

國立清華大學

碩士論文

基於深度學習及穿戴式慣性測量單元
之步態分析

Gait Parameters Analysis Based on
Leg-and-shoe-mounted EcoIMU and Deep Learning

系 所 : 資訊工程學系 碩士班

學號姓名 : 105062574 林伯欣 (Po-Hsin Lin)

指導教授 : 周百祥 教授 (Prof. Pai H. Chou)

中華民國一百零七年七月

摘要

本篇論文提出一個基於慣性測量單元的穿戴式系統，結合一系列方法用於估算步距。系統從安裝在行人的兩腿及鞋子上的三軸陀螺儀和三軸加速度計搜集角速度和加速度。藉由一個基於長短期記憶(LSTM)的神經網路，我們偵測腳步事件發生的時間點，接著使用一個由腿長及關節角度組成的步態模型，以及一個同樣基於長短期記憶神經網路的回歸方法計算步距。

實驗使用六個受試者以及三種不同走路速度的走路資料，用來檢驗提出的方法對於不同使用者及不同走路速度的兼容性。在偵測鞋跟踏地及鞋尖離地事件的實驗中，我們得到 -0.015 秒的平均誤差及 0.046 秒的標準差，證明長短期記憶可以有效的偵測腳步事件發生的時間點。在估算步距的實驗中，使用步態模型及回歸方法同樣得到 $0.22 - 0.3$ 公分的平均誤差及 3.8 公分的標準差。實驗結果顯示提出的步態模型改進了步距估算的準確度，以及在回歸方法中，從步態模型取出特徵有助於長短期記憶神經網路學習到更準確的步距。

Abstract

This thesis proposes a wearable system and a chain of methods for estimating stride lengths from inertial measurement units (IMU). The system collects inertial sensor data from several IMUs mounted on the legs and shoes of the walker, where each IMU provides data in terms of angular velocity from a triaxial gyroscope and acceleration from a triaxial accelerometer. The data are first processed by a Long Short-Term Memory (LSTM)-based method to determine the timing of step events. The raw IMU data and extracted features are also fed to LSTM to construct a regression model for learning stride lengths. A mechanical model that calculate stride lengths by the angles at joints and leg lengths is also proposed.

The experiments consist of a user-dependency test and a walking-speed dependency test. The results show that the proposed step event detector can detect heel-strike and toe-off events with -0.0008 s to 0.015 s mean errors and 0.015 s to 0.046 s precisions. The proposed stride-length estimator, whose performance is measured in terms of mean error \pm precision, achieves -0.3 ± 3.8 cm for the mechanical model and -0.22 ± 3.8 cm for the LSTM model with extracted features. The results also show that using the features extracted from our mechanical model makes the LSTM model learn better compared to the LSTM model using raw IMU data.

Contents

Contents	i
Acknowledgments	v
1 Introduction	1
1.1 Motivation	1
1.2 Contribution	2
1.3 Thesis Organization	2
2 Related Work and Background Theory	3
2.1 Step Events Detection Methods	3
2.1.1 Accelerometer-based Methods	3
2.1.2 Gyroscope-based Methods	4
2.1.3 Comparison Between Gyroscope and Accelerometer	4
2.1.4 Machine-learning Methods	5
2.2 Stride Length Estimation Methods	6
2.2.1 IMU-based Methods	6
2.2.2 Camera-based Methods	7
2.2.3 Other Kinds of Methods	7
2.2.4 Machine-Learning Methods	8
2.3 Background Theory	8
2.3.1 Long Short-Term Memory	8
3 Technical Approach	10
3.1 Step Event Detection	10
3.1.1 Data Preparation	10

3.1.2	LSTM Classification	12
3.1.3	Post-processing	12
3.2	Stride Length Estimation	12
3.2.1	Mechanical Model	12
3.2.2	LSTM Regression	15
4	System Architecture and Implementation	17
4.1	Node Subsystem	17
4.1.1	Sensor Calibration	19
4.2	Host Subsystem	19
4.2.1	Handling BLE Packet Loss	19
5	Evaluation	21
5.1	Data Collection	21
5.1.1	Ultrasonic sensor	22
5.1.2	Force-Sensitive Resistor (FSR)	22
5.2	Step Event Detection	24
5.2.1	User Dependency	24
5.2.2	Walking Speed Dependency	26
5.2.3	Sensor Selection	27
5.2.4	Comparing with related works	27
5.3	Stride Length Estimation	27
5.3.1	Cross-subjects Test	29
5.3.2	Walking Speed Dependency	31
5.3.3	User-dependent Test	32
5.3.4	Performance Comparison Between Mechanical Model and LSTM Model	33
5.3.5	Performance Comparison with Related Works	34
6	Conclusions and Future Work	36
6.1	Conclusions	36
6.2	Future Work	37

