1], Junxi Zhu[1], Christopher J. Visco[3], Farah Hameed[3], Joel Stein[3], Xianlian Zhou[4], Hao Su[1†]

[1]Lab of Biomechatronics and Intelligent Robotics (BIRO), Department of Mechanical and Aerospace Engineering, North Carolina State University, Raleigh, NC, 27695, USA

[2]Department of Mechanical Engineering, City University of New York, New York, NY, 10031, USA

[3]Department of Rehabilitation and Regenerative Medicine, Columbia University Vagelos College of Physicians and Surgeons, New York, NY, 10032, USA

[4]Department of Biomedical Engineering, New Jersey Institute of Technology, Newark, NJ, 07102, USA

[†] Corresponding author. E-mail: hao.su796@ncsu.edu

2. Abstract and key terms

Gait patterns are critical to health monitoring, gait impairment assessment, and wearable device control. Unrhythmic gait pattern detection under community-based conditions is a new frontier in this area. The present paper describes a high-accuracy gait phase estimation and prediction algorithm built on a two-stage artificial neural network. This work targets to develop an algorithm that can estimate and predict the gait cycle in real time using a portable controller with only two IMU sensors (one on each thigh) in the community setting. Our algorithm can detect the gait phase in unrhythmic conditions during walking, stair ascending, and stair descending, and classify these activities with standing. Moreover, our algorithm is able to predict both future intra- and inter-stride gait phases, offering a potential means to improve wearable device controller performance. The proposed data-driven algorithm is based on a dataset consisting of 5 able-bodied subjects and validated on 3 different able-bodied subjects. Under unrhythmic activity situations, validation shows that the algorithm can accurately identify multiple activities with 99.55% accuracy, and estimate ($rRMSE_0$: 6.3%) and predict 200-ms-ahead ($rRMSE_{200ms}$: 8.6%) the gait phase percentage in real time, which are on average 57.7% and 54.0% smaller than the error from the event-based method in the same conditions. This study showcases a solution to estimate and predict gait status for multiple unrhythmic activities, which may be deployed to controllers for wearable robots or health monitoring devices.

Key terms — Activities classification, gait phase detection, artificial neural networks, exoskeleton

3. Introduction

An individual's gait pattern is a trove of useful information. Observations of the gait cycle can be decoded and broken down into various measures, such as joint loading or range of motion. These measures are repeated in patterns that can identify abnormalities, adaptations, or intent when compared to population averages and previous gait cycles. Gait pattern information plays an essential role in the fields of medicine, physical rehabilitation, and sports physiology. Specifically, gait analysis can be used to monitor or improve overall health [1] and detect the progression of certain disorders (e.g., cerebral palsy [2], multiple sclerosis [3], and Parkinson's disease [4]). Wearable assistive devices like exoskeletons [5-7] and prostheses [8] have benefited from the gait analysis [6, 7].

When monitoring gait in real time, the first challenge is to accurately estimate phase information within multi non-rhythmic activities in a timely manner. Notably, inaccurate or delayed locomotion estimates could diminish any benefit derived from the estimation and may even cause injury. State-of-the-art gait phase detection algorithms have demonstrated a capacity for high performance in fixed-speed walking (rhythmic activity) in the laboratory [7] and clinical [2-4] settings. Although several methods [9, 10] have been developed to improve these locomotion estimates, accuracy and reliability remain elusive in real-world scenarios due to the variability within a locomotion task, lack of access to laboratory instrumentation, and uncontrollable environmental factors environments.

The traditional gait estimation methods employed for rhythmic walking can be divided into two main groups: finite state machine (FSM)-based methods [11, 12] and oscillator-based methods [13, 14]. The finite-state machine-based method detects gait events discretely using predefined transition rules, such as heel-strike, toe-off, stance phase, swing phase, loading-response, or pre-swing. One drawback of this method is that a change in gait frequency cannot be detected until the following gait event [11, 12]. In contrast with the FSM-based method, the oscillator-based method can react to joint angle changes without waiting for gait events to happen. Moreover, it can continuously detect the gait phase between 0 to 100% by adapting to the frequency. However, such adaptation requires time in order for the phase estimation to converge on the new cadence, such that the change is reflected gradually [13, 14].

Machine learning methods, including the hidden Markov model (HMM) [15] and Artificial Neural Network (ANN) approaches [16-19], have potential applications in non-rhythmic gait estimation through the prediction of future states. Seo et al. [16] presented a recurrent neural network (RNN) that can estimate the gait phase during non-steady speed walking. Wang et al. [17] proposed a real-time gait phase estimation method based on deep learning, which could robustly handle variations in gait speed but was unable to handle gait estimation around gait initiation and termination. Kang et al. [18] proposed an artificial neural network-based gait phase detection method to achieve hip exoskeleton control. Their method could detect the walking speed in real time and accordingly generate an estimated gait phase percentage. Our previous research used ANN to estimate the gait phase in nonrhythmic multiple individual activities including walking [19]. *It should be stressed that these methods exclusively detect the gait phase during single activity (i.e., walking) and cannot deal with transitions between multi-activities, such as between standing, walking, stair ascending, and stair descending.*

Despite the significant contributions to nonrhythmic walking pattern estimation made by contemporary research [15-19], there is currently no approach that can detect real-time gait phase percentage while also identifying the specific locomotor activities encountered in daily life. Accordingly, the Bayesian network for estimation of the locomotion mode [20], and the threshold-based locomotion mode identification [21] were developed to switch between multiple activities such as stair-ascending, stair-descending, walking, and standing, the two-tier gait classification [20]. However, these *papers exclusively focused on activity classification, excluding the estimation of the gait phase in different activities.*

The second challenge limiting the assistive control performance of wearable robots is the delay between human movement and device torque assistance. Such a delay comes with the time needed to communicate the sensory input and the computation time of the algorithm, motor control, and human-robot interaction, which cumulatively result in incorrect assistance and potentially give rise to safety issues, e.g., falling. Fortunately, gait prediction can help the controller generate a predicted torque profile to compensate for the assistance delay. Although previous studies focused on predicting kinematics such as the joint angle [21], this approach could only partially characterize some stride traits and instead was mainly used in position control. Other studies utilized electromyographic activity and its associated electromechanical delay to predict movement before it even occurred [22]. However, the gait prediction performance here is not robust due to the unavoidable noise generated by muscle activity detection. A method to accurately predict the gait phase percentage and remove the controller lag could be beneficial with lower-limb assistive devices.

To address the challenges associated with accurately estimating the continuous gait phase for multi-activity non-rhythmic conditions and predict delay for control, we propose a two-stage artificial neural network-based gait phase detection method using two thigh-mounted IMU sensors. An event-based method incorporating footswitches is used as a baseline for comparison. The contribution of the present paper is to propose an algorithm that can 1) accurately (99.55%) classify four motions (standing, walking, stair ascending, and stair descending); 2) accurately estimate ($rRMSE_0$: 6.3%) and 200-ms-ahead predict ($rRMSE_{200ms}$: 8.6%) the gait phase percentage in real time under unrhythmic situations, which are on average 57.7% and 54.0% smaller than the error made by the event-based method in the same conditions.

The proposed algorithm can identify the gait cycle in non-rhythmic gait conditions and switch between different activities based on the classification of multi-activities. Since the control timing exerts a crucial influence over the performance of wearable robot assistance, our algorithm also predicts the future gait phase between strides for unrhythmic gaits, which has the potential to improve wearable device controller performance through delay compensation.

