



# Article Detecting Gait Events from Accelerations Using Reservoir Computing

Laurent Chiasson-Poirier <sup>1,\*</sup>, Hananeh Younesian <sup>2</sup>, Katia Turcot <sup>2</sup> and Julien Sylvestre <sup>1</sup>

- <sup>1</sup> Interdisciplinary Institute for Technological Innovation (3IT), Department of Mechanical Engineering, Université de Sherbrooke, Sherbrooke, QC J1K 2R1, Canada
- <sup>2</sup> Centre Interdisciplinaire de Recherche en Réadaptation et Intégration Sociale (Cirris), Department of Kinesiology, Université Laval, Quebec, QC G1M 2S8, Canada

\* Correspondence: laurent.chiasson-poirier@usherbrooke.ca

Abstract: Segmenting the gait cycle into multiple phases using gait event detection (GED) is a wellresearched subject with many accurate algorithms. However, the algorithms that are able to perform accurate and robust GED for real-life environments and physical diseases tend to be too complex for their implementation on simple hardware systems limited in computing power and memory, such as those used in wearable devices. This study focuses on a numerical implementation of a reservoir computing (RC) algorithm called the echo state network (ESN) that is based on simple computational steps that are easy to implement on portable hardware systems for real-time detection. RC is a neural network method that is widely used for signal processing applications and uses a fast-training method based on a ridge regression adapted to the large quantity and variety of IMU data needed to use RC in various real-life environment GED. In this study, an ESN was used to perform offline GED with gait data from IMU and ground force sensors retrieved from three databases for a total of 28 healthy adults and 15 walking conditions. Our main finding is that despite its low complexity, ESN is robust for GED, with performance comparable to other state-of-the-art algorithms. Our results show the ESN is robust enough to obtain good detection results in all conditions if the algorithm is trained with variable data that match those conditions. The distribution of the mean absolute errors (MAE) between the detection times from the ESN and the force sensors were between 40 and 120 ms for 6 defined gait events (95th percentile). We compared our ESN with four different state-of-the-art algorithms from the literature. The ESN obtained a MAE not more than 10 ms above three other reference algorithms for normal walking indoor and outdoor conditions and yielded the 2nd lowest MAE and the 2nd highest true positive rate and specificity when applied to outdoor walking and running conditions. Our work opens the door to using the ESN as a GED for applications in wearable sensors for long-term patient monitoring.

Keywords: gait event detection; reservoir computing; echo state network; IMU sensors

# 1. Introduction

The functional analysis of human locomotion allows the gait cycle to be defined in time intervals [1] called gait phases. They are distinguished by specific gait events. Gait event detection (GED) can thus be considered a fundamental processing step needed for gait analysis. The two main gait events are the heel strike (HS) and the toe off (TO), which, respectively, define the beginning of the stance phase and of the swing phase. Multiple clinical analysis and applications use gait events, such as foot clearance estimation [2], to monitor the physical behaviour of participants with neurological [3,4] or osteoarthritic [4] disease. Some of the clinical applications requiring real-time GED include functional electrical stimulation [5,6] and gait retraining [7,8]. Among several technologies used for GED, user-friendly, light, wireless and compact inertial measurement unit (IMU) sensors are crucial to provide follow-up in the daily movements of the participants. IMU sensors have thus



**Citation:** Chiasson-Poirier, L.; Younesian, H.; Turcot, K.; Sylvestre, J. Detecting Gait Events from Accelerations Using Reservoir Computing. *Sensors* **2022**, *22*, 7180. https://doi.org/10.3390/s22197180

Academic Editors: Lennart Scheys and Ilse Jonkers

Received: 1 August 2022 Accepted: 14 September 2022 Published: 21 September 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https:// creativecommons.org/licenses/by/ 4.0/). become an essential tool for research and diagnosis. However, the accelerations captured by an IMU sensor are exposed to variable movements (nominal cadence, regularity, symmetry) inherent to the ecological environment and pathological walking conditions. Pathologies, disease progression and participant fatigue contribute to the movement variation and the speed of the gait. External conditions, such as the type of ground and the walking trajectory, can also change the gait dynamics. Several complex IMU-based GED processing algorithms have therefore been widely explored over the past decade to increase the precision and the robustness of GED under various walking conditions. GED algorithms use different strategies, such as biomechanical models [9], empirical filtering and signal processing methods [5,10–14], time-frequency analysis [15,16], symbol- or statistic-based methods [17,18] and hidden Markov models [19]. Most algorithms need to be designed for acquisition protocols that use only one sensor position on the body to reduce complexity and cost [5,10,12,15,19,20].

The problem with the methods reported thus far is that wearable systems continuously tend to become smaller and more autonomous to allow good performance in long-term ecological settings. To do so, algorithms stored in the device have to be light in memory and in calculation. Moreover, ecological conditions that increase the variability in the data also increase the processing complexity of most current algorithms. For example, biomechanical models or empirical filtering methods have limited usability in ecological conditions because they often require parameter changes when new gait conditions are considered [5,21]. On the other hand, a robust GED algorithm such as [15] showed very high accuracy for many different gait conditions, but uses a time-frequency transform and a two-dimensional Gaussian curve fit in the time-frequency plane to identify HS and TO events, which can be difficult to implement in small portable devices.

Moreover, previous studies are limited by most algorithms being designed to only distinguish between two gait phases by detecting two gait events: the HS and the TO [10,12,13,15,16]. They distinguish the phase when the foot is in contact with the ground (stance phase) from when it moves (swing phase). Some applications need to detect a higher number of gait events inside the stance and the swing phases, as is the case for functional electric stimulation (FES), which used for patients with neurological diseases such as Parkinson's or in post-stroke readaptation [5,6,22]. Additionally, the inner phase of the stance can be a meaningful temporal parameter to identify differences between normal and pathological gaits by observing patients before and after surgical treatments for ankle osteoarthritis [23].

The use of artificial neural networks (ANNs) in gait analysis is currently gaining popularity, especially for human activity recognition (HAR) tasks [24–27], which are increasingly important in human-machine interactions. ANN show good performance when dealing with the large amount of variability from the environment (hardware acquisition system, the walking surface, noise) and the subject (clothing, shoes, carrying items, walking speed, direction, age, gender or physiological conditions) [24]. Some studies that address the use of ANNs for HAR tasks also used GED for gait cycle normalization in order to increase the performance of the recognition [24]. In addition, affordable IMU sensors and force-sensitive resistors (FSR) enable an easier large-scale gait data collection to apply ANN directly for GED [28].

This paper presents a new approach for GED by using a class of ANN algorithms called reservoir computing (RC) [29]. Recently, ANN algorithms have become prominent due to highly accessible digital data [30]. RC specializes in processing temporal signals such as accelerations. It can also be implemented quickly with a fast and systematic training methodology to adjust coefficients using a linear regression. This means that it can be adapted quickly for different ecological conditions. No other algorithms seem to be as suitable to meet the requirements of low computational complexity necessary for hardware implementation as well as robustness over a wide range of ecological conditions.

In this study, we use the numerical implementation of RC called the echo states network (ESN) [31]. Many applications use the ESN algorithm [32], but only a few are

applied to biomechanics, such as gesture recognition [33], muscle drive-in actuation [34] and exoskeleton control [35]. The ESN model is much simpler to implement on hardware devices compared to most of the advanced GED models recently developed. It can be applied to many walking conditions without increasing the algorithm complexity. To the best of our knowledge, the current literature does not address the application of ESN for GED.

As other ANN algorithms, RC models can be seen as black box models that learn how to process an input to give a useful output prediction. The RC model is applied using a supervised training method, where the predictions are adjusted to be as close as possible to a useful target that is hard to compute otherwise. In this study, we exploited the RC to learn the best relationship between example datasets of input and output target pairs, with the inputs being the acceleration signals and the output being binary indicator signals indicating the gait events. This method is separated in two steps: training and testing. The training step adjusts the trainable parameters of the RC model to establish the optimal relationship between the targets and the RC model predictions. We used many examples of input-target pairs based on experimental measurements. The parameters of the RC model are adjusted to minimize the error between the prediction computed by the RC and the target. The testing step evaluates the capability of the RC to correctly predict the target from new and unseen data (testing dataset). This indicates how well the RC will perform in operation. We measured the RC performance by the error between the model predictions and the targets of this new independent dataset.

The novelty of our study is that it demonstrates the benefits of the ESN approach for GED by adjusting the model for the detection of multiple events (in addition to HS and TO) over multiple walking conditions using a fast training methodology. No GED algorithm in the literature had the ability to offer similar adaptability with the low complexity needed for hardware implementation in simple wearable devices. Our objective is to characterize the ESN performance for GED for six types of event in the gait cycle. We considered different aspects impacting its performance: the characteristics of the training data used, the nature of the sensor data (orientations and positions on the body and the ground walking conditions), as well as the detection capability of the different types of events. Multiple training and testing datasets from three different databases are used to understand the impact of each of these aspects. We also compared the performance of the ESN to other algorithms in the literature detecting the HS and the TO gait events.

The methodology presents how to collect, format and process the gait data using a reservoir computer algorithm. Section 2.1 introduces the three databases of experimental gait data used for our experiments. Section 2.2 explains the process to prepare the gait data in a usable format for the ESN training and testing. Section 2.3 presents the ESN algorithm in detail, and Section 2.4 presents a comparative GED algorithm tested in this work. Section 2.5 presents the standard metrics used to evaluate the GED algorithm performance. Section 3 presents the results of the ESN and the comparative algorithms. Finally, Section 4 is a discussion about the advantages and limitations of the ESN, and it establishes a comparison of the ESN performance with the other algorithms from the literature.

## 2. Methods

The evaluation process of the GED performance of the ESN and the Teager–Kaiser energy operator (TKEO) algorithms is presented in the flowchart of Figure 1. In this process, we used experimental data of the acceleration and ground reaction force measured on participants during variable walking tests, as detailed in Section 2.1. We applied preprocessing steps, including identifying gait events from force sensors signals, structuring the data in timeseries and partitioning the data in training set and a testing set, as explained in Section 2.2. We used a supervised training methodology, which implies that the evaluation process of the ESN was separated into two main processing steps: training and testing. As explained in Section 2.3, the training step established the mean gait cycle duration of the gait data ( $\tau$ ) and the output weight matrix (**W**<sub>out</sub>) of the ESN. The testing step of the ESN predicted the gait events using the peaks in its output signal (Y). We used the TKEO GED algorithm [10], presented in Section 2.4, for comparison purpose in the testing step. We established the ESN and the TKEO's performance with the error between the prediction events and the target events, expressed as the positive prediction value (PPV), the true positive rate (TPR), the T1 score and the mean absolute error (MAE) between event times, as explained in Section 2.5.



**Figure 1.** Procedure used to evaluate the performance of the ESN and the TKEO algorithms. Arrow boxes represent calculation steps and square boxes represent data. The experimental data were collected during walking tests as explained in Section 2.1 and preprocessed as explained in Section 2.2. The input timeseries  $U_k$  is acceleration signals recorded on the participant left foot. The output timeseries  $Y'_k$  is the gait event targets identified with ground force sensors and formatted as a multi-channel binary signal. Each channel indicated the event time indices of one gait event class. The timeseries lists of the inputs  $\{U_1, ..., U_{k}, ..., U_{N_{TS}}\}$ , and the outputs  $\{Y'_1, ..., Y'_{k'}, ..., Y'_{N_{TS}}\}$  represent a subset of multiple input and output timeseries for each participant and walking condition considered in the procedure. The timeseries are separated in two groups, one for the training step of the ESN algorithm and one for the testing step of both ESN and TKEO algorithms. The ESN algorithm steps are represented by the red boxes and explained in Section 2.3. The TKEO algorithm steps are represented by the blue boxes and explained in Section 2.4. The green boxes evaluate the error between predicted and target events and establish performance criteria as presented in Section 2.5.

#### 2.1. Data Collection

We use three databases to test the ESN under a large variability of the walking conditions, namely the MAREA, CIRRIS and UDS database, all including gait acceleration and ground force signal timeseries recorded on healthy adults. The MAREA gait database (MAREA: Movement Analysis in Real-world Environments using Accelerometers https://wiki.hh.se/caisr/index.php/Gait\_database, (accessed on 4 February 2021)) is an open-source database of accelerations and prelabelled gait events recorded under various walking and running conditions that has been already used to test various GED algorithms [10,15,20]. We also assembled the CIRRIS and the UDS databases for the purpose of this study, respectively, at the CIRRIS research center and at the Université de Sherbrooke. All participants in the CIRRIS and UDS databases gave written informed consent to participate in the study according to experimental procedures approved by the IRDPQ-CIUSSS-CP research ethical committee (2021–2269) (IRDPQ-CIUSSS-CP: Institut de Réadaptation en Déficience Physique de Québec et Centre Intégré Universitaire de Santé et de Services Sociaux de la Capital Nationale). Because of the different acquisition frequencies in each database, the data were resampled at 200 Hz using polyphase filtering [36]. Each database was established under independent conditions (treadmill, indoor, outdoor, pathology simulation, obstacles). Table 1 gives an overview of all conditions recorded in each database, including ground conditions, number of participants, duration and walking speed. Details of the data collection process for each database are given below.

