Real-Time Gait Phase Detection on Wearable Devices for Real-World Free-Living Gait

Jiaen Wu, Barna Becsek, Alessandro Schaer, Henrik Maurenbrecher, George Chatzipirpiridis, Olgac Ergeneman, Salvador Pané, Hamdi Torun, and Bradley J. Nelson, *Fellow, IEEE*

***Abstract*—Detecting gait phases with wearables unobtrusively and reliably in real-time is important for clinical gait rehabilitation and early diagnosis of neurological diseases. Due to hardware limitations of microcontrollers in wearable devices (e.g., memory and computation power), reliable real-time gait phase detection on the microcontrollers remains a challenge, especially for long-term real-world free-living gait. In this work, a novel algorithm based on a reduced support vector machine (RSVM) and a finite state machine (FSM) is developed to address this. The RSVM is developed by exploiting the cascaded K-means clustering to reduce the model size and computation time of a standard SVM by 88% and a factor of 36, with only minor degradation in gait phase prediction accuracy of around 4%. For each gait phase prediction from the RSVM, the FSM is designed to validate the prediction and correct misclassifications. The developed algorithm is implemented on a microcontroller of a wearable device and its real-time (on the fly) classification performance is evaluated by twenty healthy subjects walking along a predefined real-world route with uncontrolled free-living gait. It shows a promising real-time performance with an accuracy of 91.51%, a sensitivity of 91.70%, and a specificity of 95.77%. The algorithm also demonstrates its robustness with varying walking conditions.**

***Index Terms*—Real-time gait phase detection, embedded system algorithms, wearable sensors, gait rehabilitation, real-world free-living walking.**

Manuscript received 15 April 2022; revised 18 October 2022; accepted 2 December 2022. Date of publication 12 December 2022; date of current version 7 March 2023. This work was supported by European Union's Horizon Research and Innovation Program under the Marie Sklodowska-Curie under Grant 764977. *(Corresponding author: Jiaen Wu.)*

Jiaen Wu is with the Multi-Scale Robotics Lab, ETH Zurich, CH-8092 Zurich, Switzerland, and also with the Magnes AG, 8001 Zurich, Switzerland (e-mail: wujiae@ethz.ch).

Barna Becsek, Alessandro Schaer, Henrik Maurenbrecher, George Chatzipirpiridis, and Olgac Ergeneman are with the Magnes AG, 8001 Zurich, Switzerland (e-mail: barna@magnes.ch; aschaer@magnes.ch; henrikm@student.ethz.ch; chgeorge@magnes.ch; oergeneman@magnes.ch).

Salvador Pané and Bradley J. Nelson are with the Multi-Scale Robotics Lab, ETH Zurich, CH-8092 Zurich, Switzerland (e-mail: vidalp@ethz.ch; bnelson@ethz.ch).

Hamdi Torun is with the Department of Mathematics, Physics and Electrical Engineering, Northumbria University, Newcastle upon Tyne NE1 8ST, U.K. (e-mail: hamdi.torun@northumbria.ac.uk).

Digital Object Identifier 10.1109/JBHI.2022.3228329

I. Introduction

GAIT characteristics (e.g., gait speed), considered as the sixth vital sign, can be used as a digital biomarker for personalized health monitoring and assessment [1], [2]. Probing and evaluating gait characteristics such as the gait cycle is essential for clinical applications, including but not limited to estimating the risk of falls [3], measuring the efficacy of interventions and rehabilitation [2], and early diagnosis of various neurological diseases [4], [5], [6].

A gait cycle is characterized by different gait phases. Robust intra-stride gait phase detection (i.e., detecting phases within one stride) is the basis for mobile gait analysis, as most gait characteristics, including inter-stride gait features (i.e., changes between different strides), are derived from the intra-stride temporal gait phases [7]. Real-time gait phase detection can be utilized for closed-loop real-time feedback, which is essential for clinical gait rehabilitation as it enables patients to recognize their walking abnormalities through the real-time feedback and make conscious corrections immediately, and also help patients with spinal cord injury or brain trauma to restore the walking function [8], [9]. It can also be utilized for many clinical rehabilitation applications, such as controlling the timing of stimulation sequences for functional electrical stimulation (FES) [9], [10] and epidural electrical stimulation (EES) [11], [12]. As shown in [11], the real-time gait analysis by visual observation can be used to enable EES during walking, which can significantly improve the walking ability of impaired patients when compared to continuous EES that is not based on gait patterns. Another study has shown that the real-time detection of the swing phase based on a doctor's observation can be utilized to trigger FES to compensate for the foot-drop problem [10].

Though various computational technologies have been developed for offline gait phase detection as reviewed in [13], [14], only a very small number of works on online gait phase detection have been published [7], [15], [16], [17]. Among those online algorithms, rule-based approaches have been mostly reported due to their intuitiveness, and low computational complexity [7]. However, those approaches usually involve rule and threshold setting, as well as signal peak detection. The rules and thresholds are determined empirically based on preliminary data, thus they need to be recalibrated continuously for different people. The presence of a signal peak can only be confirmed after both the rising and falling signal edges have appeared, which may cause a significant delay, on the order of hundreds of milliseconds [7].

2168-2194 © 2022 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: Mepco Schlenk Engineering College. Downloaded on August 18,2023 at 04:05:16 UTC from IEEE Xplore. Restrictions apply.

Machine learning techniques for gait analysis have grown in prominence in recent years due to their high accuracy and automated nature. Mannini et al. placed an IMU on the foot instep and detected gait phases in real-time by using Hidden Markov Model [16]. However, their algorithm was only validated with offline treadmill walking data by a leave-one-subject-out method. Chen et al. used a standard support vector machine (SVM) for detecting gait phases, i.e., flat foot (FF), heel off (HO), toe off (TO), and swing, based on seven force-sensitive resistors (FSR) and an accelerometer placed on shoes [17]. The algorithm was validated on treadmill walking data and achieved an overall accuracy of 94.08%. In this work, the sensor data were transferred to a remote computer via Bluetooth and the online algorithm was executed on the computer instead of a local microcontroller on shoes. This limits the use of this device to lab environments. A continuous full day free-living walking assessment is not possible with this system. Besides, the readout of FSR sensors are dependent on the user's body weight. Therefore, the system needs calibration for different users. Vu et al. proposed a deep learning algorithm based on IMU data to detect the gait cycle percentage [18], which is defined as the percentage of the current sample in the gait cycle. The algorithm's performance was only assessed on the offline testing data from seven healthy subjects, no real-time validation experiments were conducted. This deep learning algorithm also runs on a remote computer.

Most of the state-of-the-art real-time gait phase estimation studies reported in the literatures [19] have only validated their algorithms offline with healthy subjects walking in the laboratory. Few studies have implemented their algorithms in systems to demonstrate their real-time estimation capabilities, not to mention their real-time performance on a real-world uncontrolled free-living gait. Of the few studies that have shown the real-time demonstration, most have implemented their algorithms on remote systems [17], [18], [20], such as mobile phones. With an online phase detection algorithm running on a remote device, users can only walk in a confined environment. If users are not in the sensor range, the gait phases cannot be detected. This also limits the use of real-time closed-loop functionalities such as FES. Most importantly, the aforementioned automatic online algorithms are validated using walking data collected under controlled and supervised conditions, e.g., in a laboratory or a hospital [21]. The variability across those in-lab walking strides is quite low. However, the uncontrolled and unsupervised free-living gait, i.e., walking in the real world, is more complex and irregular, which often incorporates dynamic walking speed, varying walking surface, and inclinations [22]. A recent study from Khandelwal et al. has shown that the published gait event detection algorithms that perform well for walking in a controlled indoor environment exhibit significantly degraded performance when evaluated with a less controlled outdoor walking, with a combined median F_1 score decreased from 0.98, 0.94 to 0.82, 0.53 for heel strike (HS) and TO, respectively [22]. Reliable real-time gait phase detection for uncontrolled free-living walking in the real-world still remains a big challenge, especially for memory and computation-limited embedded systems (microcontrollers on wearable devices) that require low power operation.

In this paper, a robust algorithm incorporating reduced support vector machine (RSVM) and finite-state machine (FSM) is proposed for real-time (on the fly) reliable gait phase detection on uncontrolled free-living gait in the real-world, with resource-limited microcontrollers (Fig. 1(a) and (b)). Compared to the standard support vector machine (SVM), we use a cascaded k-means clustering approach to reduce the model size and increase the computational speed that fits the embedded environment while still keeping the high classification accuracy. An FSM is designed to model the sequential property of gait phases and complement the RSVM with contextual information about a logical temporal sequence of gait phases. This algorithm is implemented on a microcontroller of a miniature wearable device that is embedded in a shoe and used to further trigger real-time haptic feedback provided by an actuator (Fig. 1(a)). The real-time (on the fly) effectiveness of the proposed algorithm is validated with long-term fully uncontrolled and unsupervised free-living gait performed in the real-world by twenty healthy subjects, who are guided to wear the sensor-embedded shoes with the proposed algorithm running online, and walk around the predefined paths on a busy street at three different walking speeds.