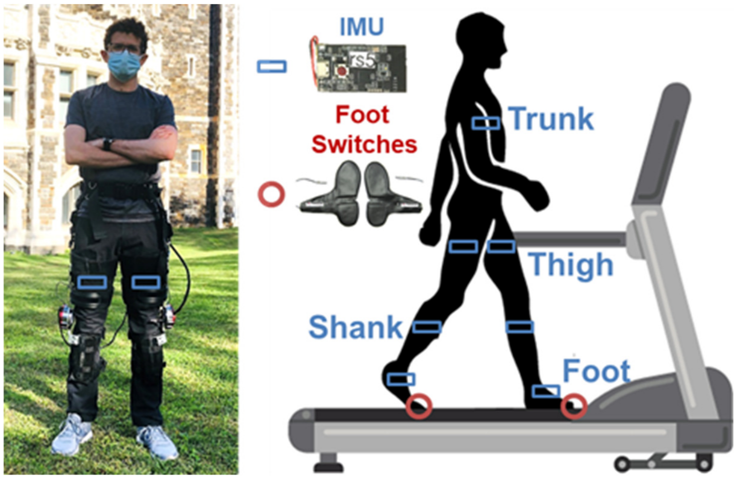


Figure 1: Sensing configuration for activity classification, gait phase estimation, and prediction, which can be used for both gait monitoring and robot control. As in the

training procedure, the footswitches were used to provide the ground truth of gait phase labeling.

The proposed method can be fully applied to nearly all kinds of lower-limb assistance devices. Alternatively, it can be isolated as an undergarment portable sensing suit for gait monitoring of able-bodied and disabled people alike (see Fig. 1). The present study offers a solution to estimate and predict gait status for multi non-rhythmic activities, which could be deployed in controllers of wearable robots, as well as in health monitoring devices.

4. Materials and Methods

Activities classification has been widely analyzed within the field of rehabilitative robotics [23, 24]. Linear discriminate analysis (LDA) and support vector machines (SVM) are commonly used methods [25-27] but provided limited performance to process the activities transition in a short latency well maintain high accuracy. Other models like the k-nearest neighbors' algorithm (KNN) or Gaussian mixture models have also been studied, but without significant improvement in comparison to LDA [23, 28]. Our previous work [19] showed that, compared with logistic regression (86.84% mean accuracy), both random forest (99.24% mean accuracy) and neural network (97.65% mean accuracy) classifiers have the potential to accurately estimate the gait cycle for multiple activities. Although the accuracy of the random forest classifier is a little bit higher, we still chose the neural network classifier due to the lower disk space requirement [29] which is imperative for deployment on microcontrollers (such as Arduino). We used a two-stage neural network structure to implement the activity classification and gait phase estimation algorithm. The first stage employed a classifier to recognize the activity. The second stage selected a regressor to estimate the gait phase percentage according to the activity.

4.1 ANN-Based Activities Classification, Gait Estimation, and Prediction

In this study, we designed a network utilizing the raw acceleration and angular velocity data. Our previously designed portable knee exoskeleton was used to support this study (see Fig. 1). The activity data were measured by the embedded wireless IMUs (as input), whilst the ground truth gait phase data were measured by the footswitches. The outputs are 1) the current activity label; 2) the estimated and predicted gait phase percentage in real time. The algorithm architecture is detailed in Fig. 2.

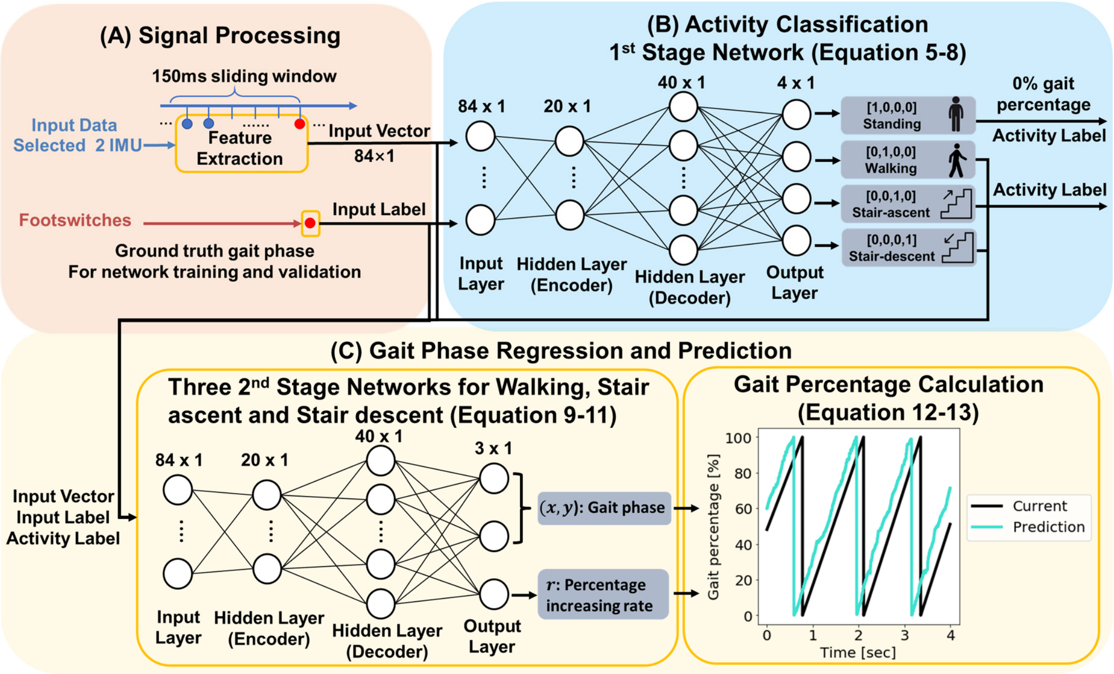


Figure 2: The architecture of the proposed two-stage artificial neural network for unrhythmic activities classification, gait phase estimation, and prediction. (A) Signal processing: The raw input data includes angular velocities and accelerations, and the features were extracted through a 150 ms sliding window. The synchronized footswitches provided the ground truth gait phases to be used in network training and validation. (B) Activity classification: The 1st-stage network. The input vector, hidden layers (encoder and decoder), and output layer had sizes of 84×1, 20×1, 40×1, and 4×1, respectively. The activity output [1 0 0 0], [0 1 0 0], [0 0 1 0] and [0 0 0 1] represented standing, walking, stair ascending, and stair descending, respectively. The output gait phase percentage was set to 0 when standing (activity output =0) was detected (C) Gait phase regression and prediction: Three 2nd-stage networks were created for walking, stair ascending, and stair descending, respectively. The input vector, hidden layers (encoder and decoder), and output layer had sizes of 84×1, 20×1, 40×1, and 4×1, respectively. The activity output (x, y) represents the gait phase and can be transferred to percentage, whilst r represents the percentage increasing rate for gait prediction measured in units of percentage per second.

The algorithm structure contained four networks in total: one activity detection classification network and three separate regression networks for gait phase estimation and prediction specific to the walking activity (one network each for walking, stair-ascending, and stair-descending). With regard to the distinct patterns of motion, a separate regression network consisting of a two-stage neural network for each walking activity is expected to outperform a single regression network for all activities, although this would likely cause training complications for the regression network. A second benefit of using a two-stage neural network structure is that an assistive device can be used in the output of the first stage to select the most appropriate assistance profile for the identified activity in the following control block. Each of the networks has one encoder layer with 20 neurons and one decoder layer with 40 neurons.

In addition to estimating the current gait phase percentage, we designed an algorithm to predict the future gait phase percentage by learning the gait phase rate-of-change. Specifically, the gait phase percentage at a specific time point in the future was extrapolated from the current gait phase percentage using the gait phase rate-of-change. Gait phase prediction has the potential to improve lower-limb robot control performance by compensating for the control phase lag.