Database	Walking Condition		# Participants	Duration (min)	Speed (m/s)		]	Гуре and Nun	nber of Times	eries Selected	in Each Subse	et	
						MAR <sub>TW</sub>	MAR <sub>TI</sub>	MAR <sub>IW</sub>	MAR <sub>OW</sub>	MAR <sub>OWR</sub>	MAR <sub>G1</sub>	MAR <sub>G2</sub>	MAR <sub>all</sub>
MAREA	Treadmill walking	TW	11	$\approx 5$	1.1 to 2.2	11					11	11	11
	Treadmill run	TR		$\approx 5$							11	11	11
	Treadmill incline walk	TI		12	SS *		11				11	11	11
	Indoor flat walk	IW	11	3	SS			11			11	11	11
	Indoor flat run	IR		3	SS				-				11
	Outdoor walk	OW	0	3	SS				9	9		9	9
	Outdoor run	OR	9	3	SS					9		9	9
		Total (1	N <sub>TS</sub> **)			11	11	11	9	18	33	62	73
CIRRIS						CIR <sub>N</sub>	CIR <sub>TA</sub>	CIR <sub>TL</sub>	CIR <sub>FF</sub>	CIR <sub>all</sub>			
	Normal walk	Ν		2		10				10			
	Right toe with external angle	TA	2	for each	0.8, 1, 1.2,		10			10			
	Right trunk leaning	TL	2	speed	1.4, 1.6			10		10			
	Right foot flat	FF							10	10			
		Total (N <sub>TS</sub> **)				10	10	10	10	40			
UDS						UDS <sub>T1</sub>	UDS <sub>T2</sub>	UDS <sub>T3</sub>	UDS <sub>T12</sub>	UDS <sub>T23</sub>	UDS <sub>T13</sub>	UDS <sub>all</sub>	
	Normal walk	T1			SS	15			15		15	15	
	walk with obstacles	T2	15	8	SS		15		15	15		15	
	walk around obstacles	T3			SS			14		14	14	14	
		Total (1	N <sub>TS</sub> **)			15	15	14	30	29	29	44	

## Table 1. Database characteristics.

\* SS: self-selected speed; \*\* N<sub>TS</sub>: the number of timeseries in each subset is equal to the cumulative number of participants selected in each walking condition. This gives an indication of the amount of data variability in each subset. The larger N<sub>TS</sub> is, the greater the variability of the gait data is (more patients and more different gait conditions). In the last 8 columns, colored box indicate the walking conditions not considered, and the numbers indicate how many records were considered for each walking conditions.

The MAREA database was collected on 11 subjects under 5 indoor conditions: flat treadmill walking (TW) and running (TR), inclined treadmill (TI), indoor flat walking (IW) and running (IR). The database also covers 9 subjects under 2 outdoor conditions: outdoor walking (OW) and running (OR). Each subject had a 3-axis Shimmer3 accelerometer ( $\pm 8$  g) attached to their waist, left wrist and left and right ankles (Figure 2). Two force-sensitive resistors were also placed under the foot, at the heel and the toes, to acquire foot pressure measurements on the ground.



**Figure 2.** IMU and FSR positioning for the three databases. In CIRRIS tests, the IMU was placed outside the shoes using straps. For MAREA tests, the three IMU were attached with straps at the three positions shown, and the FSR were placed into the shoe soles. In UDS tests, the IMU and the FSR were placed on a sock inside the shoes. The orientations are defined as the antero-posterior (AP), the medio-lateral (ML) and the vertical (V) axis.

## 2.1.2. CIRRIS Database

The CIRRIS database was recorded on two participants during normal walking (N) and while subjects were simulating 3 different gait strategies related to knee osteoarthritis [37–39]: right toe with an external angle (TA), flat strike of the right foot (rFF) and right trunk leaning (rTL). For each randomized condition, both participants walked for 2 min at 5 different speeds (0.8, 1, 1.2, 1.4, 1.6 m/s). A 3-axis Physilog 4 accelerometer ( $\pm 8$  g) was placed on the top of both shoes, localized as shown in Figure 2. Ground reaction forces for each foot were recorded separately using a treadmill with two embedded force plates (Bertec, Columbus, OH, USA, 1000 Hz). All signals were synchronized using a heel impact on the treadmill.

## 2.1.3. UDS Database

The UDS database was recorded on 15 participants in three walking circuits in an indoor gym. The circuits were a 12 m traverse on a gym track. Three different tracks were used in the tests, as shown in Figure 3: the normal (T1), the obstacle (T2) and the walk around (T3) tracks. Each participant was equipped with an Adafruit MMA8451 Triple-Axis Accelerometer ( $\pm 2$  g) placed on the top of the left foot inside the shoe (Figure 2). Force-resistive sensors were placed under the left shoe, under the shoe sole. Records of 2 min of continuous round trip walking without stopping at the end were done for each track and each participant. The data of T3 for one participant was not taken into account due to incorrect positioning of the FSR.



**Figure 3.** Walking tracks of UDS walking tests. The subjects were walking continuously on the full arrow and coming back following the dotted arrow. T1 was a normal straight walk without obstacles. T2 was a walking circuit over small obstacles, including stepping on one 0.3 m wide by 1 m long by 0.2 m high hard platform (O1), stepping over 0.1 m high by 0.3 m square side platforms (O2) and stepping on 0.5 m wide by 1 m long by 4 cm thick exercise mats (O3). The participant stepped on O1 and O3 by putting their feet on it but had to pass over O2 without touching it. T3 was a walking circuit around 0.3 m wide by 1 m long platforms (O1) aligned in a row with 2 m distance between platforms.

#### 2.2. Data Formatting and Preprocessing

## 2.2.1. Target Gait Events Identification

The target gait events (GE) were defined by time indices of different peaks or drops of the foot ground reaction force. These indices were identified with FSR measurements for the MAREA and UDS databases and intrumented treadmill measurements for the CIRRIS database. Six GE classes, the heel strike, (HS), heel push (HP), foot flat (FF), heel off (HO), toe push (TP) and toe off (TO) were defined in the gait cycle, as shown in Figure 4. All GE classes were located in the stance phase, and one GE index was defined in each gait cycle.

Only some of the six gait event classes were identified in each database. For the MAREA database, only HS and TO were identified directly by the authors of the data source [15]. The HS, HP, FF, TP and TO event classes were determined for the CIRRIS database, and the HS, HO, TP and TO event classes were determined for the UDS database. Processing details of the ground reaction force signal to obtain the targets gait event are given in Appendix C.

## 2.2.2. Input and Target Timeseries Definition

The acceleration and gait event targets were recorded separately for each subject and each walking conditions. Different data subsets grouping all the records for one or more types of walking conditions were formed as presented in Table 1. All records in each subset were identified by the index k. For example, in the subset MAR<sub>all</sub>, k = 1 corresponds to the record of the first participant tested in the treadmill walking condition, k = 2 corresponds to the data from the second participant in the same condition, and k = 12 corresponds again to the first participant, but now tested in the treadmill running condition. **U**<sub>k</sub> corresponds to the acceleration signal of length T<sub>k</sub> (with T<sub>k</sub> the number of time points in the recording). **U**<sub>k</sub> may have only one row corresponding to one of the accelerometers. When various axes are considered, **U**<sub>k</sub> has multiple rows, each corresponding to an individual axis.

 $Y'_k$  corresponds to the gait event targets signal of length  $T_k$  and is a binary signal, as shown in Figure 1. A value of 1 is inserted in  $Y'_k$  at the time index of each target gait event identified from ground force mesurements, and otherwise,  $Y'_k$  is set to 0.  $Y'_k$  has multiple rows, each corresponding to one of the GE classes (HS, HP, FF, HO, TP, TO). For example, in MAR<sub>all</sub>, since only the HS and the TO were detected,  $Y'_k$  had two rows. The first row had a value of 1 only at each HS event index identified, and the second row had a value of 1 only at each TO event index identified; other entries were 0.



**Figure 4.** Event target identification process from foot pressure measurements for each three databases. Examples of foot pressure measurements, acceleration measurements, and target gait events identified in two gait cycles for each database are shown in subfigures (**b**–**d**). The heel strike (HS) is the first contact of the foot with the ground. The heel push (HP) is the maximum load response of the foot after the HS. The foot flat (FF) is the minimum load response of the foot in the middle of the stance phase. The heel off (HO) is the end of the heel contact on the ground in the second part of the stance phase. The toe push (TP) is the second maximum load response of the foot in the second part of the support phase. The toe off (TO) is the final contact of the foot on the ground. (**a**) Event identification; (**b**) MAREA database (acceleration of the foot position); (**c**) CIRRIS database; (**d**) UDS database. FSR1 is on the toe and FSR2 is on the heel.

#### 2.2.3. Subset Partitioning of the Timeseries

The training and testing process in Figure 1 was performed iteratively with multiple subset selections of walking conditions, as presented in the rightmost column of Table 1. These subsets defined the two lists of input timeseries  $\{U_1, ..., U_k, ..., U_{N_{TS}}\}$  and target timeseries  $\{Y'_1, ..., Y'_{k'}, ..., Y'_{N_{TS}}\}$ . N<sub>TS</sub> is the total number of timeseries and varies for each subset selection. The results presented in Sections 3.1, 3.3 and 3.5.1 are based on the subsets MAR<sub>all</sub>, CIR<sub>all</sub> and UDS<sub>all</sub>, where all timeseries of each database were used. In Section 3.2, we use subsets selecting a few specific walking conditions to evaluate the influence of the walking condition variability on the performance of the ESN algorithm. We also used specific condition subsets in Section 3.5.2 to compare the ESN with the TKEO and three other algorithms from the literature.

## 2.2.4. Training and Testing Sets Partioning

We partitioned the timeseries of accelerations and gait event targets in two sets by taking 70% of the total number of gait cycles for the training set, while the other was assigned to the testing set. We defined the separation index in the middle of the stance phase (MSP) using the gait event classes in each database. For the MAREA database, the MSP was defined by the means indices between HS and TO. For the UDS and CIRRIS databases, the MSP was defined with the HO and the FF, respectively. We selected the

separation index at the closest MSP to 70% of the length of the whole timeseries. We eliminated the data before the first MSP in training signal to ensure that both training and testing signals started at the same place in the gait cycle (the MSP). This preprocessing ensures the removal of a uniform gait cycle in each timeseries, considered as transient data in the ESN method, as explained in Section 2.3.1.

## 2.3. ESN Procedure

### 2.3.1. Main ESN Structure

We present here the ESN algorithm presented by Lukoševičius [31] applied for GED using recorded body accelerations. The ESN is a neural network build from three layers: the input layer, the hidden layer and the output layer, as shown in Figure 5. The hidden layer is commonly called the reservoir and is composed of N artificial neurons. The ESN processes the input accelerations  $\mathbf{u}(n)$  and returns the predictions  $\mathbf{y}(n)$  for the GE according to



**Figure 5.** ESN algorithm representation considering the HS and TO gait event classes for the output prediction.

$$\tilde{\mathbf{x}}(n) = \tanh(\mathbf{W}_{\text{in}}\mathbf{u}(n) + \mathbf{W}\mathbf{x}(n-1)) \tag{1}$$

$$\mathbf{x}(n) = \alpha \tilde{\mathbf{x}}(n) + (\alpha - 1)\mathbf{x}(n - 1)$$
(2)

$$\mathbf{y}(n) = \mathbf{W}_{\text{out}}\mathbf{x}(n),\tag{3}$$

where  $\mathbf{u}(n)$  is a column vector of size N<sub>in</sub>, and where N<sub>in</sub> corresponds to the number of acceleration channels in the input augmented by 1 to add a bias value. n is the time-step index.  $\mathbf{y}(n)$  is a column vector of size N<sub>out</sub>.  $\mathbf{W}_{out}$  is the output matrix of size N<sub>out</sub> × N, where N is the number of neurons in the reservoir, and Nout is the number of gait event classes predicted. When the model is used only for HS and TO events,  $N_{out} = 2$ .  $N_{out} = 2$ , 4 and 5, respectively, for the events for the MAREA, UDS and CIRRIS databases when all events are used.  $\mathbf{x}(n)$  is a column vector of size N corresponding to the states of the individual artificial neurons in the reservoir. The variable  $\alpha$  is the leaking rate of the reservoir. It can be set between 0 and 1 and is used to adjust the response time of the ESN. The node connection matrix **W** of size N  $\times$  N and the input layer matrix **W**<sub>in</sub> of size  $N \times N_{in}$  are fixed and are initialized from random uniform distributions. A distribution between [-1, 1] is used to set the elements of the matrices W and W<sub>in</sub>. The sparsity of W is adjusted by setting to zero a fraction 1-P of randomly chosen elements, where P is the sparsity. The spectral radius,  $\rho$ , is set by dividing all elements of **W** by its absolute maximum eigenvalue and multiplying it by  $\rho$ . For the input layer matrix  $W_{in}$ , its sparsity is adjusted by setting a fraction of  $1-P_i$  of randomly chosen elements to zero. The input bias scaling of the matrix  $W_{in}$  is adjusted by multiplying its first row by  $S_b$ . The input scaling is adjusted by multiplying all rows but the first of  $W_{in}$  by  $S_i$ . The parameters used to set the ESN are shown in Table 2. We set these parameters to be fixed for all results. They were chosen based on an exploration of the CHARC properties [40], as presented in Appendix A.