The proposed algorithm exploits only the gyroscope signals, avoiding the need for additional sensors. No prior calibration is required for different people with different walking conditions, such as walking speeds and walking surfaces. We demonstrate its capability to identify gait phases in real-time (on the fly) on a source-limited microcontroller for both indoor and outdoor (real-world uncontrolled and unsupervised free-living) walking.

II. Data Collection

A. Subjects of the Study

Two groups of healthy subjects were recruited in this study. The first group consisted of four healthy subjects. They were instructed to walk on a treadmill. The datasets collected from these subjects were separated into training and testing datasets for training and optimizing the proposed algorithm offline. This study was conducted in accordance with Good Clinical Practice guidelines and the Declaration of Helsinki after receiving a declaration of clearance from the local ethics committee (BASEC Nr Req-2019-00715).

The second group consisted of twenty healthy subjects (distinct from the first group subjects) ranging in age from 24 to 41 years old, with an average age, 29.75 ± 3.39 years, wearing European shoe sizes from 38 to 47. They were instructed to perform outdoor walking. The datasets collected from these subjects were used to validate the performance of the trained model in real-life conditions, running in real-time on the embedded electronics. The research protocol for this study was approved by grant (EK 2021-N-198) from the Ethics Commission of the ETH Zurich.

All subjects were able to walk normally with no known injuries or abnormalities that would affect their gait. Written

Authorized licensed use limited to: Mepco Schlenk Engineering College. Downloaded on August 18,2023 at 04:05:16 UTC from IEEE Xplore. Restrictions apply.

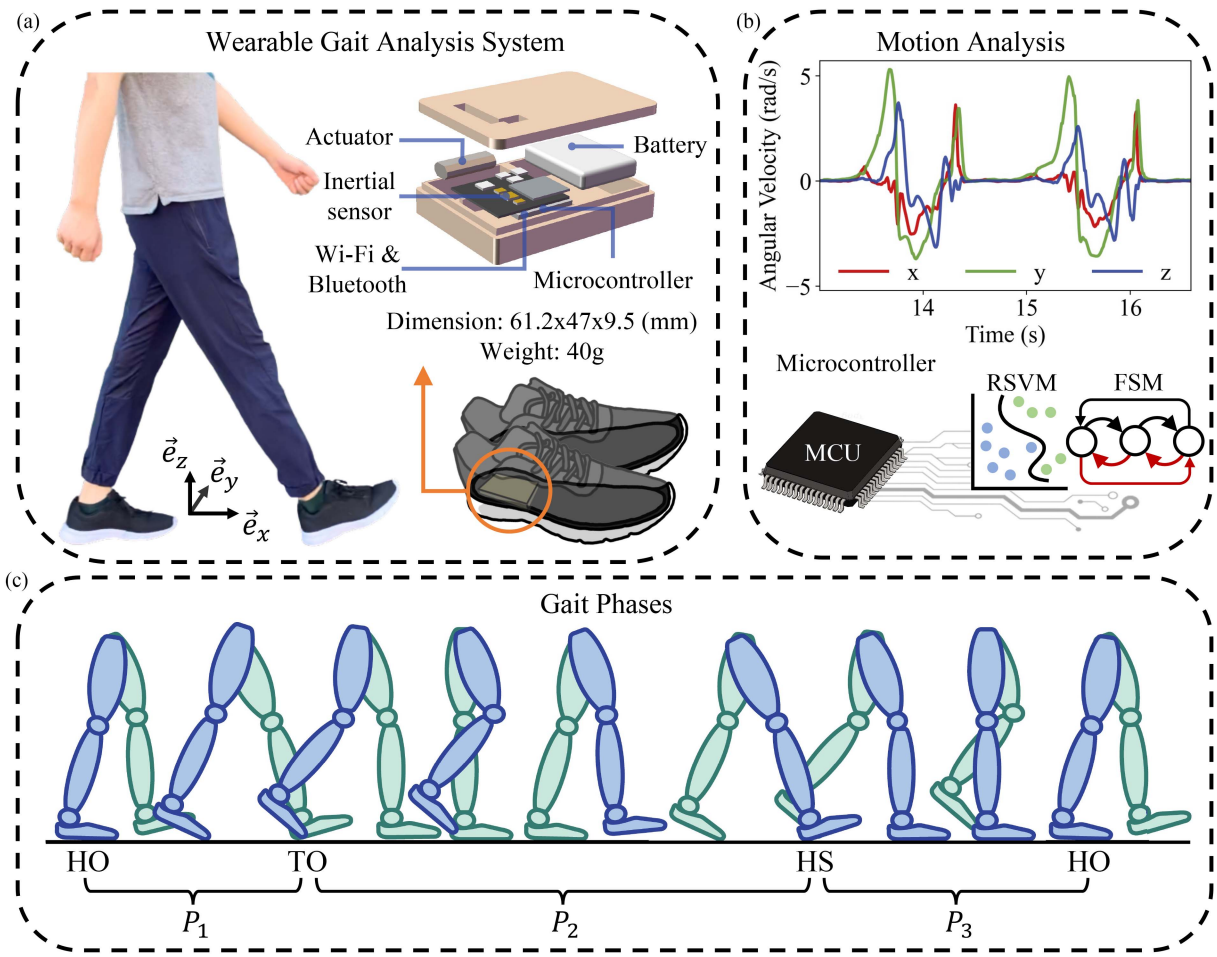

Fig. 1. A schematic illustration of the wearable gait analysis system. (a). The wearable gait analysis system consists of an actuator, an inertial sensor, a Wi-Fi and Bluetooth module, a battery, and a microcontroller. The system is embedded in a pair of shoes at the location under the heel. (b). The motion of the foot is captured by the sensing system during human walking. Only gyroscope signals are used for real-time gait phase detection. A hybrid algorithm of RSVM and FSM is developed to process raw motion signals on the local microcontrollers and output the gait phases in real-time. (c). An illustration of gait phases considered in this paper. P_1: HO–TO, P_2: TO–HS, P_3: HS–HO.

informed consents were provided by all subjects before the experiment.

B. Data Collection System

All walking data were collected by the wearable gait analysis system *Nushu* developed by Magnes AG [23]. It consists of a pair of shoes and a mobile (iOS) application. The shoes are equipped with a low-power system on custom–developed electronics including on-board inertial sensors (LSM6DSM, STMicroelectronics, Geneva, Switzerland) and a dual-core microcontroller unit (ESP32, Espressif Systems Co., Ltd, Shanghai, China) with integrated WiFi and Bluetooth modules for wireless communication. The chip has an internal memory of 520 KB SRAM and 448 KB ROM. It also integrates external memory of 16 MB SPI flash, and 4 MB SPI RAM. The CPU is operated at 240 MHz for all the experiments. The on-board inertial sensors imply both an accelerometer and a gyroscope, where only the gyroscope is used. The gyroscope data acquisition frequency is set to 100 Hz. The range of gyroscope is set to ± 2000 DPS.

C. Data Acquisition

The first group of healthy subjects was instructed to wear the sensor-embedded shoes and walk on a treadmill with a normal gait. Each subject was asked to walk three times on the treadmill with three different treadmill speeds set at 0.53 m/s, 0.86 m/s, and 1.11 m/s. Note that the average walking speed of each subject is not necessarily equal to the treadmill speed. During each trial, the subject performed around 50 strides for each side of the left and right foot.

The second group of healthy subjects was instructed to wear the sensor-equipped shoes and walk in a real-world environment outside the laboratory freely along the selected path on the street, as shown in Fig. 2. The selected walking path is on a busy street includes turns (white arrow), uphill (pink arrow), downhill (green arrow), straight lines (red arrow), curved lines/crossings (blue arrow), and crossings of the roadway (yellow arrow), with fluctuating road surface and inclinations. It has a total of approximately 500 m walkway for each bout. In the validation walking experiment, there were constantly pedestrians walking

Authorized licensed use limited to: Mepco Schlenk Engineering College. Downloaded on August 18,2023 at 04:05:16 UTC from IEEE Xplore. Restrictions apply.

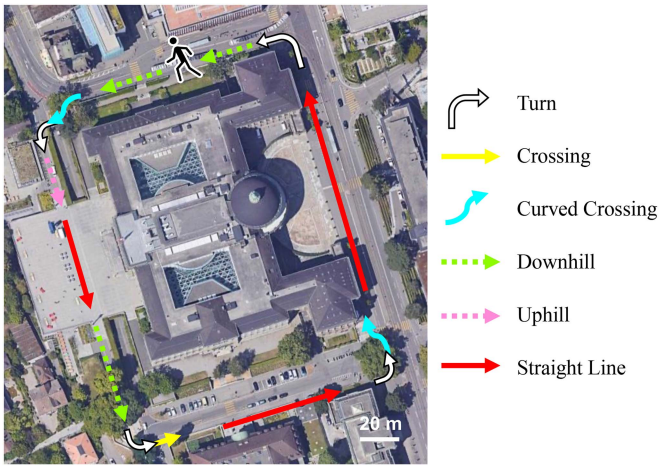

Fig. 2. Top view of the selected street path utilized for the outdoor real-world uncontrolled free-living walking experiments. The arrows with different colors show the closed-loop path defined for the experiments. The solid white arrow represents the turns, the solid yellow arrow represents the crossing of the road, the solid blue arrow represents the curved crossing of the road, the dotted green arrow represents the downhill, the dotted pink arrow represents the uphill, and the solid red arrow represents the straight line. The bouts started and stopped where the human stick figure is depicted.

around, the walking route was thus affected and cannot be kept straight. Each subject was asked to walk three times at three different speeds, i.e., slow speed, normal speed, and fast speed. Each subject was free to interpret what "slow," "normal" and "fast" meant for them. During these experiments, the real-time gait phase detection algorithm was running on the microcontroller of the shoe system along with the sensors during each subject's walking experiment. The gait phase prediction results were logged together with raw sensor data. The total number of free walking strides collected for real-time algorithm validation is around 50000.