4.1.1 Sensor Configuration and Motion Dataset Collection

A total of five subjects with an average age of 27.0 ± 2.1 years, a height of 1.70 ± 0.08 m, and a weight of 69.2 ± 16.3 kg participated in the training process. All subjects signed an informed consent form approved by the Institutional Review Board (IRB) of the City College of New York. To detect and extract the motion signals, seven IMUs (HI219M, HiPNUC Technology) were mounted on the subjects' major body segments. On each leg, we placed one IMU at the front of the thigh segment (2 cm above the knee joint), one at the back of the shank segment (2 cm above the ankle joint), and one in the middle of the instep of the foot, with a total of three IMUs on each leg. The six leg-mounted IMUs were responsible for recording the kinematics of each segment. An IMU was positioned at the center back of the trunk segment to measure the whole-body motion of the subject. The IMUs had a sampling frequency of 200 Hz and measured 3-axis acceleration and angular velocity. The IMUs were programmed with built-in software functions to calculate the Euler angles for each segment. To measure the ground truth gait phase data, we calculated the time interval between two ground contacts from the footswitch (B&L Engineering, CA, USA) in each shoe. The IMUs and footswitches were incorporated into a free-moving exoskeleton designed by the authors (see Fig. 1).

The subjects were fitted with all of the sensors and completed a range of walking tasks at specified speeds to compile the dataset for training the two-stage neural network. The locomotion tasks were: 1) standing still for 4 minutes; 2) walking on a treadmill at speeds of 0.6, 0.9, 1.2, and 1.5 m/s for 1 minute each; 3) ascending a total of 108 steps (18 steps, 6 times) at a speed of 80, 100, and 120 steps/minute; and 4) descending stairs at the same speed conditions as with ascending. The protocol for motion data collection is set out in Fig. 3.

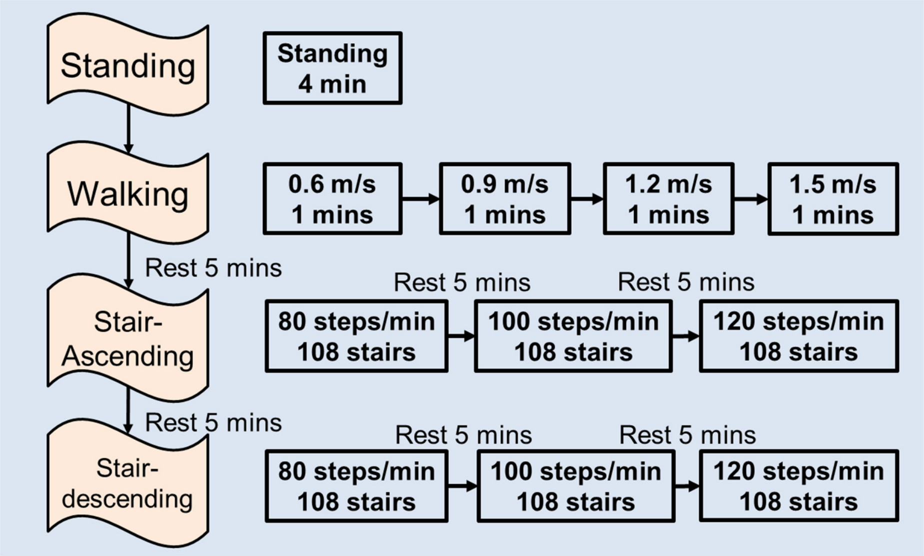


Figure. 3. The protocol for motion dataset collection from 5 subjects for offline training, including standing, walking (0.6, 0.9, 1.2, and 1.5 m/s), stair ascending, and descending (80, 100, and 120 steps per minute).

4.1.2 Sliding Window Size and Sensor Selection

As using raw real-time signals alone produces noisy results, the signal history is essential to achieve a smoother performance. Hence, we implemented a sliding window method [18] to extract the features. To identify the optimal time window size, we applied a leave-one-subject-out cross-validation method, which utilized a baseline network consisting of a hyperbolic tangent activation function and one hidden layer of 100 neurons.

The validation losses of each of the four networks (one activity classification and three separate phase regressions) and the mean of all four networks (purple bars) are illustrated in Fig. 4 (A). The validation losses were calculated through the cross-entropy loss function (Equation 1) in order to represent the accuracy under different conditions. The error bars of the validation loss represent the 95% confidence interval. The mean results show that using a 150 ms sliding window leads to the lowest validation loss (0.031±0.008), compared with using 50 ms (0.039±0.011), 100 ms

(0.035±0.007), 200 ms (0.032±0.009), 250 ms (0.033±0.009), and 300 ms (0.033±0.008) sliding windows. Specifically, the losses of gait phase regression decreased as the window size increased, stabilizing after 150ms. Besides, the loss of activity detection achieved the minimum at around 150ms. On the other hand, a smaller sliding window allowed the model to rapidly respond to activity changes. These findings suggested that 150ms was the optimal choice. Therefore, the time window size for the two-stage network in the following study was set as 150ms.

To minimize the computational cost of our neural network, we reduced the dimension feature space used as the algorithm input by optimizing the number of sensors. We identified the most relevant sensors using the cross-validation method mentioned earlier. As shown in Fig. 4 (B), the average validation losses (purple bars) of the four neural networks were 0.031±0.008 (all sensors (All)), 0.103±0.047 (unilateral thigh (U-TH)), 0.055±0.021 (unilateral multi-sensors (U-ML)), 0.029±0.006 (bilateral thighs (B-TH)), 0.041±0.018 (bilateral shanks (B-SH)), and 0.046±0.012 (bilateral feet (B-FT)). We opted to use the input from the bilateral thigh sensors for the remainder of the study as it exhibited low validation loss and required a low number of sensors, thus reducing the hardware and software complexity and rendering it a more adaptable solution.

As we observed that the Euler Angle data drifted with time during walking, we determined the 3-axis acceleration and 3-axis angular velocity from the sensors to be the optimal input vector for the artificial neural network. We used 7 features (maximum, minimum, mean, standard deviation, first, middle, and last) from each signal channel during the sliding time window that was combined to form the input vector. Thus, the input vector had a size of 84×1.

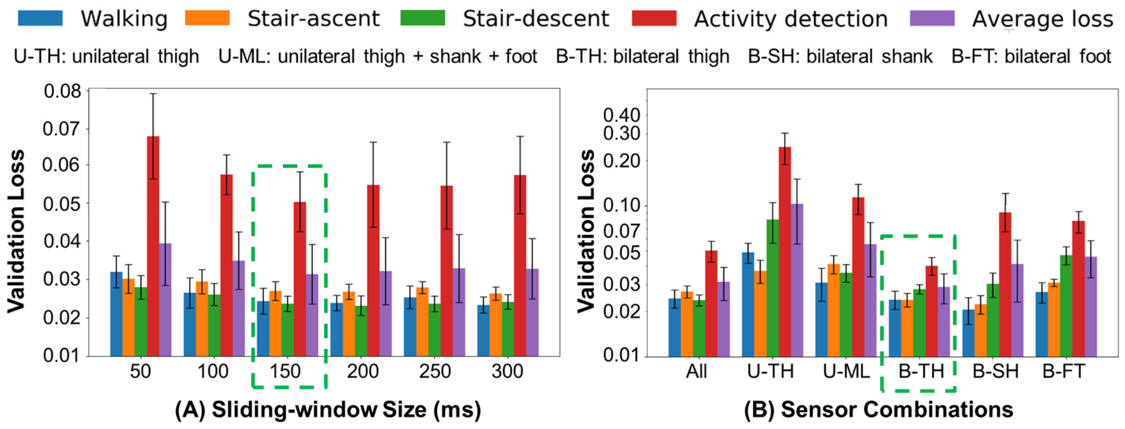


Figure 4: The validation losses during (A) sliding window size selection and (B) sensor selection. The validation losses were calculated through the cross-entropy loss function (see Equation 1) to represent the accuracy using different conditions. The error bars for each validation loss value represent the 95% confidence interval. The results show that using a 150 ms sliding window (A) and bilateral thigh IMUs data (B) produces the lowest validation losses. On this basis, the proposed method uses two bilateral IMU sensor data as inputs, with 150 ms sliding windows for feature extraction.