Variable Name	Symbol	Value	
Number of nodes	Ν	100	
Leaking rate	α	0.1053	
idden layer spectral radius	ρ	0.7471	
Hidden layer sparsity	P	0.21	
Input layer scaling	$S_i$	2.300	
Input layer sparsity	$\mathbf{P}_i$	0.41	
Input layer bias scaling	$S_{b}$	2.911	

**Table 2.** Echo state network hyperparameters.

Equations (1)–(3) were iterated for n running over all the time-steps of each input timeseries  $\mathbf{U}_k = [\mathbf{u}(1), \mathbf{u}(2), \dots \mathbf{u}(T_k)]$  of size  $N_{in} \times T_k$ , where  $T_k$  is the total number of time-steps for the input timeseries  $U_k$ . We forced the states to zero at the first time-step,  $\mathbf{x}(0) = \mathbf{0}$ . For the indices of the first gait cycle, the reservoir states  $\mathbf{x}(n)$  were different compared to other gait cycles since the initial value was 0. We considered the time indices of the first 1.25 s ( $n \le 250$ ) as a transient phase, and we eliminated them before further processing. We concatenated the states  $\mathbf{x}(n)$  of the reservoir over all the time-steps of one timeseries to form the states matrix  $\chi_k = [\mathbf{x}(251), \mathbf{x}(252), \dots \mathbf{x}(T_k)]$  of size N × (T<sub>k</sub>-250).

 $\gamma$ 

#### 2.3.2. ESN Training

Hidde

Regularization parameter

The training of the ESN fixes the weights of the output matrix W<sub>out</sub> and determines the gait cycle duration  $\tau$ , defined as the mean duration between each HS gait events. As explained in Section 2.2.3, the ESN processes input accelerations from multiple timeseries,  $\{\mathbf{U}_1, ..., \mathbf{U}_k, ..., \mathbf{U}_{N_{TS}}\}$ , each of size  $N_{in} \times T_k$ . Equations (1) and (2) are first used to compute the reservoir state matrices  $\chi_k$  for each input timeseries of the training dataset.

We defined the same number of target matrices (output signal excluding the first 1.25 s),  $\{\mathbf{Y}'_1, ..., \mathbf{Y}'_{k'}, ..., \mathbf{Y}'_{N_{TS}}\}$ , each of size  $N_{out} \times (T_k$ -250). We then computed the  $\mathbf{X}\mathbf{X}^T$  and  $\mathbf{Y}'\mathbf{X}^T$  using the summation of premultiplied state matrices for each individual timeseries as  $\mathbf{X}\mathbf{X}^T = \sum_{k=1}^{N_{\text{TS}}} \chi_k \chi_k^T$  and  $\mathbf{Y}'\mathbf{X}^T = \sum_{k=1}^{N_{\text{TS}}} Y'_k \chi_k^T$ , where  $\chi_k^T$  is the transposition of the matrix  $\chi_k$ . (The matrix **X** of size **N** ×  $T_{tot}$ , is the matrix of all  $\chi_k$  concatenated together, and **Y**, of size  $N_{out} \times T_{tot}$ , is the matrix of all  $Y'_k$  concatenated together, where  $T_{tot}$  is the total number of time-steps over all the timeseries,  $T_{tot} = \sum_{k=1}^{N_{TS}} T_k$ . In practice, we are not computing these matrices **X** and **Y** explicitly, because if all state matrices  $\chi_k$  and  $Y'_k$  are kept in memory in order to build the matrix  $\mathbf{X}$ , this quickly induces memory saturation since many timeseries are considered.) Wout is computed using a ridge regression,

$$\mathbf{W}_{\text{out}} = \mathbf{Y}' \mathbf{X}^T (\mathbf{X} \mathbf{X}^T + \gamma \mathbf{I})^{-1}, \tag{4}$$

to minimize the mean square error between the output target events and the predicted event timeseries. The matrix I N  $\times$  N is the identity matrix, and  $\gamma$  is a regularization parameter used to prevent overfitting. The parameter  $\gamma$  in Equation (4) was adjusted to  $1 \times 10^{-6}$  (Table 2) by trying multiple values on a logarithmic scale and keeping the one giving the best NRMSE according to Equation (5) using the MAR<sub>all</sub>, CIR<sub>all</sub> and UDS<sub>all</sub> as a validation set.

Finally, the mean gait cycle duration,  $\tau$ , was established from the HS gait event target of all the timeseries in the training set. A lower range gait cycle duration for each timeseries,  $\tau_k$ , was computed using the 5th percentile of the time distances separating each HS events indicated in  $Y'_k$ . The value  $\tau$  is computed as the mean of each timeseries gait cycle duration,  $\tau = \frac{1}{N_{TS}} \sum_{k=1}^{N_{TS}} \tau_k$ . The value  $\tau$  was equal to 0.96, 1.17 and 1.19 s per step, respectively, for the MAREA, CIRRIS and UDS databases considering subsets with all the conditions selected (MAR<sub>all</sub>, CIR<sub>all</sub> and UDS<sub>all</sub>). The MAREA database includes more running conditions than CIRRIS and UDS, which explains the lower value of  $\tau$ .

 $1 \times 10^{-6}$ 

#### 2.3.3. ESN Testing

During the testing, we used the Equations (1)–(3) with the  $W_{out}$  learned in the training phase to compute a prediction output  $Y_k$  from the ESN for each input timeseries of the testing dataset.  $Y_k$  was a multichannel matrix of continuous signals with the same size as the gait event target matrix  $Y'_k$ . Each channel in  $Y_k$  was the prediction for one gait event class. We evaluated the error of the ESN by the normalized root mean squared error (NRMSE), defined as

NRMSE<sub>k</sub> = 
$$\sum_{i=1}^{N_{out}} \sqrt{\frac{1}{T} \frac{\sum_{n=1}^{T} (Y_{k,i,n} - Y'_{k,i,n})^2}{\sigma_{k,i}^2}},$$
 (5a)

$$NRMSE = \frac{1}{N_{TS}} \sum_{k=1}^{N_{TS}} NRMSE_k,$$
(5b)

where *k* is the index of the input timeseries, and i and n are the row and column indices, respectively, related to the event type and the time-step in  $Y'_{k,i,n}$  and  $Y_{k,i,n}$ .  $\sigma_{k,i}$  is the standard deviation of the *i*<sup>th</sup> column target signal  $Y'_{k,i}$ .

We finally used a peak finder algorithm [41] to obtain the time indices of the peaks of each prediction channel signal,  $Y_{k,i}$ . The time indices of the peaks in each channel corresponded to the predicted time of the gait event according to each gait event class. We used a threshold of  $0.65\tau$  to limit the minimum allowed time between two peaks from a given event class in order to minimize the misdetection of events, as explained in Appendix B.

#### 2.4. Comparative Procedure

One of the objectives of this work was to establish a performance comparison between an ESN and other IMU GED algorithms. The Teager–Kaiser energy operator (TKEO) method proposed by Flood et al. [10] uses an energy operator to determine the two energy peaks associated with the initial contact and the final contact of the foot on the ground, referred here as the HS and the TO gait events. This method has demonstrated good performance on the MAREA walking database for the mean absolute error metric (MAE) [10].

Similar to the ESN, the TKEO method has the advantage of minimizing the number of assumptions about gait (e.g., foot symmetry, regularity) to process events. In this paper, the TKEO method was implemented following the indications of [10].

The TKEO method consists of a two-step nonlinear filtering process executed individually on each timeseries  $\mathbf{U}_k$  of the test set. The first step computes a timeseries  $\Phi_k$  of size  $T_k$ representing the amount of energy at each time point of the acceleration  $\mathbf{U}_k$ , expressed in terms of the function  $F_{TKEO1}$  [10],

$$\Phi_k = F_{TKEO1}(\mathbf{U}_k),\tag{6}$$

where  $\mathbf{U}_k$  is a single channel of acceleration. The HS events are detected from the indices of the peaks in  $\Phi_k$  using a peak finder algorithm [41] with a time detection threshold defined as 0.7 of the first autocorrelation peak of  $\Phi_k$  [10]. As a second step, a function  $F_{TKEO2}$  [10] computes a modified energy function  $\Psi_k$ , also of size  $T_k$ ,

$$\Psi_k = F_{TKEO2}(\Phi_k, \{i_{HS}\}_k),\tag{7}$$

based on the energy  $\Phi_k$  and the HS event indices  $\{i_{HS}\}_k$  in order to eliminate the peaks previously detected for the HS in  $\Phi_k$  and to smooth the signal. The TO events are detected from the indices of peaks in  $\Psi_k$  using the same peak finder algorithm used for the HS events and with a time detection threshold equal to 0.7 of the first autocorrelation peak of  $\Psi_k$ . We tested our implementation of this algorithm with the same reference datasets as in [10]. As shown in Figure 6, there is a good agreement. In order to compare the precision of the ESN, TKEO and other models from the literature, we removed the median bias from the errors in all results for all methods. (A median bias in the overall prediction can appear for any type of GED algorithm and can be easily removed in applications without changing the GED algorithm performance.) This reduced the MAE obtained by the TKEO compared to the value reported in the literature [10].

Small variations in the standard deviation were observed compared to the value reported by Flood [10]. This could be related to our implementation in the Python programming language, where the peak finder algorithm [41] was different compared to the one used in the original MATLAB implementation [42]. Since the values of the MAE obtained were mostly the same compared to the reference, our implementation was considered as a valid comparisons algorithm of the TKEO reported by [10] tested here on all our databases.



**Figure 6.** Average (dots) and  $\pm 1$  standard deviation (error bars) of the MAE for the values reported in reference [10] and our own implementation of the TKEO algorithm, under conditions of treadmill walking (TW), treadmill incline (TI) and indoor walking (IW).

### 2.5. Performance Criteria

The metrics of GED performance are based on the error between the time positions of the target and the predicted events. For each timeseries k and class of event i, the vector of the error values  $\mathbf{E}_{ik}$  (each of size  $1 \times N_{TE}(i,k)$ ) was calculated by taking the time of each target event, minus the time of the closest prediction event.  $N_{TE}(i,k)$  is the number of target events in the channel of the target vector  $Y'_{k'}$  and  $N_{PE}(i,k)$  is the number of predicted gait events detected from the GED. The two can be different for a gait event class i and timeseries k.

As proposed in other works on GED [15,20], we removed the constant bias in the error before assessing the performance metrics for each class of events by subtracting the median of the error over all timeseries. Using the absolute values of the vector  $\mathbf{E}_{ik}$ , we classified the target and prediction events into three detection outcomes following the rules presented in Table 3. A threshold of 65 ms was chosen to consider the target events correctly detected, similarly to [10]. For each class of event i, the number of true positives (TP), false positives (FP) and false negatives (FN) were cumulated for all timeseries *k*. We express results with standard metrics of the detection rate used in GED, the true positive rate or sensitivity (TPR),

$$TPR_i = \frac{N_i^{TP}}{N_i^{TP} + N_i^{FP}}$$
(8)

and the positive prediction value or precision (PPV),

$$PPV_i = \frac{N_i^{TP}}{N_i^{TP} + N_i^{FN}}.$$
(9)

These metrics are generally anti-correlated: oversensitive algorithms will have a low TPR but a high PPV, and vice versa for undersensitive algorithms. The  $T_1$  score represents the harmonic mean of the TPR and PPV,

$$T_1 = \frac{2 \operatorname{TPR} \operatorname{PPV}}{\operatorname{TPR} + \operatorname{PPV}},\tag{10}$$

where a  $T_1$  result of 1 represents perfect detection without any FP and FN. Another metric that is used in the literature is called the specificity (SPF),

$$SPF = \frac{N_i^{TN}}{N_i^{TN} + N_i^{FP}},\tag{11}$$

which measures the oversensitivity of an algorithm. Similar the TPR, the value of the SPF will be low if the algorithm is oversensitive. This metric is used to compare the ESN with other algorithms from the literature in Section 3.5.2. However, the SPF is not used to express the results in the other parts of the paper because its value is influenced by the ratio between the number of gait cycles and the total number of points in the timeseries. The walking speed and the sampling frequency will then influence this proportion and the SPF result, even if the algorithms have the same range of errors. Comparing different walking conditions or other works in the literature is then less robust.