Gait phases were labeled by an offline rule-based detection algorithm, which has been validated with the motion capture system and shows high accuracy [24]. Most importantly, unlike labeling data manually from optical motion capture systems, this labeling method is not constrained by the laboratory environment, and can label outdoor walking data offline at any time and anywhere for further real-time algorithm validation.

III. Real-Time Gait Phase Detection Algorithm

A. Gait Phases

Human gait can be divided into a series of repetitive phases and events related to its cyclic nature. Each gait cycle can be divided into two main phases i.e., stance phase and swing phase [25]. These two main phases can further be divided into sub-phases by three gait events, which are HO, TO, and HS. As shown in Fig. 1(c), three gait phases P_1, P_2, P_3 that are delimited by HO, TO, and HS for each foot are considered in this paper for real-time phase detection. They are denoted as P_1: HO-TO (terminal stance phase), P_2: TO-HS (swing phase), P_3: HS-HO (initial stance phase).

B. Feature Selection

Contrary to accelerometers, gyroscopes do not carry a constant offset due to gravity. They are more immune to vibrations and environmental noises than accelerometers [16], [26]. Moreover, it has been shown that the angular velocity patterns are not affected significantly by inter–subjects or intra–subjects variability and different walking conditions, such as walking speed, or walking slope, unless the foot-ground contact occurs at the toes rather than the heels that mostly in pathological gait [16]. Therefore, the data from a single gyroscope sensor are used for real-time gait phase detection.

By observing the signal distributions of angular velocity for different gait phases, angular velocity along the pronation axis, ω_z, angular-acceleration $\dot{\omega}_y$ and -velocity ω_y along the mediolateral axis, which is the most prominent movement, are selected as three input features. Fig. 1(a) depicts the reference frame for the coordinate axis. Due to the symmetric ankle rotation of the left and right feet, the angular velocity along the pronation axis for the right foot is flipped as $-\omega_z$. The angular acceleration, which is approximated by the discrete time derivative of the angular velocity, is computed as:

$$\dot{\omega}_y(k) \approx \frac{\omega_y(k) - \omega_y(k-1)}{t(k) - t(k-1)}, \tag{1}$$

where $\dot{\omega}_y(0) = 0$, t is the continuous time of the motion signal, $k \in \mathbb{N}$ denotes the discrete time step. This first-order derivative is included as one of the input features to provide temporal dependence information to the algorithm. Higher-order time derivatives are discarded because they can introduce an amplification of high-frequency noise and degrade the signal-to-noise ratio [27], [28]. All signals are normalized with respect to the sample means and sample standard deviations that are obtained from the training data (constants). The normalized input feature vector $\boldsymbol{x}(k)$ can be denoted as:

$$\boldsymbol{x}(k) = (\bar{\omega}_z(k), \bar{\dot{\omega}}_y(k), \bar{\omega}_y(k)) \,. \tag{2}$$

The bar indicates the normalization, i.e. $\bar{\omega}_z(k) = \frac{\omega_z(k) - \mu_{\omega_z}}{\sigma_{\omega_z}}$, where μ_{ω_z} and σ_{ω_z} are the constant sample mean and sample standard deviation calculated from the training data. The normalized $\bar{\dot{\omega}}_y(k)$ and $\bar{\omega}_y(k)$ are defined in the same way.

C. Reduced SVM for Real-Time Gait-Phase Detection

SVM has advantages in that it usually does not require a large number of training samples to construct a model compared to alternative classifiers and is usually less affected by the appearance of outliers [29]. Besides, with a clear geometrical interpretation, SVM training can always find the global optimum of the cost function [29]. Therefore, SVM is employed in this work as a nonlinear approach to classify walking data into different gait phases.

SVMs have been shown to be a promising classification model for gait phase detection, while the number of support vectors (SVs) increases with the size of the training dataset [30]. An SVM classifier is constructed by a linear combination of kernel evaluation of SVs, which are the training samples lying within the soft margin of the decision boundaries [31]. Therefore,

Authorized licensed use limited to: Mepco Schlenk Engineering College. Downloaded on August 18,2023 at 04:05:16 UTC from IEEE Xplore. Restrictions apply.

the memory usage of an SVM model and the computation time of each SVM prediction is proportional to the number of SVs. To implement this algorithm in an embedded environment that has limited memory and computational capabilities in real-time, the number of SVs should be as small as possible. To reduce the SVM model size and the execution time while keeping the superior SVM performance, a cascaded k–means clustering method is proposed. The cascaded k-means clustering reduces the number of SVs by selecting a small portion of the most informative learning feature vectors that are close to the decision boundaries in the feature space. The rationale behind this strategy is the following: feature vectors close to the decision boundaries are more likely to become SVs of the final separation hyperplane, while feature vectors lying far from the decision boundaries have less effect on building the final separation hyperplane [31], hence removing the SVs "far away" from the decision hyperplane has a low impact on the performance of the SVM. After removing the feature vectors with fewer contributions from the learning samples, the SVM classifier is constructed on the remaining samples. It has a much-reduced number of SVs and is referred to as RSVM for reduced SVM. Compared to other reduction techniques (e.g., decomposition [32], [33] and geometric [34], [35] techniques), clustering methods do not require a long processing time for reduction, and it also significantly reduces SVM training and testing time.

To distinguish the two boundaries of the feature space and the mapped feature space, in this paper, the decision boundary refers to the class boundary in the original feature space, the separation hyperplane refers to the class boundary in the mapped feature space.

To identify this small subset of most informative feature vectors that can be used as the training samples instead of the entire training data set, an iterative data selection procedure is proposed as shown in Algorithm 1. Consider a training dataset $\mathcal{D}$ composed of N feature-class pairs $(\boldsymbol{x}_i, y_i)$, i.e.,

$$\mathcal{D} = \left\{(\boldsymbol{x}_i, y_i)\,|1 \le i \le N, \boldsymbol{x}_i \in \mathbb{R}^3, y_i \in \{P_1, P_2, P_3\}\right\},$$

where $\boldsymbol{x}_i$ is the feature vector obtained from (2) for all subjects and walking trials, and y_i is its corresponding class label. For each iteration number d, given the selected data $\mathcal{D}^d$ from the previous iteration, consider its subset $\mathcal{D}_l^d$ of data with gait phase l. For each gait phase l, an unsupervised k–means clustering technique is applied to the selected samples $\mathcal{D}_l^d$ to identify the predefined c_l^d number of clusters $\mathcal{C}$. c_l^d is chosen to be $\lfloor \eta_1 N_l^d \rfloor$ and is updated at each iteration, where $\lfloor \cdot \rfloor$ is the floor function, $\eta_1 \in (0, 1)$, and N_l^d is the number of samples in gait phase l at iteration d. In each cluster, the samples that are far from the cluster boundary are more likely to be far from the decision boundaries of SVM, and thus are more likely to become insignificant SVs, i.e., SVs with small weight. Therefore, those samples are removed for further iterations. To this end, for each cluster, the Euclidean distances from the cluster centroid to the samples are calculated and sorted from closest to furthest for further sample removal. To prevent excessive removal of samples that may become SVs around the cluster boundaries, a safety region close to the cluster boundaries is defined by the cluster radius, i.e., the distance between the cluster centroid and the furthest data point of the cluster, and the number of samples within the cluster as follows. If the cluster radius is smaller than the predefined radius threshold ρ, i.e., the cluster centroid is closer to the cluster boundary, we remove $\eta_2^d \cdot 100\%$ percent of samples with a smaller distance to the centroid, i.e., samples whose distances to the cluster centroid are below η_2^d–th percentile of the ordered list are removed; If the cluster radius is larger than ρ, i.e., the cluster centroid is further from the cluster boundary, we remove $\eta_3^d \cdot 100\%$ percent of samples with a smaller distance to the centroid, where $0 < \eta_2^d < \eta_3^d < 1$ (samples whose distances to the cluster centroid are below η_3^d–th percentile of the ordered list are removed). The samples within this safety region will be kept. The samples within the cluster but outside the safety region will be removed. The remaining samples within the safety regions from all clusters are collected into $\mathcal{D}^{d+1}$ and used for SVM training. After training, the number of the obtained SVs N_{SV}^{d+1} is compared to the expected value of the SV number δ, which is required by the embedded memory and CPU, but with small performance degradation. Sample reduction threshold, η_2^d, η_3^d, will be decreased after each iteration d by a factor of $1 - \varepsilon$, where $0 < \varepsilon < 1$. We iterate d until the number of SVs, N_{SV}^d, satisfies the requirement of the embedded environment, which leads to a reduced dataset $\mathcal{D}^* = \{(\boldsymbol{x}_i, y_i)|1 \le i \le m, \}$, where i denotes the sample index of the reduced dataset, m represents the number of the remaining samples. All the thresholds and rules are determined empirically based on the training data.