4.1.3 Label Interpretation and Output Layer Design

To facilitate the first stage of the neural network, we used a 4×1 vector as a classifier. The output layer of the activity detection network consisted of four neurons, followed by a soft-max layer; together the classifier result and outputs a 4×1 vector to represent the activity category. Unique identification numbers were assigned to each activity: [1 0 0 0] for standing, [0 1 0 0] for walking, [0 0 1 0] for stair-ascending, and [0 0 0 1] for stair-descending. The activity detection training process aimed to minimize any loss from the cross-entropy loss function as formulated by (1), where y_i is the binary indicator from the ground truth label and p_i is the probability calculated by the network for the i-th activity.

$$loss_{activity} = -\sum_{i=1}^{4} y_i \log(p_i) \tag{1}$$

The second stage of the neural network used labels generated from the footswitch signals to carry out phase estimation and prediction. Meanwhile, we analyzed the foot pressure signals offline to identify heel-strike events. We then assigned each timestamp falling between two consecutive heel strikes a gait phase percentage in the range of 0-100%, denoted with ϕ. However, this gait phase percentage could not be directly used as the regression label due to its discontinuity (i.e., there was an abrupt change of gait phase percentage from 100% to 0% immediately before and after a heel strike). Moreover, the motion signals were continuous at the heel strike, indicating continuous regression results. With this in mind, we decided to implement a label interpretation as formulated in (2) to maintain the continuity of regression results. The label tuple (x, y) was a bijection to the gait phase percentage and continuous, thus satisfying our requirements. We also introduced the label r to represent the rate-of-change of gait phase percentage and predict the future status as formulated in (3), where $\Delta T_{current}$ represents the current stride duration. The final label was a 3x1 vector, meaning that the output layer of the gait phase prediction network had three neurons. The gait phase prediction training process minimized loss by utilizing the loss function formulated in (4) where $[\hat{x}, \hat{y}, \hat{r}]^{\mathrm{T}}$ represents the output of the neural network.

$$x = \cos(2\pi\phi)\,, y = \sin(2\pi\phi) \tag{2}$$

$$r = 1/\Delta T_{current} \tag{3}$$

$$loss_{percentage} = (x - \hat{x})^2 + (y - \hat{y})^2 + (r - \hat{r})^2 \tag{4}$$

4.1.4 Hidden Layer Design

To reduce the computational complexity and allow the algorithm to run on Arduino portable embedded systems (>200 Hz), we designed a compression network structure consisting of two hidden layers. Notably, this structure was inspired by the autoencoder idea [24]. This compression structure is formed by inserting one relatively small layer between two large layers. The small layer is called the encoder layer, which takes the responsibility to compress the information from the predecessor layer into a small vector. Then the encoded vector is fed to the successor layer, which is also called the decoder layer and participates in the following calculation. To reduce the amount of calculations, there exists a limitation on the size of the encoder layer. The total number of calculations is $(H_p + H_s)H_e$ for the network with the encoder layer, while it is H_pH_s for the network without this structure. The sizes of the encoder layer, predecessor layer, and successor layer are denoted by H_e, H_p and H_s respectively. To reduce the calculation total by at least α, where 0 < α < 1, the inequality $(H_p + H_s)H_e \leq (1 - \alpha)H_pH_s$ is held, which infers the limitation on H_e as specified in (5). The larger α is, the more calculation is reduced. This upper bound is called the α-limitation.

$$H_e \leq (1 - \alpha)H_pH_s/(H_p + H_s) \tag{5}$$

To verify the functionality of this encoder-decoder structure, we applied a cross-validation procedure, during which two networks were evaluated. Without loss of generality, only the walking gait phase estimations are applied for simplicity. The first network is the baseline network with a hidden layer of 100 neurons. The second one deploys the proposed compression structure and inserts an encoder layer of 30 neurons between the input layer and the hidden layer. This network is 25% smaller than the baseline network. The validation results showed that there was no obvious difference between the performance of the activity classification and gait phase percentage networks (the losses are 0.015 ± 0.005 and 0.014 ± 0.003 respectively).

This implies that the encoder-decoder structure successfully maintains network performance while reducing computation requirements. Thus, this structure was implemented for adjacent large layers following equation (5) during the following training process.

4.1.5 Neural Network Training Process

The training process was completed in the Python environment using a TensorFlow backend. Each network was trained for at most 200 epochs. Since the networks were relatively small, instead of implementing a drop-out method to prevent overfitting, we deployed an early stop criterion that detected the failure across 5 continuous epochs to decrease the loss.

Before beginning to train the neural networks, a cross-validation process is required to tune the hyper-parameters, which include the number and size of hidden layers, the size of encoder layers, the choice of activation function, and the learning rate, and the batch size. The number of hidden layers was selected from 1 to 3, and the size was swept between the range from 10 to 100 with a step of 10. The size of the encoder layer was swept between the range from 10 to its 0.25-limitation with a step of 10. If the 0.25-limitation was less than 10, then the encoder layer was omitted. The activation function was selected among the rectified linear unit, sigmoid, and hyperbolic tangent. The learning rate was selected among 1e-5, 1e-4, and 1e-3. The batch size was selected among 32, 64, 128, and 256. The final selected hyper-parameters for the proposed two-stage network are listed in Table 1.

Table 1. Hyper-parameters for the proposed two-stage network

Parameter	**Value**
Learning rate	1e-4
Batch size	128
Max iterations	40000
Number of hidden layers	2
Number of neurons in hidden layer 1	20
Number of neurons in hidden layer 2	40
Activation function	Hyperbolic tangent

The two hidden layers feature 20 and 40 neurons, respectively. The first hidden layer (20 neurons) acts as an encoder, whilst the second hidden layer (40 neurons) acts as a decoder to simplify the neural network and reduce the computational costs without impacting performance [30], as shown in Fig. 2 (B)(C). During the final training procedure, the validation set is randomly selected out of the 5 subjects. After training the networks with the optimized hyper-parameters and the early stop criterion, the final validation losses were 0.030, 0.012, 0.014, and 0.021 for the activity detection network, walking, stair-ascending, and stair-descending gait phase prediction networks respectively.

4.2 Online Real-Time Evaluation Experiments

Following the offline neural network training, an experiment was designed to complete an online real-time evaluation. The classification result was evaluated by timestamp-level accuracy and activity transition detection latency. Meanwhile, the gait phase estimation and prediction results were presented as relative root-mean-square error and benchmarked (similar to [18]) using a typical event-based method [31].

4.2.1 Experiment Protocol

Three subjects (different from those used in offline training) participated in this experiment, with an average age of 25.3±0.94 years, a height of 1.74±0.09 m, and a weight of 79.3±9.10 kg. Based on the proposed sensor selection procedure, only the selected thigh IMUs were worn during the experiment. Again, the participants wore the footswitches to provide a ground truth of the gait phase percentage for analysis. A trained observer flipped a switch each time the activity changed to provide the ground truth for activity labels.

Participants were required to complete different tasks so we could evaluate the performance of the activity detection network and the gait phase prediction network at steady and non-steady state speeds. The protocol of the algorithm evaluation experiments is shown in Fig. 5.