Table 3. Classification of the target events (TE) and the prediction events (PE) for FN, TP and FP.

<b>1. False Negative (FN)</b> Target event not detected	<b>2. True Positive (TP)</b> Target event correctly detected	<b>3. False Positive (FP)</b> Prediction event incorrect					
TE with a time distance higher than 65 ms from the closest PE.	TE with a time distance equal or lower than 65 ms from the closest PE.	PE with a time distance with the closest TE higher than 65 ms or higher than any time distance of other PE and the TE.					
$N_i^{FN} = \# \{  \mathbf{E}_{ikl}  > 65 \text{ ms} \  1 \le k \le N_{\text{TS}}, 1 \le l \le N_{TE}(i,k) \}$	$N_i^{TP}$ = # {   <b>E</b> <sub>ikl</sub>   <65 ms  1 ≤ k ≤ N <sub>TS</sub> , 1 ≤ l ≤ N <sub>TE</sub> (i,k) }	$N_i^{FP}$ = # { $N_{PE}(i,k) \mid 1 \le k \le N_{TS}$ } – $N_i^{TP}$					
<b>4. True Negative (TN)</b> All remaining points							
The number of remaining time points in the timeseries unselected as a TP, FP or FN event $N_i^{TN} = \#\{N_T(k) 1 \le k \le N_{\text{TS}}\} - (N_i^{TP} + N_i^{FP} + N_i^{FN})$ $N_T(k) \text{ is the total number of time points in each timeseries}$							

The GED algorithm's performance was also measured using the mean absolute error (MAE). For each timeseries k and event class i, the mean absolute error MAE<sub>*ik*</sub>, in seconds, is calculated as the mean of the absolute of the error vector  $\mathbf{E}_{ik}$  (already recentered for each event by the median error over all timeseries). The MAE<sub>*ik*</sub> results are grouped by gait event class, and the distributions over all timeseries are displayed in whisker plots.

## 3. Results

We present results in five sections. In Section 3.1, we focus on the influence of the training data (training length, choice of the database in training versus testing) used to evaluate the performance of the ESN. In Section 3.2, we evaluate the influence of the walking conditions on HS and TO events using the antero-posterior (AP) acceleration of the foot. In Section 3.3, we focus on the choice of axis: AP, medio-lateral (ML) and vertical (V), as well as various sensor positions (foot, wirst and waist), as the input signal of the model. In Section 3.4, we compare the performance between all the event classes. Finally, in Section 3.5, we compare the ESN performance with other GED algorithms in the literature on the MAREA database based on the prediction of the HS and TO events.

## 3.1. Training Data

This section evaluates the influence of two aspects of the training data on the ESN performance. The NRMSE and the MAE results are presented as a function of the amount of training data in Section 3.1.1. The MAE (mean and standard deviation) for each dataset is presented for various training and testing sets in Section 3.1.2.

## 3.1.1. Training Data Length

Figure 7 shows the NRMSE and MAE as a function of the length of the training data. All timeseries in the testing set were kept with a fixed length equal to the last 30% part of the

full recorded data, as explained in Section 2.2.4. The MAE is presented with the 50th and 75th percentiles of all the MAE<sub>*ik*</sub> of all gait event classes *i* and timeseries *k*. Looking at the 75th percentile of the MAE, the performance of the ESN converges for a total training time of approximately 2000 s. The training for the three databases combined (ALL) did not need more time to converge than each individual database. The NRMSE for the three databases combined is  $0.980 \pm 0.002$  after convergence. Considering that the average cadence in the different databases is about 0.9 steps/s, 2000 gait cycles is a good standard to train the ESN to obtain the best precision for GED. We observed both the NRMSE and the MAE as a convergence criterion to establish the minimum training size, but the MAE metric is more representative of the ESN performance for GED.

In addition, we see in Figure 7 that the NRMSE keeps decreasing with longer training even if performance in terms of MAE stays approximately the same. In many machine learning applications, the NRMSE is a gold standard to evaluate the training convergence. Here, the NRMSE continues to decrease with training time much longer than 2000 cycles before obtaining a stable value. However, the NRMSE expressed the error of all points between the prediction signal and the target binary signal. The positions in time and the sharp shape of the peaks in the prediction signal can influence the NRMSE. Results show that even if the NRMSE decreased with higher training length over 2000 gait cycles of training data, the ESN did not perform better for GED in terms of MAE. The MAE calculation considers only the time position of each detected peak. Even if the noise of the prediction signal might decrease with more training data, the MAE would stay the same if the positions of these peaks stayed the same. Thus, the MAE is a better indicator of the performance than the NRMSE for GED.



**Figure 7.** NRMSE and MAE as a function of the length of the training set (with the AP axis acceleration input) considering all events for each database. The errors were evaluated using the  $MAR_{all}$ ,  $CIR_{all}$  and  $UDS_{all}$  subsets. The three subsets were considered together in the set ALL, with the HS and TO gait event types only. The MAE and NRMSE are different for all databases, mostly because of the harder and higher number of gait event types detected in the CIRRIS and UDS databases than in the MAREA.

## 3.1.2. Training between Datasets

We show the median and interquartile range of the MAE for different training and testing sets in Figure 8. To allow the use of the ESN with training data timeseries from a different database, the predictions considered only two classes of events identified for all three databases, the HS and TO. All timeseries in each database were considered. The results on the diagonals of the matrices in Figure 8 correspond to the case where the training set and the testing sets were coming from the same database. The off-diagonal results correspond to the cases where the training and the testing sets came from different databases (the first 70% of the timeseries from one database was used for the training set and the last 30% of the timeseries from another database was used for the testing set). The first three rows correspond to the cases where only one of the three databases was

used in the training and the last row (or the last column) corresponds to the case where all databases were used for the training (or the testing).

As expected, the best results were obtained when the training and testing data came from the same population. We obtain intermediate results when the training was conducted using all databases. The ESN performance was not quite as good as when the training and testing database matched exactly, but its error remained under a value of MAE of 20 ms. High errors were obtained when the training was carried out using a different database than the one used for testing.





#### 3.2. Performance under Various Walking Conditions

Figure 9 shows that the ESN algorithm is robust enough to perform similarly under all of the walking conditions tested in this work. The prediction results for variable walking conditions are shown for the HS and TO gait events, common to the three databases. Each category along the horizontal axis represents the result for a specific subset of walking condition considered in the train and test sets, as presented in Table 1.

These categories showed that the MAE distributions increased slightly when more walking conditions were included in the subsets. In the MAREA database, the prediction performance was mostly equivalent between indoor conditions (MAR<sub>TW</sub>, MAR<sub>TR</sub>, MAR<sub>IW</sub>, and MAR<sub>G1</sub>) and outdoor conditions (MAR<sub>OW</sub> and MAR<sub>OR</sub>). Mixed conditions (MAR<sub>G2</sub> and MAR<sub>all</sub>) obtained higher MAE due to the larger variability of the walking speed of the participants. The MAE distribution of the TO event class in the subsets CIR<sub>all</sub> was also slightly higher compared to the MAE of other subsets in the CIRRIS database. The results in the UDS database were stable over all the subsets. Overall, the detection performance of the ESN kept a MAE mostly under 30 ms for the HS and under 60 ms for the TO with the variability of all the walking conditions on each database.

Only the condition of indoor flat running (IR) increased the MAE distribution slightly over 110 ms, as shown in the leftmost boxplot of Figure 9a. The average of the MAE distribution was also significantly above its median. This was because the bias prediction errors of these few timeseries under the IR condition was really different compared to the others. Recentering the error of all the timeseries with a uniform bias equal to the median error of all the timeseries (see Section 2.5) causes a high error for these few timeseries were much larger, increasing the mean MEA, but did not affect the rest distributions over all timeseries. Similar cases of higher mean value of the MAE distribution were also observed in the UDS database. In all cases, we see on the right-side axis that the values of the TPV and the PPV were very close from each other, which means that the number of FP and FN was similar. This confirms that the choice of the time threshold of 0.65  $\tau$  was adequate.

120

100

60

40

20

0

120

100

80 60

40

20

TO MAE (ms)

HS MAE (ms) 80

120 100 T1 scores (ms) 80 HS MAE 60 0.4 ۾ 40 0.2 Ĕ 20 С 0.0 120 1.0 100 0.8 g TO MAE (ms) 80 0.6 II 60 0.4 ≩ 40 0.2 8 20 0.0 UDST2 UDST UDS<sub>72</sub> UDS<sub>73</sub> UDS712 UDS<sub>713</sub> UDSa conditi subset Walk (c)

Figure 9. MAE, FPR, PPV and T1 score of the ESN as a function of the subsets of walking condition. (a) MAREA database; (b) CIRRIS database; (c) UDS database. The mean value of MEA is expressed as a green triangle and the median as a central line, while boxes are the 25th and 75th percentiles and the whiskers are the extensions of the last value of  $MAE_{ik}$  of the distribution over and under 1.5 times the interquartile range. The values of TPR, PPV and  $T_1$  are plotted with three lines and are relative to the right-side axis. (Same for all boxplot).

#### 3.3. Performance between Orientations and Positions

This section presents the MAE, TPR, PPV and T1 results for all the event classes. Figures 10–12 are boxplots of the MAE grouped individually by event class. Each category on the horizontal axis represents a different choice of axes for the accelerations used as the input between the AP, ML and V orientations (as shown in Figure 2). The first three categories were defined using one orientation of the acceleration ( $N_{in} = 2$ ). The next three categories denoted with two axes were defined using two orientations of acceleration  $(N_{in} = 3)$ , and the last category was defined with the three orientations  $(N_{in} = 4)$ . The selected acceleration axes were the same for both the training part and the testing part.





**Figure 10.** MAREA database results of MAE, TPV, PPV and T1 performance of the ESN for GED on subset MAR<sub>all</sub>. The boxplots are related to the left axis and represent the MAE distribution over all the timeseries of the testing set. The lines, relative to the right axis, represent the value of the TPV, PPV and T1 score. The horizontal axis categories are relative to different input axis combinaisons. Each vertically aligned subplot represents the results for one event class. The three subfigures represent the result from different IMU positions. (a) Foot sensor position; (b) wrist sensor position; (c) waist sensor position.



**Figure 11.** Same as Figure 10 for the CIRRIS database. The ESN for GED on subset CIR<sub>all</sub> with foot IMU position.



**Figure 12.** Same as Figure 10 for the UDS database. The ESN for GED on subset  $UDS_{all}$  with foot IMU position.

We observed variable performance of the ESN according to the sensor localisation. The results in Figure 10a were obtained with an input defined by the acceleration recorded on the foot. For the results in Figure 10b,c, the input was defined with the accelerations recorded on the waist and the wrist, respectively, and gave significantly poorer results than the foot positions. The median MAE when the foot acceleration was used as input was between 17 and 54 ms, and the T1 score was mostly over 0.75, compared to approximately 170 and 230 ms and 0.3 when the waist or the wrist accelerations were used as the input. These results show that the ESN GED has much higher performance with foot acceleration signals than upper body accelerations.

We also observed small differences in GED performance according to the choice of axes for the accelerations used as the input. The categories using two or three axes did not seem to offer a significant improvement in performance compared to the categories using a single axis. The best result obtained using only one axis was the AP, and the input categories with several channels which included the AP axis were also more efficient.

#### 3.4. Performance According to Various Gait Event Classes

The performance of the ESN varies significantly over the different gait event classes. Observations for each gait event class are listed below: (1) The HS event class obtained the best predictions. The upper whisker of the box plot remained under 30 to 50 ms for the three databases and almost all acceleration axes, and the T1 scores were over 0.92. (2) The HP event class (in the CIRRIS database) also obtained excellent prediction, i.e., with higher whiskers of the MAE values of 50 ms and T1 scores of 0.94. (3) The FF and HO event classes in the CIRRIS database and in the UDS database were predicted with lower accuracy. All acceleration axes for these event classes had an upper whisker of 100 ms (except for the V axis of the HO in the UDS database, which exceeded 200 ms) and a T1 score between 0.6 and 0.8. (4) The TP event class was predicted with variable performance depending on the database. The MAE distributions in the CIRRIS database were quite stable with respect to the different acceleration axes and had an upper whisker of 100 ms and a T1 score of 0.9. In the UDS database, the error results were more variable, with an upper whisker ranging between 70 and 200 ms for various acceleration axes and T1 scores of 0.75. (5) The TO event class prediction performance were also variable according to the database. Results for the MAREA were worst, with MAE distributions up to 120 ms and T1 scores of 0.85. The distributions were clearly better in the CIRRIS database, where the MAE distribution

was under 50 ms and the T1 score was over 0.95. The results in the UDS database were almost as good as for CIRRIS with an upper whisker of the distribution of 60 ms and a T1

score of 0.93. In summary, the event classes at the start of the stance phase, the HS and the HP were better predicted, with a MAE below 50 ms. The event classes in the middle of the stance phase, the FF and the HO, were predicted with less precision, with a MAE of up to 100 ms. The gait events in the end of the stance phase (i.e., TP and TO) were predicted with high variability, the MAE varied from values just below 50 ms to values up to 120 ms.