D. Finite State Machine

The SVM and RSVM show good performance for gait phase classification, while it does not consider the temporal sequence of the gait phases. Therefore, some illogical misclassifications occur between the successive gait phases, especially close to the phase transition point. To address this issue, an FSM is designed to post-process the decision from the RSVM by considering the temporal gait phase sequence and making full use of the past information.

As illustrated in Fig. 3, each gait phase of HO-TO, TO-HS, and HS-HO is considered as a state $\{P_1, P_2, P_3\}$ in the FSM. As gait phases occur in a repetitive temporal sequence during walking, there are only three allowed state transitions i.e., $P_1 \to P_2$, $P_2 \to P_3$, and $P_3 \to P_1$. Among those three state transitions, only the transition, with the previous state duration longer than a predefined time threshold τ, is considered as a valid transition. Other state transitions, e.g., $P_1 \to P_3$, $P_2 \to P_1$, and $P_3 \to P_2$, which cannot happen in (normal) walking, are considered as a forbidden transition and are rejected by the FSM. The procedure of the FSM is shown in Algorithm 2. $P(k) \in \{P_1, P_2, P_3\}$ denotes the RSVM predicted gait phase at time step k. P_E is the allowed transition state from the previous state. T represents the time counter, and Δt is the sampling time. $P_v(k)$ represents the valid gait phase at time step k.

Authorized licensed use limited to: Mepco Schlenk Engineering College. Downloaded on August 18,2023 at 04:05:16 UTC from IEEE Xplore. Restrictions apply.

Algorithm 1. Iterative Training Data Reduction Procedure for RSVM Based on k-Means Clustering.

Input: learning dataset $\mathcal{D}$
initialize:
  $d \leftarrow 0$, $\mathcal{D}^0 = \mathcal{D}$, $N_{SV}^0 \leftarrow 30964$, $\delta \leftarrow 3800$, $\eta_1 \leftarrow 0.004$, $\eta_2^0 \leftarrow 0.7$, $\eta_3^0 \leftarrow 0.8$, $\rho \leftarrow 1.0$, $\varepsilon \leftarrow 0.2$;
end
while $N_{SV}^d > \delta$ **do**
  for $l = P_1, P_2, P_3$ **do**
    let set $\mathcal{D}_l^d$ contain the samples in $\mathcal{D}^d$ with gait phase l, and $N_l^d = |\mathcal{D}_l^d|$;
    Set $c_l^d \leftarrow \lfloor \eta_1 N_l^d \rfloor$;
    identify a set $\mathcal{C}$ of c_l^d clusters using k–means clustering on samples $\mathcal{D}_l^d$;
    for each cluster $\in \mathcal{C}$ **do**
      create an ordered list (sorted from least to greatest) that contains the Euclidean distances from the cluster centroid to each sample in the cluster;
      if cluster radius $> \rho$ **then**
        remove samples whose distances to the cluster centroid are below η_2^d–th percentile of the ordered list;
      else
        remove samples whose distances to the cluster centroid are below η_3^d–th percentile of the ordered list;
      end if
    end for
  end for
  update $\eta_2^{d+1} \leftarrow (1-\varepsilon) \cdot \eta_2^d$;
  update $\eta_3^{d+1} \leftarrow (1-\varepsilon) \cdot \eta_3^d$;
  update $\mathcal{D}^{d+1}$ by collecting all the remaining samples of all the clusters from all three gait phases;
  update N_{SV}^{d+1} by SVM trained on $\mathcal{D}^{d+1}$;
  update $d \leftarrow d+1$;
end while
Output: reduced data set $\mathcal{D}^*$.

Algorithm 2. Pseudocode of FSM for Real-Time Gait Phase Detection.

Input: gait phase at previous time step $P(k-1)$, current time step gait phase $P(k)$.
initialize:
  $T \leftarrow 0$;
end
begin
  if $P(k) == P(k-1)$ **then**
    no state transition;
    set $P_v(k) \leftarrow P(k)$;
    update $T \leftarrow T + \Delta t$;
  else if $P(k) == P_E$ **then**
    if $T > \tau$ **then**
      valid state transition;
      set $P_v(k) \leftarrow P(k)$;
      reset $T \leftarrow 0$;
    else
      invalid state transition;
      set $P_v(k) \leftarrow P(k-1)$;
      update $T \leftarrow T + \Delta t$;
    end if
  else
    forbidden state transition;
    set $P_v(k) \leftarrow P(k-1)$;
    update $T \leftarrow T + \Delta t$;
  end if
end
Output: valid gait phase at current state, $P_v(k)$.

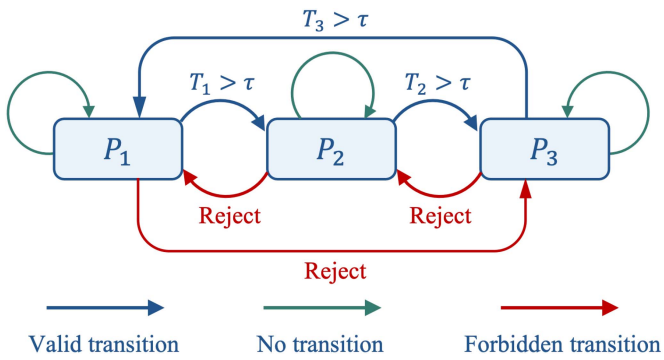

Fig. 3. A schematic illustration of algorithm FSM. P_1, P_2, P_3 correspond to the three gait phases HO-TO, TO-HS, HS-HO, respectively. They are considered as three states of FSM. T_1, T_2, T_3 are the state duration for three gait phases. τ is a predefined time constant. The blue arrows represent valid transitions, the green arrows represent no transition, the red arrows represent forbidden transitions.

E. System Implementation

The kernel function used in this paper is the radial basis function (RBF) [36], which is:

$$\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left(-\gamma \|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2\right) = \exp(\hat{x}_{ij}), \quad (3)$$

where $\hat{x}_{ij} = -\gamma\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2 \leq 0$, $\forall\, \boldsymbol{x}_i, \boldsymbol{x}_j \in \mathcal{D}, \forall\, \gamma > 0$. The rationales behind this choice are as follows. First, RBF kernel is translation invariant, i.e., $\kappa(\boldsymbol{x}_i, \boldsymbol{y}_i) = \kappa(\boldsymbol{x}_i + h, \boldsymbol{y}_i + h)$, where h is any arbitrary vector. This property allows the data points to maintain the same similarity, when the entire data set is shifted without changing the relative positions of the data points. Second, unlike linear, polynomial, and sigmoid kernels which are functions of inner products of the data points, the RBF kernel is a function of the Euclidean distance between the data points. Third, the RBF kernel is the normalized kernel, which holds the property that $\kappa(\boldsymbol{x_i}, \boldsymbol{x_j}) = 1$, when $\boldsymbol{x_i} = \boldsymbol{x_j}$.

In practice, the evaluation of the exponential function is an expensive operation, which is a matter of concern in particular on resource-limited platforms such as microcontrollers. To decrease the computational complexity, we approximate the RBF

Authorized licensed use limited to: Mepco Schlenk Engineering College. Downloaded on August 18,2023 at 04:05:16 UTC from IEEE Xplore. Restrictions apply.

kernel function by another function $g(\hat{x}_{ij})$ defined as follows:

$$\exp(\hat{x}_{ij}) \approx g(\hat{x}_{ij}) := \begin{cases} 0 & \text{if } \hat{x}_{ij} \le \chi_1 \\ \exp(\hat{x}_{ij}) & \text{if } \chi_1 < \hat{x}_{ij} < \chi_2 \\ 1 + \sum_{n=1}^{8} \frac{(\hat{x}_{ij})^n}{n!} & \text{if } \chi_2 \le \hat{x}_{ij} \end{cases} \tag{4}$$

where χ_1 and χ_2 are the non-positive constants that determine the region of the approximation. When the exponent $\hat{x}_{ij}$ is small, the kernel is close to zero and thus is approximated to 0 in (4). When $\hat{x}_{ij}$ is close to zero, the Taylor series expansion of the exponential function is a good approximation with low computation cost as considered in (4). The degree of the Taylor series expansion, i.e., 8, is chosen by a trade-off between the approximation precision and the calculation speed. The constants χ_1 and χ_2 in (4) are determined such that the relative error between $g(\hat{x}_{ij})$ and $\exp(\hat{x}_{ij})$ is less than 10% for any $\hat{x}_{ij} \in (-\infty, 0]$. The values of the χ_1 and χ_2 used in this paper are -7.596, -2.205.

Both SVM and RSVM are constructed using Python library `sklearn`[1] on the first dataset acquired from treadmill walking. The hyperparameters of these models are optimized by the grid search approach with cross–validation [37]. After training with optimized hyperparameters, the RSVM model is transposed to C using `sklearn-porter`[2] in order to implement it on the MCU. The final, compiled model size of the RSVM is 180 KB.