For the experiment, tests were performed in three categorized tasks: (1) the activity-changing test, (2) the steady-speed test, and (3) the unrhythmic (speed-varying) test. Task (1) required the patients to complete certain activity sequences ([standing, walking, standing], [standing, stair-ascending, standing], [standing, stair-descending, standing]) at a self-selected speed for 10 seconds for each activity, repeating each sequence five times. Tasks (2) and (3) required the subject to perform walking, stair-ascent, and stair-decent at varying speed levels. We recorded the detected activity, the estimated and predicted gait phase percentage, and the ground truth (i.e., the observation and the footswitches signals) for offline analysis.

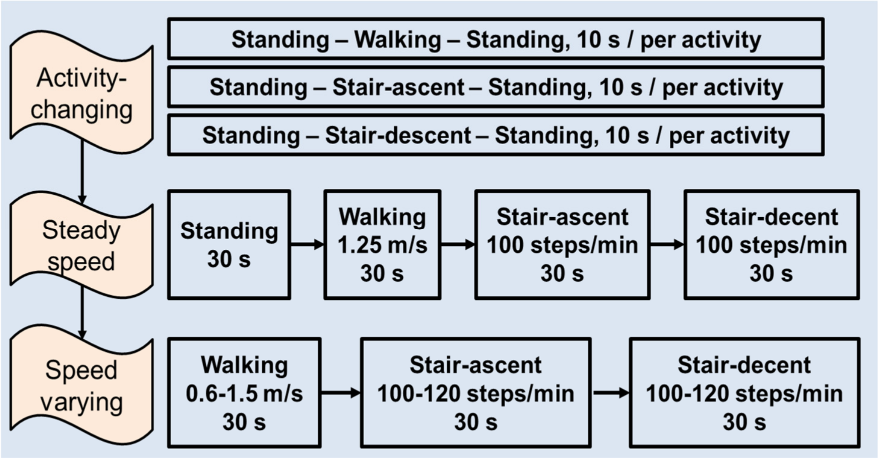

Activity-changing
Standing – Walking – Standing, 10 s / per activity
Standing – Stair-ascent – Standing, 10 s / per activity
Standing – Stair-descent – Standing, 10 s / per activity
Steady speed
Standing 30 s
Walking 1.25 m/s 30 s
Stair-ascent 100 steps/min 30 s
Stair-decent 100 steps/min 30 s
Speed varying
Walking 0.6-1.5 m/s 30 s
Stair-ascent 100-120 steps/min 30 s
Stair-decent 100-120 steps/min 30 s

Figure. 5. The protocol of the algorithm evaluation experiments on three subjects (different subjects to the offline training), including the activity changing, steady speed, and unrhythmic tests for walking, stair ascending, and stair descending to allow for gait estimation and prediction.

4.2.2 Model Implementation

This section describes the implementation of our algorithm in the microprocessor in detail. As shown in Fig. 2 (A), we input the raw signals from IMUs (acceleration and angular velocity with a sampling frequency of 200 Hz) into the sliding window function, which calculated input vector $\boldsymbol{X}$ with a size of 84x1.

The 1st stage neural network used input vector $\boldsymbol{X}$ in Equations (6)-(9) to calculate the output vector and detect activity. The structure of the network for activity classification is shown in Fig. 2 (B).

$$\boldsymbol{E}_A = \tanh(\mathbf{W}_{e,A}\boldsymbol{X} + \boldsymbol{b}_{e,A}) \tag{6}$$

$$\boldsymbol{D}_A = \tanh(\mathbf{W}_{d,A}\boldsymbol{E}_A + \boldsymbol{b}_{d,A}) \tag{7}$$

$$\boldsymbol{P}_A = \mathrm{softmax}(\mathbf{W}_{o,A}\boldsymbol{D}_A + \boldsymbol{b}_{o,A}) \tag{8}$$

$$activitiy = \arg\max(\boldsymbol{P}_A) \tag{9}$$

Among these, $\boldsymbol{E}_A$ was the encoded vector from the encoder layer; $\boldsymbol{W}_{e,A}$ was the encoder layer weight matrix; $\boldsymbol{b}_{e,A}$ was the encoder layer bias; $\boldsymbol{D}_A$ was the vector from the decoder layer; $\boldsymbol{W}_{d,A}$ was the decoder layer weight matrix; $\boldsymbol{b}_{d,A}$ was the decoder layer bias; $\boldsymbol{P}_A$ was a 4x1 vector representing the probability for each activity; $\boldsymbol{W}_{o,A}$ was the output layer weight matrix; $\boldsymbol{b}_{o,A}$ was the output layer bias. $Activity$ was used as the unprocessed activity label. If the predicted activity was standing, the output gait phase percentage would be 0. Otherwise, if the predicted label was walking or stair-related activity, the algorithm would send the input vector $\boldsymbol{X}$ to the corresponding 2nd stage neural network for gait phase estimation.

Three 2nd stage neural networks were used for walking, stair-ascending, and stair-descending, respectively. The estimated and predicted gait phase percentage was calculated by Equations (10)-(13). The structure of the gait estimation and prediction networks is set out in Fig. 2 (C).

$$\boldsymbol{E}_a = \tanh(\mathbf{W}_{e,a}\boldsymbol{X} + \boldsymbol{b}_{e,a}) \tag{10}$$

$$\boldsymbol{D}_a = \tanh(\mathbf{W}_{d,a}\boldsymbol{E}_a + \boldsymbol{b}_{d,a}) \tag{11}$$

$$[x, y, r]^{\mathrm{T}} = \mathbf{W}_{o,a}\boldsymbol{D}_a + \boldsymbol{b}_{o,a} \tag{12}$$

$$\phi_{current} = \mathrm{arctan2}(x, y) \times \frac{100\%}{2\pi} \tag{13}$$

$$\phi_\lambda = \phi_{current} + r\lambda \tag{14}$$

Among them, $\boldsymbol{E}_a$ was the encoded vector from the encoder layer; $\boldsymbol{W}_{e,a}$ was the encoder layer weight matrix; $\boldsymbol{b}_{e,a}$ was the encoder layer bias; $\boldsymbol{D}_a$ was the vector from the decoder layer; $\boldsymbol{W}_{d,a}$ was the decoder layer weight matrix; $\boldsymbol{b}_{d,a}$ was the decoder layer bias; (x, y) represented the transformed gait phase percentage, while r represented the percentage increasing rate. $\boldsymbol{W}_{o,a}$ was the output layer weight matrix; $\boldsymbol{b}_{o,a}$ was the output layer bias. $\phi_{current}$ was the current gait phase percentage and ϕ_λ was the future gait phase percentage after time λ based on the current gait phase percentage and percentage increasing rate r. This future gait phase percentage helped to compensate for the lag introduced during the prediction process by calculation and signal transmission. The value of λ was specified for different systems. To demonstrate the network's prediction capability, we set λ to 200 ms in this study.