#### 3.5. Algorithms Comparison

We compared the ESN with four algorithms from the literature. We first compared it with our implementation of the TKEO over the three different databases. The TKEO algorithm had similar characteristics as ESN: it required a low number of assumptions and only a few calculation steps were needed to compute the HS and TO event predictions. A second comparison was made with three other algorithms that required complex calculation steps and gave higher levels of precision compared to most of the results seen in the literature.

These three comparative algorithms have been selected because their results, taken from [15], are based on the same data and the same walking conditions as three of the subsets defined in the MAREA gait database. This allows a fair comparison with the ESN. Moreover, these three algorithms have different characteristics that we want to compare:

- The KH [15] has a good precision. It is one of the best GED algorithms for IMU analysis that provides some of the most precise detection in the literature.
- The Ru [5] can detect more than two conventional gait events (HS and TO) since it was designed for functional electrical stimulation (FES).
- The Au [16] is also a fairly simple algorithm that needs only a few filtering and logical decision steps and is based on a few parameters tuned empirically.

#### 3.5.1. Comparison with TKEO

Figure 13 compares the ESN and the TKEO results of the distribution of MAE for the three databases obtain on all the walking conditions (the subsets MAR<sub>all</sub>, CIR<sub>all</sub>, UDS<sub>all</sub> of Table 1). The input acceleration from the AP axis of the foot, as was proposed in the original TKEO method [10]. The ESN was trained specifically, using training data of one database at a time.



**Figure 13.** Comparison of MEA, FPR, PPV and T1 score on the three databases for the ESN and the TKEO algorithms.

In the MAREA database, the ESN and TKEO algorithms performed in a similar way. For both algorithms, the MAE distribution was under 30 ms for the HS events and mostly under 100 ms for the TO events. A T1 score over 0.92 was obtained for the HS and 0.85 for the TO. On the two other databases, the TKEO algorithm was highly inefficient compared to the ESN. The TKEO had a MAE of up to 500 ms and a T1 score around 0.6 compared to 50 ms and around 0.94 for the ESN. The MAE of the TKEO in the UDS and CIRRIS databases represented an error equivalent to half of the walking cycle.

Figure A3 provides more comparative results of the two algorithms with the other walking conditions subsets presented in Table 1. The ESN algorithm thus showed robust prediction capabilities over all databases compared to TKEO. Figure A3 shows similar comparative results for the two algorithms when specific walking conditions are considered in each of the databases.

### 3.5.2. Comparison with Kh, Ru and Au

Figure 14 shows results taken from the literature [15] for three other algorithms and their comparison with the performance of the TKEO and ESN algorithms. The same walking conditions of the MAREA database were used (the subsets MAR<sub>TW</sub>, MAR<sub>OW</sub>, MAR<sub>OWR</sub> of Table 1). The results for the event classes HS and TO are presented with different walking conditions with the MAE<sub>*i*</sub> distribution and by the TPR and SPF values. The MAE<sub>*ik*</sub> distributions are described by their mean and their standard deviation.



**Figure 14.** Comparison of the TKEO and ESN algorithms to other algorithms in terms of the TPR, the SPF and the mean (central point) and  $\pm 1$  standard deviation (error bar) of the MAE. The subsets MAR<sub>IW</sub>, MAR<sub>OW</sub> and MAR<sub>OWR</sub> of the MAREA database were used, as were the results of the Kh, Au and Ru algorithms from [15]. The performance criteria used here for TKEO and ESN were adjusted to match the criteria used for Kh, Au and Ru. A threshold of 40 ms was used to consider the events as TP. The MAE was evaluated with the error from the TP events only. The number of points in each timeseries was considered for a sampling rate of 128 Hz.

In order to ensure a fair comparison, the results of the MAE given for the TKEO and ESN were evaluated in the same manner as other results from the literature [15], using the error of only correctly detected events with a threshold of 40 ms.

The Kh and Ru algorithms were more precise and less variable than the other algorithms. In the Kh algorithm, the detection was almost perfect with a true positive threshold of 40 ms and a TPR and SPF always around 0.99. For the Ru algorithm, the detection was also almost perfect in the IW and OW subsets, but a lower accuracy was obtained in the OWR subset, where running conditions were considered. The Au algorithm was oversensitive for TO events, considering a low SPF value in the three subsets, meaning that many false positive TO events were detected. The TKEO and the ESN implementations obtained equivalent results. The MAE distributions of the TKEO and the ESN were mostly higher than the three other algorithms (here on the correctly detected events, with errors under 40 ms only). The maximal difference of the mean MAE was about 10 ms larger in the case of the TO in IW subset. They obtained poorer detection performance than the Kh and Ru algorithms in the IW and OW subsets, but they kept approximately the same results in the OWR subset, contrarily to Kh and Ru, showing that they were less influenced by the walking speed variations. The detection rate of the ESN and the TKEO was always acceptable (0.75 of TPR in the worst case) for all conditions, contrary to the Au and the Ru algorithms that showed oversensitivity for TO even in the subset MAR<sub>OWR</sub>.

## 4. Discussion

We discuss here five important aspects of the application of the ESN for GED: (A) Training characteristics of the ESN have been found to be critical; (B) The ESN robustness over various aspects of input data; (C) The ESN robustness over different types of gait event predictions; (D) Simplicity of the ESN compared to other state-of-the-art algorithms; (E) The clinical perspectives and some further investigations needed regarding the ESN algorithm.

#### 4.1. Training Characteristics to Ensure Good Performance

## 4.1.1. Minimum Training Duration

A total of 2000 gait cycles was a good standard to obtain the smallest MAE over three different databases containing at least 40 (for the case of the  $CIR_{all}$ ) different walking conditions and participants, as shown in Figure 7. This means that each participant needs to perform about 50 steps (less than 1 min) of continuous walking in each expected condition to ensure stable performance. If more variability was considered in the training data from other pathologic gaits or with new walking environments, this minimal training duration might need to be increased.

#### 4.1.2. Training for Specific vs. General Applications

Multiple walking conditions, sensor locations and saturation characteristics induce many differences in the amplitude and the signal shapes over the three databases. These differences can significantly increase the prediction error for the ESN. Changing gait data conditions between the training and testing sets was the main issue leading to decreased performance. With specific training data, the ESN precision is maximized for similar testing data. However, we observed in Figure 8 that the mean absolute error can increase by over 10 times if a condition unseen in the training is used later in testing. We also observed, in the last row in the matrices of Figure 8, that the ESN was able to generalize event predictions over all datasets when the training set considered all databases. Generalization, in this application, indicates the possibility to maintain the ESN accuracy for GED even with datasets with high variability and different sources. We can see that high variability in the training data also slightly increases the MAE. In other words, the generalization capability of the ESN is obtained with a small tradeoff in terms of precision. This work showed that both general and specific training data could be advantageous to ensure good GED performance, depending on the application and the training data availability. The general training data need to be sufficiently variable under the multiple conditions that can appear in applications and could ensure the robustness of the ESN under all of these conditions. The specific training data should be sufficiently close to one condition of use (same user, devices, body position and walking conditions) and could ensure higher precision for this specific condition.

# 4.2. Robustness over Various Types of Input Data

## 4.2.1. Walking Conditions

Figure 9 showed that ESN obtained good predictions over all the walking conditions tested for the HS and TO events. The HS event was predicted with equivalent MAE over all

conditions and database. For the TO event, the performance obtained was more variable, as presented in the Section 2.1, and appears to be related to the variability of acceleration profiles during TO events. A peak in the acceleration signal is visible in each gait cycle happening at the TO event, but it takes various forms depending on the database, patient and walking condition. We can see such differences in Figure 4, where the shape of the acceleration signal at the end of the stance phase is different between the UDS, CIRRIS and the MAREA databases. Increasing these variabilities in the shapes seemed to affect the consistency of the prediction performance because, as shown in Figure 9, the performance of the TO event detection clearly degrades with the amount of variability in the walking conditions in the three databases. In the MAREA database, the time series had 11 subjects with 5 walking conditions and 9 subjects with 2 outdoor walking conditions. In the CIRRIS database, only 2 subjects performed 4 simulated walking patterns at 5 continuous and regular speeds on a treadmill. In the UDS database, all 14 subjects walked on 3 different tracks with a self-selected speed level. We can therefore qualitatively see that the variability of walking conditions seems to be smaller in the CIRRIS database, larger for the UDS database and largest for the MAREA database. The performance of the TO predictions (Figures 10a, 11 and 12) followed this level of variability (CIRRIS performs best, followed by UDS, then MAREA).

## 4.2.2. Sensor Position

The input defined by the acceleration of the foot is more relevant for GED then upper body accelerations, as shown in Figure 10b,c. As discussed in the literature [12], the acceleration of the upper body is more disturbed by parasitic movements uncorrelated with the gait cycle. The prediction of a walking event with a wrist or a waist sensor represents a more complex task. We saw similar observation in the literature [10] that highlighted the difference in prediction performance between algorithms exploiting upper body accelerations and those exploiting lower body accelerations.

## 4.2.3. Sensor Orientation

The orientation of the accelerometer also influenced the ESN performance. A uniaxial acceleration along the AP axis located on the foot was the best choice to optimize the precision over all types of events and for the three databases. Defining input with additional axes obtained the equivalent error and did not seem to add significant information extractable for the RC. The AP acceleration is usually the primary acceleration considered in other types of GED algorithms [10,11] and can be considered a highly relevant signal for the gait cycle. Nevertheless, the ESN is robust enough to work for GED with multiple axis orientations and combinations because the MAE results remained the order for all the different axes configurations and combinations of inputs tested.

#### 4.2.4. ESN Adaptability Compared to TKEO

Figure 13 shows that the ESN was much more robust than the TKEO algorithm over all the walking conditions included in the three databases. We observed equivalent predictions from the ESN and the TKEO algorithms on the MAREA database, but on the UDS and the CIRRIS databases, only the ESN was able to make reliable predictions.

The lack of robustness of the TKEO algorithm is explained by the way the algorithm distinguishes the HS from the TO events. As explained in detail in the Appendix E, the prediction of the TKEO depends on the assumption that the acceleration signal has a higher peak at the heel strike than at the toe-out. This assumption is true for the MAREA database (the sensor was placed at the ankle), but not in the two other databases (the sensors were placed on the middle top of the foot).

In addition, the TKEO is not a trainable algorithm, a fundamental difference compared to ESN. Physical hypotheses supporting the TKEO algorithms cannot be changed without making major modifications in the algorithm, and would need recoding a hardware system at a low level. Such changes to allow better detection under new conditions require a complex and time-consuming engineering process and cannot realistically be done for each new walking condition considered. In comparison, the ESN training on few data of each database is a quick process that maintains optimal performance in each database without changing anything to the structure of the algorithm. Additionally, the ESN predicts each type of event independently, thus avoiding misdetections caused by errors from other types of events. It is thus easier to adapt the ESN for various types of acceleration signals.

#### 4.3. Robustness for Multi-Event Detection

#### 4.3.1. Capability Comparison with Other Algorithms

The ESN also offers the distinct advantage of being trainable for a high number of events, which is not the case for most of the other methods. The Kh algorithm [15] and the TKEO [10] are limited to the detection of the HS and TO only. These event types are the most important in practice, used to identity the swing phase and the stand phase, but other clinical purposes can benefit from the detection of a higher number of event types.

The Ru algorithm [5] was created for real-time functional electrical stimulation (FES) applications, where the distinction of four gait phases is needed. This method uses a rulebased state machine to detect a higher number of events inside the gait cycle. Transition criteria between events that follow the interrelated logic of the gait cycle are used to establish four different events and avoid misdetection between the events. The results of [5] showed that the HS is precisely detected, but mid-stance, pre-swing and swing transition phases, similar to the FF, HO and TO events, are harder to detect, with errors extending, respectively, around 50, 100 and 150 ms of the interquartile range. Results obtained with the ESN were as good as Ru or better, as shown in Figures 10–12, where the distribution range of all errors remained between 50 and 120 ms for all the events. This range of error (5–10% of the gait cycle) is less than the error (13% of the gait cycle) tolerated for FES, as proposed in a previous study [43]. This shows the capability of the ESN to detect a high number of events with the required precision for applications such as FES.

Another limitation mentioned in [5] is related to the thresholds and transition delay criteria needed to detect pre-swing events (similar to the HO). These are set empirically with offline processing and could become invalid if variations occurred in the speed of walking. Additional work to establish rules for adaptation of the model could be avoided by the systematic training-based method of the ESN.