IV. Results and Discussion

The gait phase detection model is trained and optimized using training data containing 75% samples of the first dataset that were collected from treadmill walking. To evaluate the performance of the classifiers for gait phase detection, three metrics are employed: accuracy, sensitivity, and specificity.

A. Offline Performance of Gait Phase Detection

The offline performance of RSVM is first assessed using the testing data containing 25% samples of the first dataset, which were collected on a treadmill at three different walking speeds: 0.53 m/s, 0.86 m/s, 1.11 m/s. The classification results are compared to the SVM as stated in Table I. The performance metrics of the classification for each gait phase are calculated by gathering all walking data from all subjects. It can be observed that as the number of SVs is reduced as a result of k–means clustering, sensitivity, specificity, and accuracy of RSVM are decreased by roughly 4%, 2%, and 5%, respectively when compared to SVM.However, the model size of RSVM and the computation time running on the microcontroller for each prediction of RSVM are significantly reduced by 88% and 97%, as shown in Table II and Fig. 4.

The SVM has a total of 30964 SVs, and it takes up about 1.5 MB of memory on the embedded system. After reducing the model size of the SVM, the number of the SVs for RSVM is 3793. The RSVM only takes up 180 KB of memory on the embedded system. To evaluate the computational efficiency of the real-time algorithm implementation, a timing experiment for 100 gait phase predictions running on the microcontroller is conducted. The average execution time for each phase prediction is 557367 μs and 15348 μs for SVM and RSVM, respectively. Hence, with a small degradation in the classification performance when comparing RSVM to SVM, the number of SVs and the model size are reduced by about 88%, and the computation speed for each prediction on the microcontroller is increased by 36 times. RSVM's model size and computational speed satisfy our requirements for real-time gait phase prediction on the microcontroller, i.e., the model size is less than 200 KB, and the prediction frequency is greater than 25 Hz.

In Fig. 5, an example of gait phase detection results for offline RSVM is illustrated, and recall that the true phases in Fig. 5(a) and (b) are labeled by the rule-based algorithm. Though RSVM shows good classification performance for the

TABLE I
The Comparison of the Offline Classification Performance Metrics for SVM and RSVM Based on the Testing Dataset Collected on a Treadmill

Gait Phase	Sensitivity		Specificity		Accuracy	
	SVM	RSVM	SVM	RSVM	SVM	RSVM
P_1	91.67%	90.75%	99.03%	95.89%	96.11%	91.26%
P_2	97.20%	93.51%	98.74%	96.14%		
P_3	97.82%	90.05%	96.02%	94.74%		
Overall	95.56%	91.44%	97.93%	95.59%		

TABLE II
The Comparison of the Number of SVs, the Model Size and the Computation Time for Each Prediction on Microcontroller Between SVM and RSVM

Classifier	N_{SV}	Model Size	Computation Time
SVM	30964	1.5 MB	557367 μs
RSVM	3793	180 KB	15348 μs

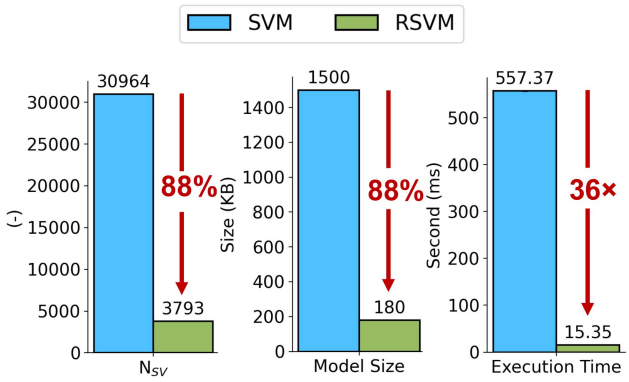

Fig. 4. The comparison of RSVM and SVM model property.

[1]https://github.com/scikit-learn/scikit-learn
[2]https://github.com/nok/sklearn-porter

Authorized licensed use limited to: Mepco Schlenk Engineering College. Downloaded on August 18,2023 at 04:05:16 UTC from IEEE Xplore. Restrictions apply.

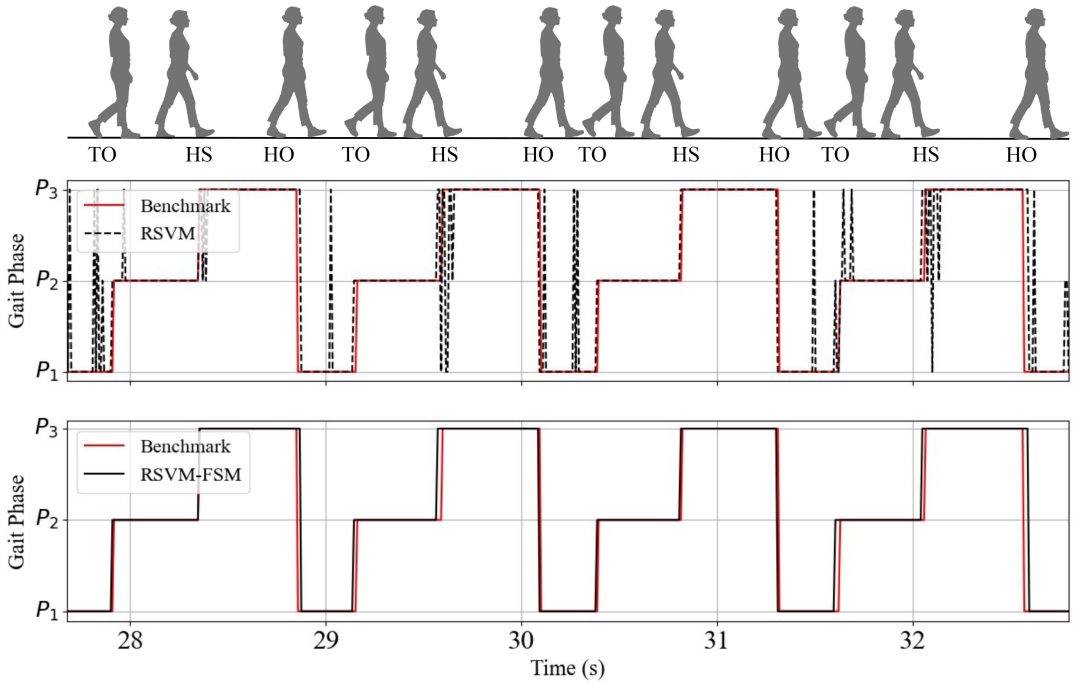

Fig. 5. Performance illustration of RSVM and RSVM–FSM for offline gait phase detection. (a) Gait phases detection results from RSVM. (b) Gait phases detection results from RSVM–FSM.

TABLE III
THE OFFLINE CLASSIFICATION PERFORMANCE METRICS OF RSVM–FSM BASED ON THE TESTING DATASET COLLECTED FROM TREADMILL

Gait Phase	Sensitivity	Specificity	Accuracy
	RSVM–FSM	RSVM–FSM	RSVM–FSM
P_1	91.22%	97.14%	92.35%
P_2	93.05%	94.83%	
P_3	90.27%	96.23%	
Overall	92.12%	96.23%	

gait phase detection, as shown in Fig. 5(a), there are systematic misclassifications for $\mathrm{P}_2 \rightarrow \mathrm{P}_3$, and $\mathrm{P}_3 \rightarrow \mathrm{P}_1$ transitions. This phenomenon is discussed in Section III-D. To address this issue, the developed FSM is incorporated into the RSVM, which leads to the results illustrated in Fig. 5(b). Fig. 5 provides a clear visualization of the comparison results between the classification performance of RSVM incorporated with and without FSM. We can see that using the FSM can eliminate almost all the misclassifications of RSVM. The detailed performance metrics of RSVM–FSM based on testing data from the first dataset are listed in Table III. Compared to the RSVM in Table I, RSVM-FSM shows improved overall performance, but for P_2 gait phase, the values of sensitivity and specificity are slightly reduced. Since the intention of the FSM design is to correct the illogical gait phase transitions in the temporal sequence, such slight performance degradation is acceptable from an application perspective. Therefore, the RSVM–FSM model is employed for real-time gait phase detection on the microcontroller.

B. Inter-Subject Real-Time Performance for Gait Phase Detection

Attributed to the high computational efficiency of the real-time RSVM-FSM algorithm, the entire process within one sampling interval on the microcontroller, including sensor data reading, gait phase prediction, and prediction results logging, can be completed within 20 ms. Considering that other processing such as wireless communication also occupied the microcontroller, the sampling time for the real-time gait phase prediction and prediction result logging is set to 30 ms (33.33 Hz). While as stated in Section II-C, raw sensor data are still logged at 100 Hz.