4.2.3 Evaluation Metrics

We used timestamp-level accuracy r_a (15), and the activity transition detection latency Δt_t (16) to evaluate the activity detection network. T_c denoted the number of correctly classified timestamps, T was the total number of timestamps, N_t denoted the total number of activity transitions, $t_{t,n}$ and $\hat{t}_{t,n}$ denoted the detection and ground truth of the nth activity transition timing.

$$r_a = T_c/T \tag{15}$$

$$\Delta t_t = \frac{1}{N_t}\sum_{n=1}^{N_t}\left(t_{t,n} - \hat{t}_{t,n}\right) \tag{16}$$

The evaluation metrics for gait phase prediction were the relative root mean square error (rRMSE) of the gait phase percentage ($rRMSE_{\lambda}$, $\lambda \in \{0{,}200ms\}$ for the estimated current gait phase percentage and the predicted future gait phase percentage, respectively (17). In these equations, N denotes the number of total gait cycles and N_d denotes the number of detected gait cycles. These were recognized as detected if the gait phase percentage gradually increased to 100% before dropping to 0%. $\hat{\phi}_t$ is the estimated gait phase percentage at timestamp t. $\hat{\phi}_{t+\lambda}$ is the predicted gait phase percentage made at timestamp t for $t+\lambda$ by (13), $\phi_t$represents the ground truth of the gait phase percentage at timestamp t,$\phi_{t+\lambda}$ represents the ground truth of the gait phase percentage at timestamp $t+\lambda$ and $\bar{\phi}$ represents the average of the gait phase percentage.

$$rRMSE_{\lambda} = \frac{1}{\bar{\phi}} \sqrt{\frac{1}{T-\lambda} \sum_{t=1}^{T-\lambda} (\phi_{\lambda,t} - \hat{\phi}_{t+\lambda})^2} \tag{17}$$

4.2.4 Event-based Method for Benchmark

In addition to our neural network gait phase estimation algorithm, we also implemented a commonly applied event-based method as the baseline approach to allow for comparison (much like in [18]). The gait phase percentage was reset to 0 each time the footswitches detected ground contact and then increased at a constant rate. The algorithm calculated the increasing rate using the reciprocal of the current stride's duration, which was estimated by averaging the duration of the three previous strides. Where the actual duration was shorter than the estimation, the gait phase percentage was reset to 0 before achieving 100%. By contrast, if it was longer than the estimation, the gait phase percentage was held at 100% until the next heel strike. We utilized the increasing gait phase rate to predict future gait phase percentages in our experiments (same as Equation (13)).

5. Results

Following the experimental protocol described in Section 4.2.1, three subjects participated in this experiment to evaluate activity classification, gait phase estimation, and prediction performance.

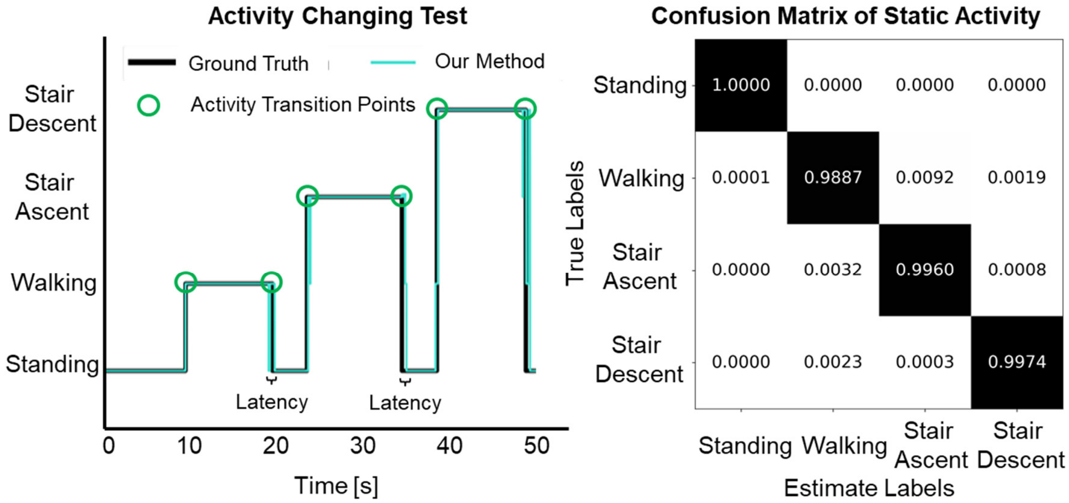
5.1 Activity Classification Evaluation

Accuracy: After applying the majority voting method, we found 1500 out of 260000 timestamps were misclassified, resulting in 99.55% accuracy. Fig. 6 (b) provides the detected activity over time as compared to the ground truth activity states in Fig. 6 (a) and the confusion matrix. As can be seen from the figure, the misclassified timestamps are mainly distributed around the activity transition point.

Latency: The latency of the neural network in identifying the activity for each type of transition is shown in Table 2. The latency is the time difference between the transition as noted by the observer and as found by the neural network. The average latency across activities was 236 ± 177 ms.

Table 2. The transition latency between activities

Δt_t[ms]		To			
		Standing	Walking	Stair-ascent	Stair-descent
From	Standing	N/A	-15.00	127.79	180.87
	Walking	224.50	N/A	-	-
	Stair-ascent	453.21	-	N/A	N/A
	Stair-descent	413.67	-	N/A	N/A



(A) (B)

Figure 6: The classification results of the activity changing test. (A) Activity detection results. The misclassified timestamps are mainly distributed around the activity transition point as a result of the transition latency. The latency is the time difference between the proposed point where the algorithm detected the transition and the ground truth. The average latency across activities was 236 ± 177 ms. (B) The confusion matrix for activity detection results shows that for standing the activity classifier is perfect (100%). Moreover, the activity classifier can accurately classify

walking (98.87%), stair ascending (99.60%), and stair descending (99.74%), resulting in an average of 99.55% classifier accuracy.

5.2 Gait Phase Estimation and Prediction Performance Compared with the Event-based Method

The results of the online estimation and prediction of gait phase percentage were compared with the footswitches ground truth in Fig. 7. Both our methods performed well under steady-speed gait phase estimation compared to the event-based method, which continually made errors during prediction and speeds varying estimation. Notably, our method fit the ground truth well. Regarding prediction, the event-based method failed between strides as it needed to wait for the heel strike. Contrastingly, our method continuously predicted the intra-stride and inter-stride gait phase percentages. Furthermore, the estimated gait phase percentage using the event-based method did not reach 100% when speed increased. Meanwhile, when speed decreased the estimated phase held at 100%.

The relative root-mean-square errors for gait estimation by both the neural network algorithm and the event-based method are summarized in Fig. 8 (a). Although the estimation error of the neural network algorithm ($rRMSE_0$ of walking, stair-ascent, and stair-descent are [4.10%, 4.58%, 5.32%], respectively) was higher than the event-based method ($rRMSE_0$: [1.66%, 4.55%, 3.51%]) under steady-speed conditions, the performance of both methods was satisfactory. In unrhythmic conditions, the estimation error of the neural network method ($rRMSE_0$: [4.18%, 6.66%, 8.03%]) was on average 57.7%smaller than the error made by the event-based method in the same conditions ($rRMSE_0$: [11.0%, 15.3%, 18.5%]).

The performance of the gait phase prediction using the two methods is summarized in Fig. 8 (b), which was higher than the estimation results in both the steady-speed and unrhythmic tests. In the steady-speed condition, the prediction error of the neural network algorithm ($rRMSE_{200ms}$ of walking, stair-ascent, and stair-descent are [4.52%, 4.88%, 5.50%], respectively) was on average 42.7% smaller than the event-based method ($rRMSE_{200ms}$: [8.56%, 8.89%, 8.43%]). Although the neural network method did not produce accurate predictions in unrhythmic conditions ($rRMSE_{200ms}$: [8.56%, 8.89%, 8.43%]), it made 54.0% fewer errors than the event-based method in the same condition ($rRMSE_{200ms}$: [14.56%, 18.76%, 22.79%]).