#### 4.3.2. Expected Performance According to Event Classes

We demonstrated that the ESN can predict up to six gait event classes (HS, HP, FF, HO, TP and TO) but with different performance for each of them, as discussed in Section 3.3. The maximal value of the MAE over for all walking conditions is between 30 ms for the easiest event to detect (HS) and 120 ms for the hardest events (FF, HO). We looked at the shape of the walking acceleration signals, as illustrated in Figure 4, to interpret this dispersion in the predictive performance. Gait events that happen at the beginning of the stance phase (HS and HP) are much more precisely detected. These events occur when the foot encounters the ground. The resulting impact creates the largest peak in acceleration through the walking cycle. These events are then easily distinguishable by selecting the highest acceleration peak of the gait cycle. Events happening in the middle of the stance phase (FF and HO) were less precisely detected. They appear at the center of the stance phase when the foot is static on the ground and no features in the acceleration can be captured. Their prediction is therefore a more complex challenge for an ESN, as we observed in our results.

We found that TO events, occurring at the end of the gait cycle, obtain variable performance. As discussed in Section 4.2.1, the foot movement is important during this part of the gait cycle, creating a peak in the acceleration, like the HS. However, this peak is more variable in shape and in amplitude depending on the walking condition and database inducing variability in the performance.

Finally, we observed two different level of precision in the result of the TP event for the two databases tested. A lower error was obtained for the CIRRIS database (60 ms MAE) and a higher one was obtained for the UDS database (120 ms MAE). We interpreted that this difference might be caused by the difference of the ground truth defined y for each database. The ground truth of TP events in the CIRRIS database was defined by the second peak of ground reaction force sensed by the treadmill. The ground truth of TP in the UDS database was defined by the peaks of FSR1 measurements positioned under the toe surface, as shown in Figure 2. These two peaks did not occur exactly in the same part of the stance phase, as shown in Figure 4d. In the CIRRIS database, the TP occurred in the middle of the stance phase, when the foot was still immobile on the ground and the acceleration was still zero, such as the FF event class. This event was predicted with similar performance as the FF event class. For the UDS database, the TP occurred later in the stand phase, at the start of the swing movement, when the acceleration signal was non-zero, and its prediction was much better rather like the TO.

According to these interpretations, we can conclude that the events close to prominent features of the accelerations are easier to predict. The events located in the gait cycle where the acceleration signal is low or more variable from one cycle to another are more difficult to detect.

#### 4.4. Simplicity Compared to Other Methods

One of the main advantages of the ESN is its simplicity without significant compromise on accuracy. The three other algorithms presented in Figure 14 were based on multistep filtering and data processing methods. The Kh algorithm from Khandewal [15] uses wavelet transforms to individually detect the nominal cadences, the HS and the TO walking events within separate parts of the time-frequency domain. The search space of event peaks is continuously adjusted according to the nominal cadence, giving robustness and precision for high cadence variability. Small delays are necessary to apply a Gaussian fit over a complete gait cycle to identify peaks without noise perturbations. The Au algorithm [16] exploits a process based on a continuous wavelet transform followed by a locality preserving projection [44] to reduce data dimensionality. The detection of the events is made by a training methodology using a probabilistic classification based on a Gaussian mixture model. The Ru algorithm [5] is based on a multi-step filtering process to locate peaks and a rule-based state machine to maintain the logical order of detection between events. The Ru algorithm is developed by iterative empirical process trials.

For the ESN model, most of the computation complexity is to optimize the hyperparameters (HP) (Appendix A) and formatting the training data (Appendix C) of the ESN. These tasks can be performed before the training or operation phases. The HP is usually set by computing many times both the training and the testing steps of the ESN using one specific set of HP at each time. Optimization methods, such as Bayesian optimization [45], can be used to converge iteratively to an optimal set of HP and can take multiple hours of computation. Once the HP is set, the computation cost to apply the ESN is very low. The training can be done offline and takes less than a second for a 100 node reservoir. The calculation steps of the ESN are even simpler in operation mode. Once the training is done, only the Equations (1)–(3) are used, with a complexity of  $O(N^2 + 14N)$  operations for N the number of nodes in the reservoir (see Appendix F). An implementation of the ESN in a Feather Huzza ESP32 [46] could complete these calculations at 400 Hz. The memory needed to implement the matrices used in the reservoir, **W**<sub>in</sub>, **W** and **W**<sub>out</sub>, is also low. In the case of N = 100 nodes, it can be as low as 1 Kbytes. Finally, a low-complexity conventional peak finder algorithm is also needed to identify peaks from the prediction of the ESN prediction. The integration of this algorithm into portable hardware systems is therefore fairly straightforward.

Under normal walking conditions indoors and outdoors, the ESN is less performant regarding the mean and standard deviation of the MAE compared to these 3 other algorithms. However, the ESN seems to be more robust over all conditions since it does not

have any oversensitivity problems for the TO detection, as the Au and Ru algorithms do. The ESN also give similar errors even for the conditions with more variability (OWR), contrary to three others that show higher mean results.

#### 4.5. Clinical Perspectives and Further Investigations

The ESN algorithm has several practical advantages for its use in clinical applications. The ESN can process GED in real time with low temporal delay, and the number of event classes predicted does not increase this delay. The ESN also has the advantage of adapting to each participant. Gait variability is a complex topic that can be related to multiple independent metrics [47]. Even for healthy walkers, many people have a unique gait pattern [48]. Gait variability and stride regularity can be affected by the age and the walking speed of the subject [49]. Since the training of the ESN is a quick and systematic procedure, it could therefore be useful to individualize the training by calculating the output weights  $W_{out}$  specific to each participant using data obtained during a clinical phase prior to the use of the ESN GED system. This allows to improve the ESN prediction efficiency for participants who have a very specific walking pattern.

This leads to further investigations needed to test such algorithm in more various types of gait. The results presented here were obtained from healthy participants only. GED performances are yet to be evaluated on patients with various conditions affecting their gait. We plan to apply the ESN for GED in a pathological population. As an example, individuals with knee osteoarthritis present heterogeneous and altered gait patterns that could influence the performance of the ESN model [50–52]. The comparison of the performance between healthy and pathologic groups of participants will allow further evaluation of the ESN's capability and robustness.

Lastly, better positioning of the sensors could also improve the ESN's capability for GED. The positioning of the sensor has been tested for only two positions on the foot and two positions on the upper body, but prediction performance elsewhere on the body remains unknown.

#### 5. Conclusions

To conclude, we show in this work that the ESN is an efficient and robust algorithm to detect gait events using the acceleration signal measured by a single-axis IMU system. The ESN can be used for various types of gait events simultaneously and achieve detection with acceptable errors over various walking conditions. The ESN had a MAE at most 10 ms higher compared to four state-of-the-art algorithms on indoor and outdoor walking conditions. Unlike these algorithms, the application of the ESN detector, after training, requires limited computing resources and embedded memory. The ESN is thus easily adaptable for hardware implementations on small microcontrollers to be used in real-time clinical applications, such as FES or other walking monitoring systems. This confirms the applicability of the ESN for GED and even other gait analysis computations that need to be done in real time in wearable (embedded) systems. This also opens the door to the applications of new relevant technology such as the neuromorphic micro electronic mechanical systems (MEMS) accelerometers [53,54] designed to capture one axis acceleration signal and process it using a physical RC micro-mechanical system. Compared to ESN, the neuromorphic MEMS have the advantages of significantly increased autonomy and decreased size in wearable gait analysis devices. Like ESN, the neuromorphic MEMS are further expected to provide more robust, trainable GED capabilities compared to other popular algorithms.

Author Contributions: Conceptualization, L.C.-P., J.S. and K.T.; methodology, L.C.-P., J.S., H.Y. and K.T.; software, L.C.-P.; validation, L.C.-P.; formal analysis, L.C.-P.; investigation, L.C.-P. and H.Y.; resources, J.S. and K.T.; data curation, L.C.-P. and H.Y.; writing—original draft preparation, L.C.-P.; writing—review and editing, L.C.-P., J.S., K.T. and H.Y.; visualization, L.C.-P.; supervision, K.T. and J.S.; project administration, K.T. and J.S.; funding acquisition, K.T. and J.S. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the New Frontiers in Research Fund, grant number NFRFE-2018-00598.

**Institutional Review Board Statement:** The human study protocol was approved by the IRDPQ-CIUSSS-CP research ethical committee (2021-2269): Institut de Réadaptation en Déficience Physique de Québec et Centre Intégré Universitaire de Santé et de Services Sociaux de la Capital Nationale.

**Informed Consent Statement:** Written informed consent was obtained from all subjects involved in the study.

**Data Availability Statement:** MAREA database was obtained from Siddhartha Khandelwal and are available at https://wiki.hh.se/caisr/index.php/Gait\_database with the permission of Khandelwal. CIRRIS and UDS databases are available on request from the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest.

#### Appendix A. Hyperparameter Optimization

The hyperparameters (HP) for the input and the bias scaling, the spectral radius and the leaking rate were identified in the literature as those having the most influence on the computing performance of the ESN [31]. The other ESN HP were taken into account from preliminary optimization tests and did not show a significant influence on either the CHARC metrics (as explain below) or the GED performance. These first four hyperparameters presented in Table 2 were thus optimized in order to reduce the error of the ESN for GED. A fast RC characterisation method called CHARC, proposed by [55], was used to obtain the best ESN HP. The CHARC method consists of a numerical procedure that estimates the performance of the ESN with metrics that are independent to the GED task. These metrics are much faster to compute with the ESN then the GED performance over all databases.

Three metrics are estimated using CHARC: the memory capacity (MC), the kernel rank (KR) and the generalization rank (GR). The MC is a measure of the ability to regenerate an input timeseries from the past states of the ESN. The KR is a metric for the ability to distinguish between different timeseries: a large value of KR means that the output states of the ESN are linearly independent for many different timeseries. The GR is a metric for measuring the ability to generalize identical inputs: a low value of GR means that the output states of the ESN computed with two different inputs remains almost linearly dependent when the input timeseries used are similar up to a certain noise level.

The CHARC exploration of the ESN parameters was done by comparing CHARC metrics and GED MAE using the protocol proposed by [55]. A medium-sized dataset (30 timeseries of 300 gait cycles) from the CIRRIS database was used due to its early data availability. It can be seen in Figure A1 that the ESN having the highest ratio KR/GR are also the ones with the highest probability to produce an optimal mean absolute error. In this work, the four first HP of Table 2 were then selected in order to maximize the ratio  $\frac{KR}{GR}$ .



**Figure A1.** Relationship between the CHARC metrics and the mean absolute error for all events. Exploration was done on the CIRRIS database with timeseries of 3 conditions: normal, right toe out and right side trunk leaning. Five different walking speeds and both subjects were selected (30 timeseries).

#### **Appendix B. Influence of Peak Detection Threshold**

The temporal threshold used in the peak finder algorithm in the testing part of the ESN procedure, as presented in Figure 1, influences the results of the metrics FPR, PPV and T1 significantly. This parameter sets the minimal distance between peaks detected in the output of the ESN. In Figure A2, the horizontal axis values represents the temporal threshold defined as a proportion of the average period of the gait cycle,  $\tau$ .

The threshold had a direct effect on the number of FP or FN obtained by the algorithm. A low threshold increased the sensitivity of the model and tended to increase the number of FP. This resulted in an over-sensitive algorithm with low PPV results. A high threshold tended to increase the number of FN. The algorithm will then missed many events and quickly increased the MAE and decreased the TPR. As shown in Figure A2, a value of 0.65–0.70  $\tau$  for the temporal threshold ensured balanced sensitivity and precision values and an optimal T1 score. When the threshold value was over 0.8  $\tau$ , the MAE stabilized at a value of approximately 500 ms, close to the half period of a gait cycle, because prediction peaks were missed for half of the gait cycles. All results presented of this paper were obtained with a threshold value of 0.65  $\tau$ .



**Figure A2.** Performance according to the temporal threshold for peak detection (the shaded areas represent the variability over all the datasets and event classes). The lower, middle and upper line of the shaded areas correspond, respectively, to the 10th, 50th and 90th percentiles of the metric.

#### Appendix C. Ground Reaction Force Preprocessing

Many preprocessing steps were used to identify the gait event targets from the ground reaction force, either from FSR signal with the UDS database or the VGRF signals of the treadmill for the CIRRIS database. An explaination of the process on each database is given here.

#### Appendix C.1. MAREA

The MAREA database (https://wiki.hh.se/caisr/index.php/Gait\_database) provides two time index vectors of correct gait events identified for each timeseries [20]. These events were detected with force-sensitive resistors (FSR) placed at both extremities of the foot. The two events were defined as the initial ground contact of the heel (HS) and the final ground contact of the toe (TO).