To assess the inter-subject real-time performance of RSVM-FSM for gait phase detection, the recorded real-time (on the fly) prediction results are compared to the benchmark, i.e., labels identified by the reference algorithm applied to the raw sensor data. Fig. 6 presents the confusion matrix of the real-time RSVM-FSM classified results by gathering all walking data from twenty subjects. Table IV summarizes the three performance metrics calculated from gathering all walking data from all twenty subjects' walking trials and three different walking speeds. We can see that the real-time RSVM-FSM algorithm achieves promising gait phase detection results with the sensitivity, specificity, and accuracy values of 91.74%, 95.79%, 91.55%, respectively. Fig. 7 presents an intuitive comparison of the gait phases detected by real-time RSVM-FSM and benchmarks that are obtained by the offline reference algorithm. In Fig. 9, a violin

Authorized licensed use limited to: Mepco Schlenk Engineering College. Downloaded on August 18,2023 at 04:05:16 UTC from IEEE Xplore. Restrictions apply.

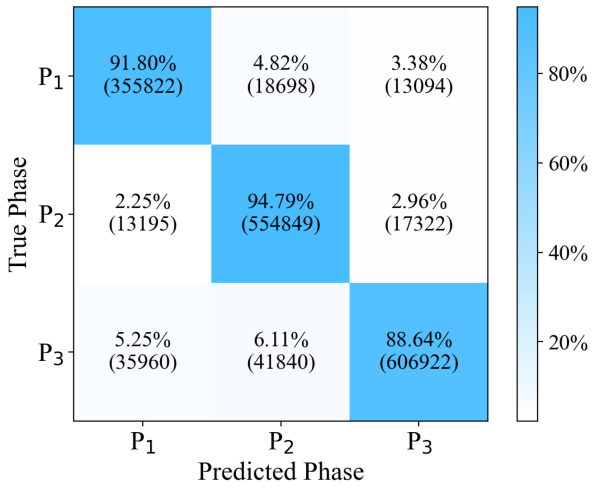

Fig. 6. Confusion matrix of real-time RSVM-FSM results for gait phase classification of twenty subjects.

TABLE IV
THE REAL-TIME CLASSIFICATION PERFORMANCE METRICS OF RSVM–FSM FOR LONG-TERM FREE WALKING

Gait Phase	Sensitivity	Specificity	Accuracy
	RSVM–FSM	RSVM–FSM	RSVM–FSM
P_1	91.80%	96.13%	91.55%
P_2	94.79%	94.35%	
P_3	88.64%	96.88%	
Overall	91.74%	95.79%	

plot of walking speed for offline model training and real-time model testing is presented. Each dot represents a walking bout with an average speed at the dot.

To compare the performance of RSVM-FSM and SVM-FSM, the raw sensor data are fed into the offline SVM-FSM classifier. The classified gait phases of SVM-FSM are compared to the benchmark. The three performance metrics are calculated by averaging the walking bouts over all twenty subjects and three different walking speeds for both RSVM-FSM and SVM-FSM. The box plot of the performance metrics of different subjects is shown in Fig. 8. Comparing to offline performance of SVM-FSM with an accuracy of 91.55% $\pm$ 2.23%, a sensitivity of 91.16% $\pm$ 2.15%, a specificity of 95.73% $\pm$ 0.98%, the performance of real-time RSVM-FSM presents an insignificant degradation, which are 91.39% $\pm$ 1.67%, 91.59% $\pm$ 1.59%, 95.67% $\pm$ 0.78% for accuracy, sensitivity, and specificity, respectively.

To assess the robustness of the real-time RSVM-FSM algorithm with varying gait parameters, the real-time classification performance is analyzed under different walking speeds, cadences, and stride lengths. The results of the generalizability analysis are shown in Fig. 11. Note that the exact stride velocity, cadence and stride length of each data set for each subject are calculated using the offline reference algorithm after the walking experiment. In the left subplot of Fig. 11, the three performance metrics for three different walking speed levels are presented. The walking speed v_w for each walking bout of every subject is calculated from the raw sensor data after the experiment. After ordering all walking bouts' walking speeds from slowest to fastest, the walking bouts are divided into three categories, i.e., low, medium, high walking speeds. The low walking speed category contains the walking bouts whose walking speed is slower than the first 3-quantile ($v_w \leq 1.21\,\mathrm{m/s}$) of the ordered list. The medium walking speed category contains the walking bouts whose walking speed is between the first and the second 3-quantile ($1.21\,\mathrm{m/s} < v_w \leq 1.43\,\mathrm{m/s}$). The high walking speed category contains the remaining walking bouts ($v_w > 1.43\,\mathrm{m/s}$). All three quantiles of cadence and stride length are defined in the same way. The cadence quantile bounds are found to be $104.84\,\mathrm{SPM}$ (steps per minute) and $115.10\,\mathrm{SPM}$. The stride-length quantile bounds are found to be $1.37\,\mathrm{m}$ and $1.54\,\mathrm{m}$.

C. Discussion

When comparing the offline classification performance of SVM and RSVM as shown in Table I, SVM shows a slightly better performance in the gait phase detection than RSVM. This is because the RSVM hyperplane is constructed only using potential SV data points. Due to the lower density of boundary data points in smaller training datasets, it may not be enough to provide sufficient separation information of the training dataset, resulting in a slight performance drop in RSVM classification. In addition, this slight performance drop of RSVM is also due to the significant reduction of the number of SVs, which is decreased by 88% in total. While comparing to the SVM, the execution time of RSVM has reduced more than 36 times as the result of a considerable reduction in the number of SVs, as shown in Table II.

From Fig. 5(a) we can see that many illogical gait phase transitions are misclassified, especially near the transitions from one phase to the next. This is due to the fact that despite gait being a periodic motion, the RSVM processes the continuous gait signal in sections, neglecting the implicit information carried by the gyroscope as a whole, and thus ignoring the temporal sequence of gait phases. The developed FSM provides the RSVM classifier with a logical temporal sequence. After applying the FSM, the misclassified illogical gait phase transitions in Fig. 5(a) are corrected into a regular sequence of gait phase as shown in Fig. 5(b). These results demonstrate the advantages and the effectiveness of the developed FSM, which leads to a solid temporal sequence of sub-gait phases.

The algorithm's real-time (on the fly) performance is evaluated with long-term real-world free-living walking on the street, as opposed to the supervised walking environment for model training data, in which subjects walked on a treadmill in a laboratory. In the daily living environment, the human gait is very dynamic, involving irregular gait speeds, fluctuating surface, and varying surface inclinations. Fig. 9 shows the walking speeds of the datasets for offline algorithm training and online algorithm testing, respectively. Each dot represents a walking bout with an average walking speed at that dot. The walking speed of

Authorized licensed use limited to: Mepco Schlenk Engineering College. Downloaded on August 18,2023 at 04:05:16 UTC from IEEE Xplore. Restrictions apply.

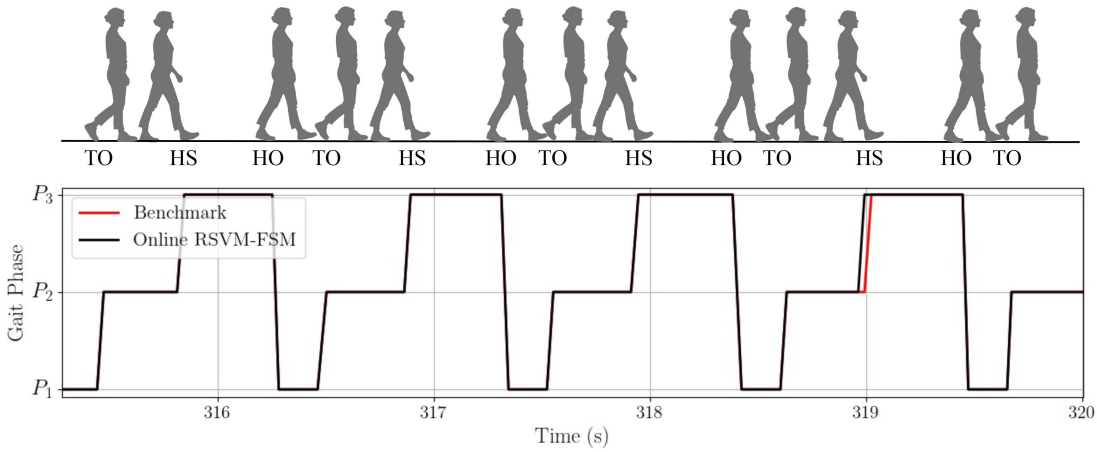

Fig. 7. An intuitive comparison of the gait phases detected by real-time RSVM-FSM and the benchmark.

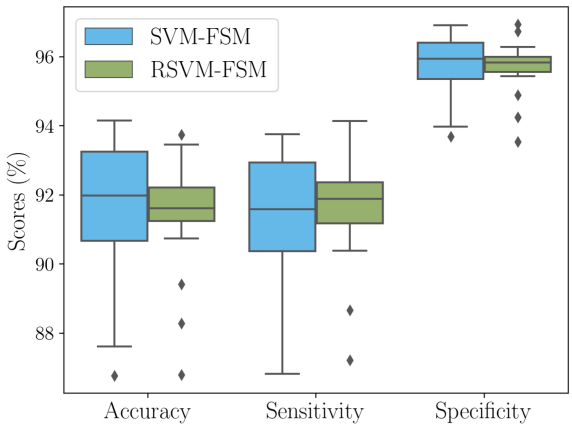

Fig. 8. Performance comparison of real-time RSVM-FSM and offline SVM-FSM gait phase detection for twenty subjects.