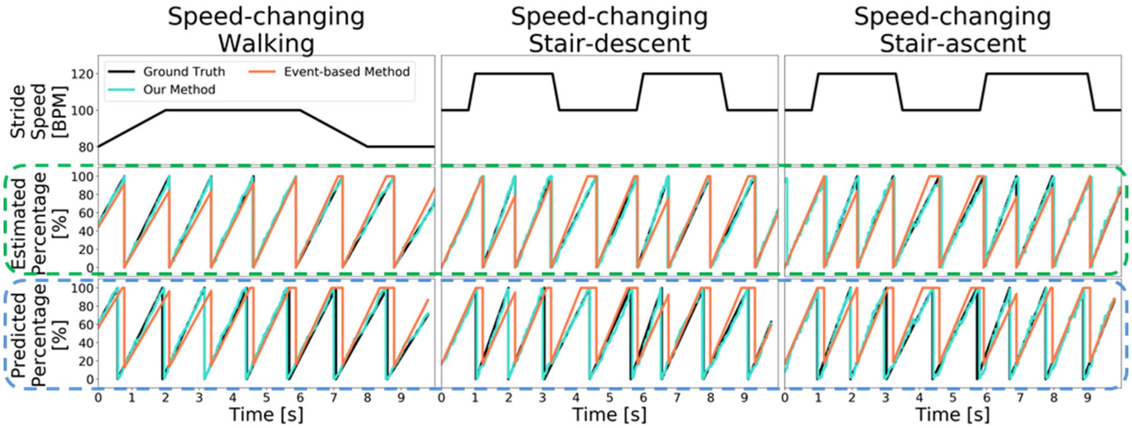
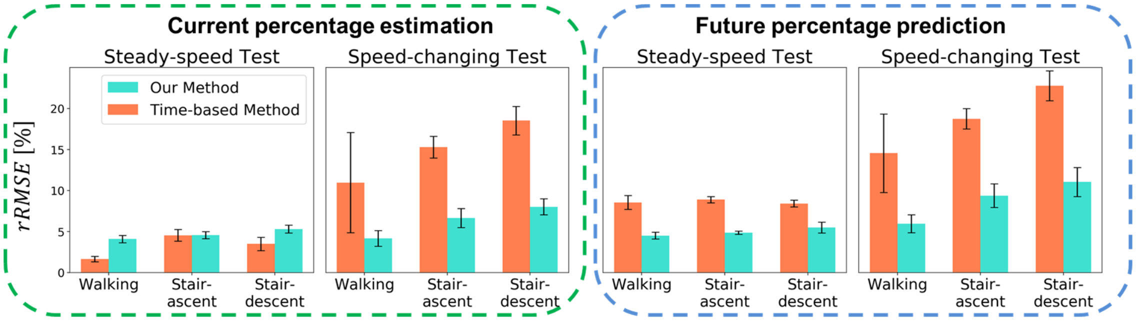


Figure 7: The performance of our method and the event-based method in the speed-changing test for walking (left), stair-ascent (center), and stair-descent (right). In terms of gait phase estimation, the event-based method performed well under steady-speed conditions for gait phase estimation. Contrastingly, our proposed method performed well under both steady-speed and speed changing conditions for both gait phase estimation (2nd row, green dashed line) and 200-ms-ahead gait phase prediction (3rd row, blue dashed line).



(a) (b)

Figure 8: Evaluation results of the (a) gait phase estimation (green dashed line) and (b) 200-ms-ahead gait phase prediction (blue dashed line) for our proposed method and event-based method under steady-speed and unrhythmic conditions. For the gait phase estimation, the event-based method performed better in the steady-speed tests, whereas our method performed better in unrhythmic situations.

6. Discussion

This paper presents a two-stage artificial neural network-based continuous gait phase detection algorithm for use in unrhythmic conditions across multi-activities using two thigh-mounted inertial measurement unit sensors. The experimental studies indicated that the proposed classification algorithm can accurately classify four motions (standing, walking, stair ascending, and stair descending) with low transition latency. The proposed gait phase regression algorithm enables highly accurate gait estimation and prediction in dynamic environments for all four activities. The algorithm was trained based on 5 individual participants' data and validated using data from another 3 participants.

The results showed that the activity classification neural network could classify the activity with an accuracy of 99.55%. For the activity classification, the misclassified timestamps were clustered around the activity transition point, suggesting transition latency caused the majority of incorrectly classified activities. The accuracy of our neural network was higher than the accuracy reported by Bartlett et al., [32] which was 98.3±3.0% for 3 activities (walking, ascending, and descending) in non-subject-specific cases. Although the activity classifier presented by Li et al. [33] exhibited greater classification accuracy (99.2%) than our activity classifier, the inclusion of a “standing” activity allowed our algorithm to handle the initiation and termination of gait, which has been a challenge for online classifiers.

In addition, our algorithm achieved a high overall accuracy of gait phase percentage estimation and prediction under multiple activities. As we used a separate neural network for gait phase prediction in each activity, our neural network exhibited satisfactory accuracy across the three locomotions. To prevent large, systematic variations in the training data, the neural network was exclusively trained in the motion patterns of interest. The small variations in the locomotion data for each activity taught the regressor to be robust to typical walking pattern deviations.

We extensively compared our method with the event-based method, the latter of which could only accurately estimate the gait phase under steady-speed conditions. Meanwhile, our proposed method could effectively handle unrhythmic gait phase estimation and prediction: it produced estimations and predicted 200 ms ahead of time the gait phase percentage in real time. Under the unrhythmic activity situations, the proposed algorithm’s gait phase estimation ($rRMSE_0$: 6.3%) and 200- ms-ahead prediction ($rRMSE_{200ms}$: 8.6%) errors were on average 57.7% and 54.0% smaller than the error made by the event-based method ($rRMSE_0$: 14.9%, $rRMSE_{200ms}$: 18.7%) in the same conditions.

Accurate online gait classification and phase estimation in a wearable system have many real-world applications. Our developed system will be useful in telemedicine to monitor the activity and gait of both able-bodied and people with disabilities in real-world environments. The algorithm can provide more detailed data on gait characteristics than a mobile phone or smartwatch carried by the patient. It also can help with wearable robot control at different speeds and in different activities. As such, different assistance profiles are typically required for different activities. Our developed method would assist in device control by accurately identifying the type of walking activity with only a short delay, thus allowing the device to quickly adopt the most appropriate assistance profile. Furthermore, many assistance profiles are functions of the gait cycle, meaning that accurate gait phase estimations or predictions would be useful in determining prosthetic or exoskeleton assistance. The prediction of the gait phase will have a greater impact on wearable device control as it will reduce the lag between sensing and action, thus allowing the assistive action to occur in synchronization with the human motion.

The limitations of the proposed methods of this work include 1) the motion dataset included five subjects and a larger dataset would be helpful to improve the universality of the algorithm 2) the data collection and testing are only used in a private environment 3) the activity classification was not tested in inclined and declined walking conditions; 4) the performance of gait phase prediction under unrhythmic conditions requires further refinement. More powerful machine learning methods run on powerful embedded systems may have the potential to have better performance, such as random forest or deep neural networks.

Future work will focus on using the proposed algorithm in specific applications for both able-bodied and people with disabilities populations. These include tracking daily activities for health monitoring and diagnosis and improving control of lower-limb robots for ambulation assistance.

7. Conflicts of Interest

No benefits in any form have been or will be received from a commercial party related directly or indirectly to the subject of this manuscript.

8. Acknowledgements

This work is supported in part by the National Science Foundation (NSF) CAREER award CMMI 1944655, NSF 2026622, NIH R01EB029765, NIDILRR 90DPGE0011. Any opinions, findings, conclusions, or recommendations expressed in this material

are those of the author (s) and do not necessarily reflect the views of the funding organizations.

9. References

[1] Y. Hutabarat, D. Owaki, and M. Hayashibe, "Quantitative Gait Assessment With Feature-Rich Diversity Using Two IMU Sensors," *IEEE Transactions on Medical Robotics and Bionics,* vol. 2, no. 4, pp. 639-648, 2020.

[2] L. Carcreff, C. N. Gerber, A. Paraschiv-Ionescu, G. De Coulon, C. J. Newman, S. Armand, and K. Aminian, "What is the best configuration of wearable sensors to measure spatiotemporal gait parameters in children with cerebral palsy?," *Sensors,* vol. 18, no. 2, p. 394, 2018.