#### Appendix C.2. UDS

Five gait events were identified with the signals of the two FSR under the foot. The signal  $\mathscr{F}_T$  was the measurement of FSR1 placed under the toe, and the signal  $\mathscr{F}_H$  was the measurement of FSR2 placed under the heel, as shown in Figure 2. Both FSR signals were low-pass filtered at 15 Hz. The heel off (HO) and the heel strike (HS) events corresponded to the rising and falling edges of the signal  $\mathscr{F}_H$ . The toe-push (TP) and toe-off (TO) corresponded respectively to the largest peak and the falling edge of the signal  $\mathscr{F}_T$ . The following procedure explains how each event was determined:

- 1. Calculate the mean period  $\tau$  of the gait cycle as the first peak in the auto-correlation function of  $\mathscr{F}_{H}$ .  $\tau$  had a mean value of 1.2 s per step over all timeseries of the UDS database.
- 2. Calculate  $\kappa$ , the moving 98th percentile of the  $\mathcal{F}_H$  signal at every 0.2  $\tau$  indices of the signal over a 0.4  $\tau$  wide window. Interpolate  $\kappa$  for all time indices using the nearest index value, then apply a moving average to  $\kappa$  with a 0.1  $\tau$  wide window.
- 3. Compute  $\tilde{\mathscr{F}}_H$  by setting values that were under  $\kappa$  to zero.
- 4. Compute  $\tilde{\mathscr{F}}_{H}^{c}$  by taking the convolution between  $\tilde{\mathscr{F}}_{H}$  and a ramp of 100 samples between -1 and 1 and normalize the result by its maximal absolute value.
- 5. Estimate the time of the HS and HO events, respectively, as the location of the maximum and the minimum peaks of  $\tilde{\mathscr{F}}_{H}^{c}$  using a peak finder function [41] with a temporal threshold of 0.7  $\tau$  and a height threshold of 0.5.
- 6. To find the event TP and TO, separate the signal  $\mathscr{F}_T$  on individual cycles  $x_l$  using the time index of the HS. Each  $x_l$  corresponds to one gait cycle of the  $\mathscr{F}_T$  signal, and l corresponds to the gait cycle index from the begining.

- 7. For each  $x_l$ , calculate  $x'_l = \frac{x_l P1(x_l)}{P99(x_l)}$ , where  $P1(x_l)$  and  $P99(x_l)$  are the 1st and the 99th percentiles of  $x_l$ . The normalized value of  $\mathscr{F}_T$ , denoted as  $\mathscr{F}_T^n$ , is then defined by the concatenation of all  $x_l$ .
- 8. Compute  $\mathscr{F}_T^{n,c}$ , as in step 4, using  $\mathscr{F}_T^n$  and a 50 sample ramp.
- 9. Estimate the time of the TP and TO events, respectively, as the maximum peaks of  $\mathscr{F}_T{}^n$  and  $\mathscr{F}_T{}^{n,c}$ , as in step 5, using a height threshold of 0.15 to find the  $\mathscr{F}_T{}^n$  peaks and 0.5 to find the  $\mathscr{F}_T{}^{n,c}$  peaks.

A verification loop was performed to ensure that the number of events for each class within all cycles was equal to one. The cycles were then separated on individual cycles between two HS events. For each class of event, the position of each event in the gait cycle  $d_{i,l}$  was determined by the distance between the event index and the first index of its cycle (HS event), where *i* is related to the gait event class and *l* is the index of the gait cycle in the signal. The mean event position in the gait cycle  $\overline{d}_i$  was determined for each event class using the mean value of all event positions  $d_{i,l}$ ,  $\overline{d}_i = \frac{1}{N_{cycle}} \sum_{l=1}^N d_{i,l}$ .  $N_{cycle}$  is the number of gait cycle in the signal evaluated by the number of HS. Then, for each gait cycle and each event classes, the following steps were taken:

- 1. The number of indices of the target event detected was counted.
- 2. If more than one index of gait event were found in the gait cycle, the indices were compared using the distance  $d_{i,l}$  from the beginning of the cycle. The event with the smallest difference from  $\overline{d}$  was kept.
- 3. If no event was found, the true gait event was placed at a distance *d* from the beginning of the cycle.

A last step was performed to connect into a single record the signals of the four individual records made on the same track for each patient. Both acceleration and target gait events of the four recorded signals were joined together at the HO events index. The HO events were chosen for connection because there is almost no amplitude in the acceleration during this part of the cycle, as shown in Figure 4d. This was limiting jumps in the acceleration signal at connection points. To do so, the data from the start to the last index before the first HO event as well as the data from the last HO event to the end were deleted from each signal of acceleration and gait event target. Each group of signals from the same track and with the same patient were then concatenated.

## Appendix C.3. CIRRIS

The gait events were detected with the vertical ground reaction force (VGRF) from the instrumented treadmill. Five types of events were identified from these data using the following procedure:

- 1. HS and TO events were obtained from the position of the maximal and minimal peaks of the convolution between the VGRF and a ramp of 50 points between -1 and 1.
- 2. TP events were the positions of the local maximum peaks in the VGRF before the TO event in a window of 50 ms.
- 3. HP events were the positions of the first local maximum peaks in a window of 25 ms after the HS.
- 4. FF were the positions of the central local minimum peak in a 50 ms window between the HP and the TP events.

The number of gait cycles recorded experimentally in the CIRRIS database was lower, around 120 cycles per patient and per gait condition. A final step was taken to increase the number of walking cycles for training and testing. The acceleration and gait event signals were first divided into individual cycles, separated at the mid-stance indices (MS), corresponding to the FF indices for CIRRIS, as explain in Section 2.2. The acceleration and target gait event signals were then generated by taking 200 gait cycles randomly chosen through all of the individual cycles and were then put together at the MS index. As mentioned in Section 2.2,

the MS event was chosen to connect acceleration because almost no amplitude is observed at this event of the cycle, limiting jumps at connection points in the acceleration.

## Appendix D. All Walking Conditions Results of TKEO and TKEO Algorithms

The Figure A3 shows results of the MAE error prediction of the HS and TO events under all the subsets defined in Table 1.



**Figure A3.** MAE, TPR, PPV and T1 comparison between the ESN and TKEO algorithms for HS and TO events for multiple walking condition subsets. (a) MAREA database; (b) CIRRIS database; (c) CIRRIS database (zoom on ESN results); (d) UDS database; (e) UDS database (zoom on ESN results).

#### Appendix E. Processing Comparison between ESN and TKEO

This section explains in further detail the lack of robusness of the TKEO compared to ESN for other databases than the MAREA. In the first step of the TKEO process, the energy found in the acceleration signal is calculated. The HS events are then discriminated from the TO events, considering that the energy peaks are higher at the HS than the TO. This hypothesis was valid for the MAREA signals, where the sensor is closer to the heel than the toe. For the two other databases, the IMU sensors were located at the middle of the foot. The sensing saturation was also limited to  $\pm 2$  g for the UDS database. Most of the energy peaks measured at the HS and the TO in the walking cycle then had similar amplitudes. For most timeseries in the CIRRIS and UDS databases, the TKEO mixed the HS and TO detections. We illustrate this in Figure A4, showing the resulting detection made by both algorithms for four gait cycles of one of the UDS timeseries. Both algorithms made a good detection in the two first gait cycles. In the last two gait cycles, the TKEO interchanged the HS and TO detections: the first energy signal computed by TKEO,  $\phi(n)$ , was used to estimate the HS peaks. In the last two gait cycles of  $\phi(n)$ , the HS event predictions were selected at the TO because their peaks were slightly bigger than the HS peaks. These errors then induced a misdetection from the second energy signal  $\chi(n)$  by changing values inside  $\phi(n)$  to zeros around HS peaks (Section 2.4). In the second cycle, the TO event was also misdetected because the time distance with the event detected in cycle 3 was smaller than the threshold used in the peak finder algorithm. We observed such errors in the majority of the timeseries of the CIRRIS and UDS databases.

These errors are good examples of how GED algorithms might be sensitive to data variability. To improve the TKEO algorithm, adjustments could be needed to preserve the ordered alternation of peak detections for each event, such as rule-based state machines [5] or alternating forcing detection [15].

In comparison, the ESN algorithm predicts each type of event independently without being affected by the shape of the signal or by misdetection dependencies between events. We demonstrate that, without redesigning the algorithm, the ESN can correctly make GED with all of the walking conditions, even when the sensor position on the foot and the hardware system used is different.



**Figure A4.** Examples of predictions on a timeseries for the UDS database for the ESN and the TKEO. Four gait cycles are represented. U(n) is the AP acceleration. The other subplots show the output signals computed from the TKEO (blue lines),  $\phi(n)$  and  $\chi(n)$ , or the ESN (red lines),  $Y_1(n)$  and  $Y_2(n)$ . The events predicted (dots) were identified from the peak finder algorithm applied on each output signal. All the events were correctly predicted by the ESN. For the TKEO, the HS of cycles 1–2 were correctly predicted. HS and TO events of cycles 3–4 were misdetected (×), and the TO of cycle 2 was not detected.

## Appendix F. Complexity of the ESN

The model calculation presented in Equations (1)–(3) has a complexity that can be calculated by the enumeration of the fundamental calculation steps included in these equations, composed of:

- Three matrix multiplications, W<sub>in</sub>u(n), Wx(n 1), and W<sub>out</sub>x(n) with a complexity, respectively, of O(N<sub>in</sub> × N), O(N<sup>2</sup>) and O(N<sub>out</sub> × N) [56];
- Two scalar multiplications,  $\alpha \tilde{\mathbf{x}}(n)$  and  $(\alpha 1)\mathbf{x}(n 1)$  and two vector additions,  $(\mathbf{W}_{in}\mathbf{u}(n)) + (\mathbf{W}\mathbf{x}(n-\tau)), (\alpha \tilde{\mathbf{x}}(n)) + ((\alpha - 1)\mathbf{x}(n-1))$  each with a complexity of  $O(\mathbf{N})$ ;
- One scalar evaluation of  $tanh(\mathbf{X})$ . The tanh function can be implemented with a table lookup and linear interpolation, requiring O(3N) operations.

The total complexity is thus  $O(N^2 + (N_{in} + N_{out} + 7) \times N)$ . In our case,  $N_{in} = 2$ ,  $N_{out} <= 5$ . This gives a complexity of  $O(N^2 + 14N)$ . The design of the reservoir with N = 100 nodes then gives a complexity of about 11,400 elementary operations per timestep to apply the RC. By implementing the algorithms in a Feather Huzza 32 microcontroller [46] with an ESN of 115 nodes, we have observed that the algorithms can run the calculation loop at over 400 Hz.

Moreover, the memory needed to store the data (numerical coefficients of the matrices  $W_{in}$ , W,  $W_{out}$  for the calculation) is about (N × N<sub>in</sub> + N<sup>2</sup> + N × N<sub>out</sub> + 2N) × 8 *bytes*, which gives approximately 8.5 K*bytes* for N = 100. Moreover, if sparse matrices are used, the memory needed drops to (N × N<sub>in</sub> + (P × N)<sup>2</sup> + N × N<sub>out</sub> + 2N) × 8 *bytes*, which is around 1 K*bytes* or a sparsity of P = 0.21, typical of our ESN.