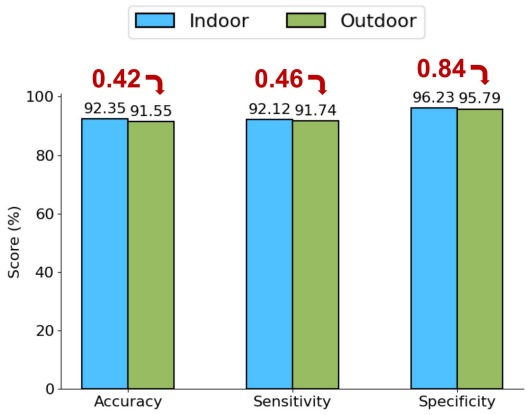

Fig. 10. The performance comparison of real-time RSVM-FSM for outdoor free-living walking and offline SVM-FSM for indoor treadmill walking.

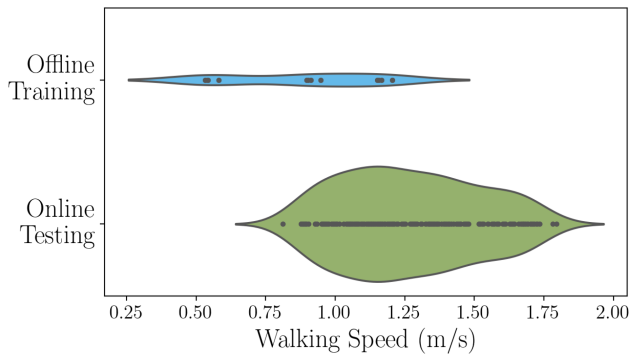

Fig. 9. The violin plot for walking speed distribution of offline training and real-time testing. Each dot represents one walking bout. The width represents the density of the data size. Wider areas of the violin plot constitute a higher density of data taking a given walking speed, whereas the thinner areas correspond to a lower density.

each data set is calculated using the offline post-processing algorithm [23]. We can see that the walking speeds of datasets used for algorithm training are lying closely around the treadmill speeds that were set at 0.53 m/s, 0.86 m/s, and 1.11 m/s. However, the walking speeds for real-time testing range from 0.81 m/s to 1.80 m/s. There is only a small overlap between the walking speed of training and testing data, which shows the excellent performance of the model when encountering unseen data. Compared to the prediction results of RSVM-FSM shown in Table III, which is intra-subject supervised walking, there is only a minor performance degradation of 0.42%, 0.46%, 0.84% in the inter-subject real-world uncontrolled free-living walking real-time prediction results shown in Table IV, for sensitivity, specificity, and accuracy. When comparing the real-time phase detection performance of outdoor walking to the offline phase detection performance of indoor walking, there is only a slight performance degradation of less than 1%, as shown in Fig. 10. This is more robust than the offline algorithms reviewed in [22], which exhibit a performance degradation of more than 15% when transfer walking from laboratory to the real-world. These results exhibit the superior capability of the proposed RSVM-FSM algorithm to identify gait phases in real-time (on the fly) both indoor and outdoor (real-world free-living walking), as well as the wide range of walking speeds it can

Authorized licensed use limited to: Mepco Schlenk Engineering College. Downloaded on August 18,2023 at 04:05:16 UTC from IEEE Xplore. Restrictions apply.

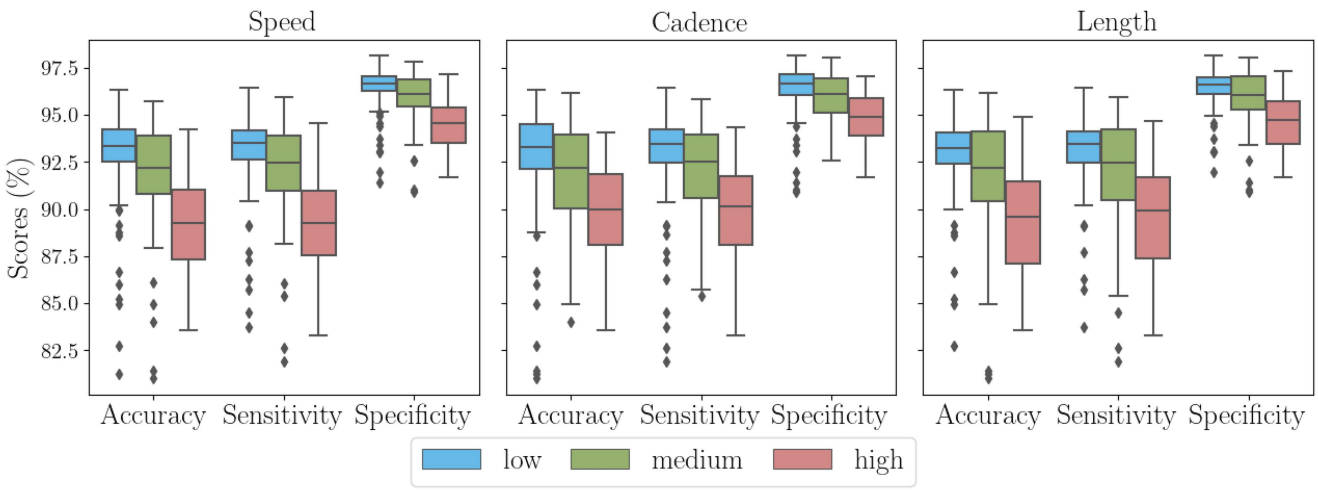

Fig. 11. The RSVM-FSM's performance of real-time gait phase detection in relation to various gait parameters, i.e., gait speed, cadence, stride length.

recognize. The results also show that the algorithm's promising performance is consistent with different walking subjects, changing walking speeds, and varying walking environments, including less controlled walking scenarios. It demonstrates the robustness and generalizability of the classifier for both intra-subject and inter-subject walking, and both indoor (controlled) and outdoor (real-world uncontrolled) free-living walking conditions.

The prediction results in Fig. 11 demonstrate a promising generalization performance of the real-time RSVM-FSM algorithm for varying gait parameters, i.e., gait speed, cadence, and stride length. As shown in the left subplot of Fig. 11, the RSVM-FSM classifier performs well at all walking speeds. The minor performance degradation when the walking speeds increase may be due to the fact that the walking speeds in the training data are slow, ranging from $0.53\,\mathrm{m/s}$ to $1.11\,\mathrm{m/s}$, whereas the walking speeds in the medium and high category for real-time testing are higher than $1.21\,\mathrm{m/s}$. In terms of the generalization of the proposed algorithm regarding varied walking cadences, the classifier also shows consistent performance as demonstrated in the middle subplot of Fig. 11. It can be observed that the performance distributions among the three cadence groups are overlapping. No obvious performance decrease is observed when increasing the walking cadence. A similar performance level is obtained for classifying gait phases among different stride length groups, indicating that the proposed RSVM-FSM is robust to the various stride length, as well as the gait cadence. The slight performance drop of the high stride length group could be explained by the fact that the classifier is constructed by the data collected from the treadmill, where stride length is limited by the walking surface of the belt.

In addition, the algorithm only uses three signals to classify the gait phases, all of which come from a single gyroscope only. This reduces the amount of information used to characterize the gait cycle compared to the studies that use both gyroscope and accelerometers [24], and thus saves the energy and the cost of the wearable device. Moreover, no prior calibration is required for the sensor data, which is a significant improvement compared to the studies that need additional calibration procedures [38].

The limitation of this study is that we assume the input activity data only contains walking rather than other activities, as the algorithm only classifies the walking data into predefined three gait phases. Hence, currently, the RSVM-FSM algorithm presented herein will output a gait phase even when the subject is not walking. The authors are currently working on a new algorithm for human activity recognition to filter out the walking data while rejecting other activity data in real-time. The new algorithm will detect the initiation and termination of walking among different human activities. By combining the two algorithms, it will be possible to monitor human activities and provide instant haptic feedback in long-term daily life. Another limitation of this study is that the proposed real-time algorithm was validated on twenty healthy subjects walking in a real-world scenario. A quantitative validation on the elderly and subjects with gait disorders should be considered in future work to provide more insights into the algorithm's ability to generalize to a wider range of gait patterns.

In the follow-up study, the actuator of the system will be exploited to deliver real-time vibrotactile feedback based on users' gait patterns by using the real-time algorithm proposed in this work. The clinical effect of real-time vibrotactile feedback on gait will also be investigated in the follow-up clinical study. Variations of the FSM, such as introducing another duration constraint to allow for back transition of the gait phase and the impact of this latency, can also be investigated.