[3] L. Filli, T. Sutter, C. S. Easthope, T. Killeen, C. Meyer, K. Reuter, L. Lörincz, M. Bolliger, M. Weller, and A. Curt, "Profiling walking dysfunction in multiple sclerosis: characterisation, classification and progression over time," *Scientific reports,* vol. 8, no. 1, pp. 1-13, 2018.

[4] C. Caramia, D. Torricelli, M. Schmid, A. Muñoz-Gonzalez, J. Gonzalez-Vargas, F. Grandas, and J. L. Pons, "IMU-based classification of Parkinson's disease from gait: A sensitivity analysis on sensor location and feature selection," *IEEE journal of biomedical and health informatics,* vol. 22, no. 6, pp. 1765-1774, 2018.

[5] S. Yu, T.-H. Huang, D. Wang, B. Lynn, D. Sayd, V. Silivanov, Y. S. Park, Y. Tian, and H. Su, "Design and control of a high-torque and highly backdrivable hybrid soft exoskeleton for knee injury prevention during squatting," *IEEE Robotics and Automation Letters,* vol. 4, no. 4, pp. 4579-4586, 2019.

[6] J. Kim, G. Lee, R. Heimgartner, D. A. Revi, N. Karavas, D. Nathanson, I. Galiana, A. Eckert-Erdheim, P. Murphy, and D. Perry, "Reducing the metabolic rate of walking and running with a versatile, portable exosuit," *Science,* vol. 365, no. 6454, pp. 668-672, 2019.

[7] 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,* vol. 356, no. 6344, pp. 1280-1284, 2017.

[8] T. Elery, S. Rezazadeh, C. Nesler, and R. D. Gregg, "Design and Validation of a Powered Knee–Ankle Prosthesis With High-Torque, Low-Impedance Actuators," *IEEE Transactions on Robotics,* 2020.

[9] C. L. Lewis and D. P. Ferris, "Invariant hip moment pattern while walking with a robotic hip exoskeleton," *Journal of biomechanics,* vol. 44, no. 5, pp. 789-793, 2011.

[10] 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, p. e56137, 2013.

[11] S. Culver, H. Bartlett, A. Shultz, and M. Goldfarb, "A stair ascent and descent controller for a powered ankle prosthesis," *IEEE Transactions on Neural Systems and Rehabilitation Engineering,* vol. 26, no. 5, pp. 993-1002, 2018.

[12] J. Jang, K. Kim, J. Lee, B. Lim, and Y. Shim, "Online gait task recognition algorithm for hip exoskeleton," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015, pp. 5327-5332: IEEE.

[13] T. G. Sugar, A. Bates, M. Holgate, J. Kerestes, M. Mignolet, P. New, R. K. Ramachandran, S. Redkar, and C. Wheeler, "Limit cycles to enhance human performance based on phase oscillators," *Journal of Mechanisms and Robotics,* vol. 7, no. 1, 2015.

[14] K. Seo, K. Kim, Y. J. Park, J.-K. Cho, J. Lee, B. Choi, B. Lim, Y. Lee, and Y. Shim, "Adaptive oscillator-based control for active lower-limb exoskeleton and its metabolic impact," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 6752-6758: IEEE.

[15] M. D. Sánchez Manchola, M. J. P. Bernal, M. Munera, and C. A. Cifuentes, "Gait phase detection for lower-limb exoskeletons using foot motion data from a single inertial measurement unit in hemiparetic individuals," *Sensors,* vol. 19, no. 13, p. 2988, 2019.

[16] K. Seo, Y. J. Park, J. Lee, S. Hyung, M. Lee, J. Kim, H. Choi, and Y. Shim, "RNN-based on-line continuous gait phase estimation from shank-mounted IMUs to control ankle exoskeletons," in *2019 IEEE 16th International Conference on Rehabilitation Robotics (ICORR)*, 2019, pp. 809-815: IEEE.

[17] F. Wang, L. Yan, and J. Xiao, "Recognition of the Gait Phase Based on New Deep Learning Algorithm Using Multisensor Information Fusion," *Sensors and Materials,* vol. 31, no. 10, pp. 3041-3054, 2019.

[18] I. Kang, P. Kunapuli, and A. J. Young, "Real-Time Neural Network-Based Gait Phase Estimation Using a Robotic Hip Exoskeleton," *IEEE Transactions on Medical Robotics and Bionics,* vol. 2, no. 1, pp. 28-37, 2019.

[19] J. Yang, T.-H. Huang, S. Yu, X. Yang, H. Su, A. M. Spungen, and C.-Y. Tsai, "Machine learning based adaptive gait phase estimation using inertial measurement sensors," in *2019 Design of Medical Devices Conference*, 2019: American Society of Mechanical Engineers Digital Collection.

[20] I. Kang, D. D. Molinaro, G. Choi, and A. J. Young, "Continuous locomotion mode classification using a robotic hip exoskeleton," in *2020 8th IEEE RAS/EMBS International Conference for Biomedical Robotics and Biomechatronics (BioRob)*, 2020, pp. 376-381: IEEE.

[21] F. Gao, G. Liu, F. Liang, and W.-H. Liao, "IMU-Based locomotion mode identification for transtibial prostheses, orthoses, and exoskeletons," *IEEE Transactions on Neural Systems and Rehabilitation Engineering,* vol. 28, no. 6, pp. 1334-1343, 2020.

[22] H. K. Yap, H. Y. Ng, and C.-H. Yeow, "High-force soft printable pneumatics for soft robotic applications," *Soft Robotics,* vol. 3, no. 3, pp. 144-158, 2016.

[23] J. Camargo, 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 Transactions on Biomedical Engineering,* vol. 68, no. 5, pp. 1569-1578, 2021.

[24] C. P. Swami, N. Lenhard, and J. Kang, "A novel framework for designing a multi-DoF prosthetic wrist control using machine learning," *Scientific Reports,* vol. 11, no. 1, pp. 1-13, 2021.

[25] B. Hu, E. Rouse, and L. Hargrove, "Fusion of bilateral lower-limb neuromechanical signals improves prediction of locomotor activities," *Frontiers in Robotics and AI,* vol. 5, p. 78, 2018.

[26] H. Huang, T. A. Kuiken, and R. D. Lipschutz, "A strategy for identifying locomotion modes using surface electromyography," *IEEE Transactions on Biomedical Engineering,* vol. 56, no. 1, pp. 65-73, 2008.

[27] H. Huang, 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 Transactions on Biomedical Engineering,* vol. 58, no. 10, pp. 2867-2875, 2011.

[28] F. Attal, S. Mohammed, M. Dedabrishvili, F. Chamroukhi, L. Oukhellou, and Y. Amirat, "Physical human activity recognition using wearable sensors," *Sensors,* vol. 15, no. 12, pp. 31314-31338, 2015.

[29] F. Gieseke and C. Igel, "Training big random forests with little resources," in *Proceedings of the 24th acm sigkdd international conference on knowledge discovery & data mining*, 2018, pp. 1445-1454.

[30] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *nature,* vol. 323, no. 6088, pp. 533-536, 1986.

[31] H. Choi, Y. J. Park, K. Seo, J. Lee, S.-e. Lee, and Y. Shim, "A multifunctional ankle exoskeleton for mobility enhancement of gait-impaired individuals and seniors," *IEEE Robotics and Automation Letters,* vol. 3, no. 1, pp. 411-418, 2017.

[32] H. L. Bartlett and M. Goldfarb, "A phase variable approach for IMU-based locomotion activity recognition," *IEEE transactions on biomedical engineering,* vol. 65, no. 6, pp. 1330-1338, 2017.

[33] H. Li, S. Derrode, and W. Pieczynski, "An adaptive and on-line IMU-based locomotion activity classification method using a triplet Markov model," *Neurocomputing,* vol. 362, pp. 94-105, 2019.