#### References

- 1. Perry, J. Gait Analysis: Normal and Pathological Function; SLACK: Thorofare, NJ, USA, 1992.
- Dadashi, F.; Mariani, B.; Rochat, S.; Büla, C.; Santos-Eggimann, B.; Aminian, K. Gait and Foot Clearance Parameters Obtained Using Shoe-Worn Inertial Sensors in a Large-Population Sample of Older Adults. *Sensors* 2013, 14, 443–457. [CrossRef] [PubMed]
- Paraschiv-Ionescu, A.; Newman, C.J.; Carcreff, L.; Gerber, C.N.; Armand, S.; Aminian, K. Locomotion and cadence detection using a single trunk-fixed accelerometer: Validity for children with cerebral palsy in daily life-like conditions. *J. Neuro Eng. Rehabil.* 2019, 16, 24. [CrossRef] [PubMed]
- Mariani, B.; Jiménez, M.C.; Vingerhoets, F.J.G.; Aminian, K. On-Shoe Wearable Sensors for Gait and Turning Assessment of Patients with Parkinson's Disease. *IEEE Trans. Biomed. Eng.* 2013, 60, 155–158. [CrossRef] [PubMed]
- Rueterbories, J.; Spaich, E.G.; Andersen, O.K. Gait event detection for use in FES rehabilitation by radial and tangential foot accelerations. *Med. Eng. Phys.* 2014, 36, 502–508. [CrossRef] [PubMed]
- 6. Pool, D.; Valentine, J.; Bear, N.; Donnelly, C.J.; Elliott, C.; Stannage, K. The orthotic and therapeutic effects following daily community applied functional electrical stimulation in children with unilateral spastic cerebral palsy: A randomised controlled trial. *BMC Pediatr.* **2015**, *15*, 154. [CrossRef] [PubMed]
- 7. Sheerin, K.R.; Reid, D.; Taylor, D.; Besier, T.F. The effectiveness of real-time haptic feedback gait retraining for reducing resultant tibial acceleration with runners. *Phys. Ther. Sport* **2020**, *43*, 173–180.[CrossRef]
- Barrios, J.A.; Crossley, K.M.; Davis, I.S. Gait retraining to reduce the knee adduction moment through real-time visual feedback of dynamic knee alignment. J. Biomech. 2010, 43, 2208–2213. [CrossRef]
- Lambrecht, S.; Harutyunyan, A.; Tanghe, K.; Afschrift, M.; De Schutter, J.; Jonkers, I. Real-Time Gait Event Detection Based on Kinematic Data Coupled to a Biomechanical Model. *Sensors* 2017, 17, 671. [CrossRef]
- Flood, M.W.; O'Callaghan, B.P.F.; Lowery, M.M. Gait Event Detection From Accelerometry Using the Teager–Kaiser Energy Operator. *IEEE Trans. Biomed. Eng.* 2020, 67, 658–666. [CrossRef]
- Selles, R.; Formanoy, M.; Bussmann, J.; Janssens, P.; Stam, H. Automated Estimation of Initial and Terminal Contact Timing Using Accelerometers; Development and Validation in Transtibial Amputees and Controls. *IEEE Trans. Neural Syst. Rehabil. Eng.* 2005, 13, 81–88. [CrossRef]
- Fasel, B.; Duc, C.; Dadashi, F.; Bardyn, F.; Savary, M.; Farine, P.A.; Aminian, K. A wrist sensor and algorithm to determine instantaneous walking cadence and speed in daily life walking. *Med. Biol. Eng. Comput.* 2017, 55, 1773–1785. [CrossRef] [PubMed]
- 13. Hanlon, M.; Anderson, R. Real-time gait event detection using wearable sensors. *Gait Posture* 2009, *30*, 523–527. [CrossRef] [PubMed]
- Schuy, J.; Mielke, T.; Steinhausen, M.; Beckerle, P.; Rinderknecht, S. Design & evaluation of a sensor minimal gait phase and situation detection Algorithm of Human Walking. In Proceedings of the 2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids), Seoul, Korea, 3–5 November 2015; pp. 20–25. [CrossRef]

- Khandelwal, S.; Wickstrom, N. Gait Event Detection in Real-World Environment for Long-Term Applications: Incorporating Domain Knowledge Into Time-Frequency Analysis. *IEEE Trans. Neural Syst. Rehabil. Eng.* 2016, 24, 1363–1372. [CrossRef] [PubMed]
- Aung, M.S.H.; Thies, S.B.; Kenney, L.P.J.; Howard, D.; Selles, R.W.; Findlow, A.H.; Goulermas, J.Y. Automated Detection of Instantaneous Gait Events Using Time Frequency Analysis and Manifold Embedding. *IEEE Trans. Neural Syst. Rehabil. Eng.* 2013, 21, 908–916. [CrossRef] [PubMed]
- 17. Sant'Anna, A.; Wickström, N. A Symbol-Based Approach to Gait Analysis From Acceleration Signals: Identification and Detection of Gait Events and a New Measure of Gait Symmetry. *IEEE Trans. Inform. Technol. Biomed.* **2010**, *14*, 1180–1187. [CrossRef]
- 18. Torrealba, R.; Cappelletto, J.; Fermin-León, L.; Grieco, J.; Fernández-López, G. Statistics-based technique for automated detection of gait events from accelerometer signals. *Electron. Lett.* **2010**, *46*, 1483. [CrossRef]
- 19. Mannini, A.; Genovese, V.; Sabatin, A.M. Online Decoding of Hidden Markov Models for Gait Event Detection Using Foot-Mounted Gyroscopes. *IEEE J. Biomed. Health Inform.* **2014**, *18*, 1122–1130. [CrossRef]
- Khandelwal, S.; Wickström, N. Evaluation of the performance of accelerometer-based gait event detection algorithms in different real-world scenarios using the MAREA gait database. *Gait Posture* 2017, *51*, 84–90. [CrossRef]
- Yang, S.; Li, Q. Inertial Sensor-Based Methods in Walking Speed Estimation: A Systematic Review. Sensors 2012, 12, 6102–6116. [CrossRef] [PubMed]
- 22. Kang, G.E.; Frederick, R.; Nunley, B.; Lavery, L.; Dhaher, Y.; Najafi, B.; Cogan, S. The Effect of Implanted Functional Electrical Stimulation on Gait Performance in Stroke Survivors: A Systematic Review. *Sensors* **2021**, *21*, 8323. [CrossRef]
- Mariani, B.; Rouhani, H.; Crevoisier, X.; Aminian, K. Quantitative estimation of foot-flat and stance phase of gait using foot-worn inertial sensors. *Gait Posture* 2013, 37, 229–234. [CrossRef] [PubMed]
- 24. Masood, H.; Farooq, H. Utilizing Spatio Temporal Gait Pattern and Quadratic SVM for Gait Recognition. *Electronics* **2022**, *11*, 2386. [CrossRef]
- Mogan, J.N.; Lee, C.P.; Lim, K.M. Advances in Vision-Based Gait Recognition: From Handcrafted to Deep Learning. Sensors 2022, 22, 5682. [CrossRef] [PubMed]
- Mogan, J.N.; Lee, C.P.; Lim, K.M.; Muthu, K.S. VGG16-MLP: Gait Recognition with Fine-Tuned VGG-16 and Multilayer Perceptron. *Appl. Sci.* 2022, 12, 7639. [CrossRef]
- 27. Al-qaness, M.A.A.; Dahou, A.; Elaziz, M.A.; Helmi, A.M. Multi-ResAtt: Multilevel Residual Network with Attention for Human Activity Recognition Using Wearable Sensors. *IEEE Trans. Ind. Inform.* 2022, *early access*. [CrossRef]
- Prasanth, H.; Caban, M.; Keller, U.; Courtine, G.; Ijspeert, A.; Vallery, H.; von Zitzewitz, J. Wearable Sensor-Based Real-Time Gait Detection: A Systematic Review. Sensors 2021, 21, 2727. [CrossRef]
- Jaeger, H. The "Echo State" Approach to Analysing and Training Recurrent Neural Networks-with an Erratum Note; German National Research Center for Information Technology: St. Augustin, Germany, 2010. Available online: https://www.ai.rug.nl/minds/ uploads/EchoStatesTechRep.pdf (accessed on 28 May 2020).
- 30. Aggarwal, C.C. Neural Networks and Deep Learning: A Textbook; Springer International Publishing: Berlin/Heidelberg, Germany, 2018. [CrossRef]
- Lukoševičius, M. A Practical Guide to Applying Echo State Networks. In *Neural Networks: Tricks of the Trade;* Montavon, G., Orr, G.B., Müller, K.R., Eds.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2012; Volume 7700, pp. 659–686. [CrossRef]
- 32. Sun, C.; Song, M.; Hong, S.; Li, H. A Review of Designs and Applications of Echo State Networks. arXiv 2020, arXiv: 2012.02974.
- Wang, W.; Liang, X.; Assaad, M.; Heidari, H. Wearable Wristworn Gesture Recognition Using Echo State Network. In Proceedings of the 2019 26th IEEE International Conference on Electronics, Circuits and Systems (ICECS), Genoa, Italy, 27–29 November 2019; pp. 875–878. [CrossRef]
- 34. Cao, Y.; Huang, J.; Xiong, C. Single-Layer Learning-Based Predictive Control with Echo State Network for Pneumatic-Muscle-Actuators-Driven Exoskeleton. *IEEE Trans. Cogn. Dev. Syst.* **2021**, *13*, 80–90. [CrossRef]
- 35. Choi, B.; Seo, C.; Lee, S.; Kim, B.; Kim, D. Swing Control of a Lower Extremity Exoskeleton Using Echo State Networks. *IFAC-PapersOnLine* **2017**, *50*, 1328–1333. [CrossRef]
- scipy.signal.resample-poly—SciPy v1.7.0 Manual. Available online: https://docs.scipy.org/doc/scipy/reference/generated/ scipy.signal.resample\_poly.html (accessed on 21 June 2021).
- Qiu, R.; Xu, R.; Wang, D.; Ming, D. The effect of modifying foot progression angle on the knee loading parameters in healthy participants with different static foot postures. *Gait Posture* 2020, *81*, 7–13. [CrossRef]
- Gerbrands, T.; Pisters, M.; Theeven, P.; Verschueren, S.; Vanwanseele, B. Lateral trunk lean and medializing the knee as gait strategies for knee osteoarthritis. *Gait Posture* 2016, *51*, 247–253. [CrossRef] [PubMed]
- 39. Pinto, R.; Birmingham, T.; Leitch, K.; Atkinson, H.; Jones, I.; Giffin, J. Reliability and validity of knee angles and moments in patients with osteoarthritis using a treadmill-based gait analysis system. *Gait Posture* **2020**, *80*, 155–161. [CrossRef] [PubMed]
- 40. Dale, M.; Stepney, S.; Trefzer, M. Designing Computational Substrates using Open-Ended Evolution. In Proceedings of the 2020 Conference on Artificial Life, Online, 13–18 July 2020; pp. 665–667. [CrossRef]
- 41. scipy.signal.find\_peaks—SciPy v1.6.3 Reference Guide. Available online: https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.find\_peaks.html (accessed on 17 June 2021).

- 42. Find Local Maxima—MATLAB Findpeaks. Available online: https://www.mathworks.com/help/signal/ref/findpeaks.html (accessed on 21 June 2021).
- Skelly, M.; Chizeck, H. Real-time gait event detection for paraplegic FES walking. *IEEE Trans. Neural Syst. Rehabil. Eng.* 2001, 9, 59–68. [CrossRef] [PubMed]
- Thrun, S.; Saul, L.K.; Schölkopf, B. Advances in Neural Information Processing Systems 16: Proceedings of the 2003 Conference; MIT Press: Cambridge, MA, USA, 2004.
- Akiba, T.; Sano, S.; Yanase, T.; Ohta, T.; Koyama, M. Optuna: A Next-generation Hyperparameter Optimization Framework. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Anchorage, AK, USA, 4–8 August 2019; pp. 2623–2631. [CrossRef]
- Adafruit HUZZAH32 ESP32 Feather Board. Available online: https://www.adafruit.com/product/3405 (accessed on 24 June 2022).
- Moe-Nilssen, R.; Aaslund, M.K.; Hodt-Billington, C.; Helbostad, J.L. Gait variability measures may represent different constructs. *Gait Posture* 2010, 32, 98–101. [CrossRef] [PubMed]
- Schwesig, R.; Leuchte, S.; Fischer, D.; Ullmann, R.; Kluttig, A. Inertial sensor based reference gait data for healthy subjects. *Gait Posture* 2011, 33, 673–678. [CrossRef]
- 49. Kim, Y.K.; Joo, J.Y.; Jeong, S.H.; Jeon, J.H.; Jung, D.Y. Effects of walking speed and age on the directional stride regularity and gait variability in treadmill walking. *J. Mech. Sci. Technol.* **2016**, *30*, 2899–2906. [CrossRef]
- 50. Kobsar, D.; Charlton, J.M.; Hunt, M.A. Individuals with knee osteoarthritis present increased gait pattern deviations as measured by a knee-specific gait deviation index. *Gait Posture* **2019**, *72*, 82–88. [CrossRef]
- Naili, J.E.; Esbjörnsson, A.C.; Iversen, M.D.; Schwartz, M.H.; Hedström, M.; Häger, C.K.; Broström, E.W. The impact of symptomatic knee osteoarthritis on overall gait pattern deviations and its association with performance-based measures and patient-reported outcomes. *Knee* 2017, 24, 536–546. [CrossRef]
- Perez-Ibarra, J.C.; Williams, H.; Siqueira, A.A.G.; Krebs, H.I. Real-Time Identification of Impaired Gait Phases Using a Single Foot-Mounted Inertial Sensor: Review and Feasibility Study. In Proceedings of the 2018 7th IEEE International Conference on Biomedical Robotics and Biomechatronics (Biorob), Enschede, The Netherlands, 26–29 August 2018; pp. 1157–1162. [CrossRef]
- 53. Dion, G.; Mejaouri, S.; Sylvestre, J. Reservoir computing with a single delay-coupled non-linear mechanical oscillator. *J. Appl. Phys.* **2018**, *124*, 152132. [CrossRef]
- 54. Barazani, B.; Dion, G.; Morissette, J.F.; Beaudoin, L.; Sylvestre, J. Microfabricated Neuroaccelerometer: Integrating Sensing and Reservoir Computing in MEMS. J. Microelectromech. Syst. 2020, 29, 338–347. [CrossRef]
- 55. Dale, M.; Miller, J.F.; Stepney, S.; Trefzer, M.A. A substrate-independent framework to characterize reservoir computers. *Proc. R. Soc. A* 2019, 475, 20180723. [CrossRef] [PubMed]
- 56. Rudra, A. (Dense Structured) Matrix Vector Multiplication; SUNY: Albany, NY, USA, 2020.