V. Conclusion

In this paper, we propose a real-time algorithm that can identify gait phase detection in real-time on memory- and computation-limited microcontrollers of wearable devices for both indoor and outdoor free-living walking. We developed a cascaded k-means clustering approach that reduces the model size of the standard SVM classifier by 88% and to increase the computational speed of gait phase prediction on

Authorized licensed use limited to: Mepco Schlenk Engineering College. Downloaded on August 18,2023 at 04:05:16 UTC from IEEE Xplore. Restrictions apply.

microcontrollers (ESP32) by a factor of 36 with negligible degradation in SVM performance. The proposed algorithm was implemented on a microcontroller of a wearable device and was validated in real-time (on the fly) by real-world free-living walking performed by twenty healthy subjects in their daily living environment. The algorithm shows a promising real-time gait phase detection performance, with a total accuracy of 91.51%, a sensitivity of 91.70%, and a specificity of 95.77%. It does not require any prior calibration for the sensor data and shows a good generalization performance for different walking environments, speeds, cadences, and stride lengths. It demonstrates great potential for real-time rehabilitation applications, such as FES/EES, and haptic feedback systems.

Conflict of Interest: B.J.N. and S.P. are shareholders in Magnes AG.

References

[1] S. Fritz and M. Lusardi, "White paper: "walking speed: The sixth vital sign"," *J. Geriatr. Phys. Ther.*, vol. 32, no. 2, pp. 2–5, 2009.

[2] S. Lord, B. Galna, and L. Rochester, "Moving forward on gait measurement: Toward a more refined approach," *Movement Disord.*, vol. 28, no. 11, pp. 1534–1543, 2013.

[3] P. Gray and K. Hildebrand, "Fall risk factors in Parkinson's disease," *J. Neurosci. Nurs.*, vol. 32, no. 4, 2000, Art. no. 222.

[4] H. Jacobi et al., "Spinocerebellar ataxia types 1, 2, 3 and 6: The clinical spectrum of ataxia and morphometric brainstem and cerebellar findings," *Cerebellum*, vol. 11, no. 1, pp. 155–166, 2012.

[5] L. Nürnberger et al., "Ultrasound-based motion analysis demonstrates bilateral arm hypokinesia during gait in heterozygous pink1 mutation carriers," *Movement Disord.*, vol. 30, no. 3, pp. 386–392, 2015.

[6] M. Pistacchi et al., "Gait analysis and clinical correlations in early Parkinson's disease," *Funct. Neurol.*, vol. 32, no. 1, pp. 28–34, 2017.

[7] H. Prasanth et al., "Wearable sensor-based real-time gait detection: A systematic review," *Sensors*, vol. 21, no. 8, 2021, Art. no. 2727.

[8] M. R. Popovic, T. Keller, I. Papas, V. Dietz, and M. Morari, "Surface-stimulation technology for grasping and walking neuroprostheses," *IEEE Eng. Med. Biol. Mag.*, vol. 20, no. 1, pp. 82–93, Jan./Feb. 2001.

[9] R. Kobetic and E. Marsolais, "Synthesis of paraplegic gait with multichannel functional neuromuscular stimulation," *IEEE Trans. Rehabil. Eng.*, vol. 2, no. 2, pp. 66–79, Jun. 1994.

[10] W. Liberson, "Functional electrotherapy: Stimulation of the peroneal nerve synchronized with the swing phase of the gait of hemiplegic patients," *Arch Phys Med*, vol. 42, pp. 101–105, 1961.

[11] F. B. Wagner et al., "Targeted neurotechnology restores walking in humans with spinal cord injury," *Nature*, vol. 563, no. 7729, pp. 65–71, 2018.

[12] S. Harkema et al., "Effect of epidural stimulation of the lumbosacral spinal cord on voluntary movement, standing, and assisted stepping after motor complete paraplegia: A case study," *Lancet*, vol. 377, no. 9781, pp. 1938–1947, 2011.

[13] R. Caldas, M. Mundt, W. Potthast, F. B. de Lima Neto, and B. Markert, "A systematic review of gait analysis methods based on inertial sensors and adaptive algorithms," *Gait Posture*, vol. 57, pp. 204–210, 2017.

[14] H. Zhao et al., "Adaptive gait detection based on foot-mounted inertial sensors and multi-sensor fusion," *Inf. Fusion*, vol. 52, pp. 157–166, 2019.

[15] D. Gouwanda and A. A. Gopalai, "A robust real-time gait event detection using wireless gyroscope and its application on normal and altered gaits," *Med. Eng. Phys.*, vol. 37, no. 2, pp. 219–225, 2015.

[16] A. Mannini, V. Genovese, and A. M. Sabatini, "Online decoding of hidden Markov models for gait event detection using foot-mounted gyroscopes," *IEEE J. Biomed. Health Inform.*, vol. 18, no. 4, pp. 1122–1130, Jul. 2014.

[17] W. Chen, Y. Xu, J. Wang, and J. Zhang, "Kinematic analysis of human gait based on wearable sensor system for gait rehabilitation," *J. Med. Biol. Eng.*, vol. 36, no. 6, pp. 843–856, 2016.

[18] H. T. T. Vu, F. Gomez, P. Cherelle, D. Lefeber, A. Nowé, and B. Vanderborght, "ED-FNN: A new deep learning algorithm to detect percentage of the gait cycle for powered prostheses," *Sensors*, vol. 18, no. 7, 2018, Art. no. 2389.

[19] M. Shushtari, H. Dinovitzer, J. Weng, and A. Arami, "Ultra-robust real-time estimation of gait phase," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 30, pp. 2793–2801, 2022.

[20] I. Kang, D. D. Molinaro, S. Duggal, Y. Chen, P. Kunapuli, and A. J. Young, "Real-time gait phase estimation for robotic hip exoskeleton control during multimodal locomotion," *IEEE Robot. Automat. Lett.*, vol. 6, no. 2, pp. 3491–3497, Apr. 2021.

[21] E. Warmerdam et al., "Long-term unsupervised mobility assessment in movement disorders," *Lancet Neurol.*, vol. 19, no. 5, pp. 462–470, 2020.

[22] S. Khandelwal and N. Wickström, "Evaluation of the performance of accelerometer-based gait event detection algorithms in different real-world scenarios using the marea gait database," *Gait Posture*, vol. 51, pp. 84–90, 2017.

[23] J. Wu et al., "An intelligent in-shoe system for gait monitoring and analysis with optimized sampling and real-time visualization capabilities," *Sensors*, vol. 21, no. 8, 2021, Art. no. 2869.

[24] J. Wu et al., "Human gait-labeling uncertainty and a hybrid model for gait segmentation," *Front. Neurosci.*, vol. 16, 2022. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fnins.2022.976594

[25] J. A. DeLisa, *Gait Analysis in the Science of Rehabilitation*. Collingdale, PA, USA: Diane Publishing, 1998.

[26] J. Rueterbories, E. G. Spaich, B. Larsen, and O. K. Andersen, "Methods for gait event detection and analysis in ambulatory systems," *Med. Eng. Phys.*, vol. 32, no. 6, pp. 545–552, 2010.

[27] T. O'Haver and T. Begley, "Signal-to-noise ratio in higher order derivative spectrometry," *Anal. Chem.*, vol. 53, no. 12, pp. 1876–1878, 1981.

[28] E. K. Antonsson and R. W. Mann, "The frequency content of gait," *J. Biomech.*, vol. 18, no. 1, pp. 39–47, 1985.

[29] C. J. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining Knowl. Discov.*, vol. 2, no. 2, pp. 121–167, 1998.

[30] I. Steinwart, "Sparseness of support vector machines," *J. Mach. Learn. Res.*, vol. 4, pp. 1071–1105, 2003.

[31] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.

[32] R.-E. Fan, P.-H. Chen, C.-J. Lin, and T. Joachims, "Working set selection using second order information for training support vector machines," *J. Mach. Learn. Res.*, vol. 6, no. 12, 2005.

[33] S. S. Keerthi, S. K. Shevade, C. Bhattacharyya, and K. R. K. Murthy, "Improvements to platt's SMO algorithm for SVM classifier design," *Neural Comput.*, vol. 13, no. 3, pp. 637–649, 2001.

[34] O. N. Almasi and M. Rouhani, "A geometric-based data reduction approach for large low dimensional datasets: Delaunay triangulation in SVM algorithms," *Mach. Learn. Appl.*, vol. 4, 2021, Art. no. 100025.

[35] J. Nalepa and M. Kawulok, "Adaptive memetic algorithm enhanced with data geometry analysis to select training data for SVMs," *Neurocomputing*, vol. 185, pp. 113–132, 2016.

[36] M. J. L. Orr, "Introduction to radial basis function networks," Tech. Rep., Center Cogn. Sci, Univ. Edinburgh, Edinburgh, U.K., 1996. [Online]. Available: https://faculty.cc.gatech.edu/~isbell/tutorials/rbf-intro.pdf

[37] C. W. Hsu, C. C. Chang, and C. J. Lin, "A practical guide to support vector classification," Tech. Rep., Nat. Taiwan Univ., 2003. [Online]. Available: http://www.datascienceassn.org/sites/default/files/Practical%20Guide%20to%20Support%20Vector%20Classification.pdf

[38] M. Goršič et al., "Online phase detection using wearable sensors for walking with a robotic prosthesis," *Sensors*, vol. 14, no. 2, pp. 2776–2794, 2014.

Authorized licensed use limited to: Mepco Schlenk Engineering College. Downloaded on August 18,2023 at 04:05:16 UTC from IEEE Xplore. Restrictions apply.