List of Figures

2.1	LSTM units	9
3.1	An example of LSTM's input data and the ground truth of TO and HS events	11
3.2	The architecture of network	11
3.3	The mechanical model of a single stride	13
3.4	The angles and sides of a triangle	13
3.5	Joint angles and lengths to be calculated	14
3.6	The architecture of network	16
4.1	System Overview	18
4.2	EcoMini	18
4.3	Axes of the gyroscope and the accelerometer in EcoMini	19
4.4	Data length alignment	20
5.1	The shoes with a ultrasonic sensor and FSRs	23
5.2	The scene of triggering FSRs for HS and TO events	23
5.3	The mean error and precision of different testing subjects	25
5.4	The Z axis of gyroscope data from different subjects	26
5.5	Walking under different speeds	26
5.6	Evaluation of sensor numbers	28
5.7	Mechanical model - Difference Subjects	30
5.8	LSTM with different inputs - Difference Subjects	31
5.9	Mechanical model - Difference Speeds	31
5.10	LSTM with different inputs - Difference Speeds	32
5.11	Testing with different numbers of training data	33
5.12	Average performance of three proposed methods	33

List of Tables

5.1	The performance of related works in gait event detection. unit: second	29
5.2	The performance of related works in stride length estimation	35



Acknowledgments

I would like to thank my advisor Prof. Pai H. Chou for his continuous guidance and encouragement. I would also like to thank all members of the Embedded Platform Lab (EPL) at the National Tsing Hua University for their advice and assistance. Winston, Jay, Thomas, YiChang, ZheTing, and Gene helped me very much in the experiments. Finally, I would like to thank my family and friends for their support and encouragements during my study.



Chapter 1

Introduction

Gait analysis refers to extracting information from the way a person walks. It is enabled by advances in sensors of various modalities and computation efficiency, and it is opening up new applications in not only health and fitness but also localization and tracking. This chapter provides a motivation and problem statement for our proposed contributions to gait analysis for stride length estimation.



1.1 Motivation

Stride length is a critical gait parameter that has been investigated much in recent years for various purposes. For example, in an indoor position system, estimating stride lengths is one part of the processing chain. When combining stride lengths with rotation angles and the starting point, we can calculate the position of subjects. With more accurate stride lengths being calculated, this positioning system can be used in tracking the position of a child in public space or further helping firefighters navigate in an indoor environment where signals from Global Positioning System (GPS) satellites are not available. In addition, stride length can also be used in healthcare. For some gait characteristic related diseases such as Parkinson's, stride length can be helpful for tracking the course of the disease and assisting in medical diagnosis.

As part of determining stride lengths, some techniques first detect step events. Several important applications for step events motivate us to detect the timing of the events. For example, they are used by functional electrical stimulation (FES) to generate muscle contraction on specific step event to help people who are paralyzed to produce functions such as walking and standing [1, 2].

For individuals who suffer from amputation, gait events can be helpful for controlling the intelli-

gent prostheses that use finite-state control strategy. If more stages in a gait cycle can be determined, the intelligent prostheses can behave more precisely [3]. Other applications in the fields of rehabilitation [4] and healthcare [5] have also been discussed much in the past.

1.2 Contribution

We propose a step event detector based on an LSTM classification model. With the data from one shoe-mounted IMU, we precisely detect the heel-strike and toe-off events. For stride length estimation, we present a mechanical model and two LSTM models. We add the motion of shoes into the traditional mechanical model and improve the RMSE by 2.17 cm. The two LSTM models estimate stride lengths using raw data and the features extracted from the mechanical model, respectively. We compare the performance of these two LSTM models and find that using model features leads to better results.

1.3 Thesis Organization

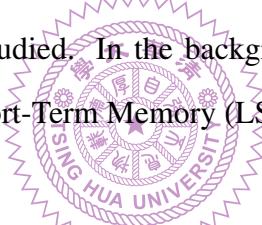


The rest of this thesis is organized as follows. In Chapter 2, we discuss the related works and the background. Chapter 3 describes the detail of the proposed step-event detector and stride length estimator. Chapter 4 introduces how we implement the system. Chapter 5 presents experimental results. Finally, Chapter 6 concludes the thesis with directions for future work.

Chapter 2

Related Work and Background Theory

This chapter reviews related work and background on gait detection and stride length estimation. We divide gait detection algorithms into those based on accelerometers and gyroscopes, and we discuss their differences. For stride length estimation, we divide the algorithms into IMU-based, camera-based, and other sensors-based methods. The uses of machine learning in both gait events detection and stride length estimation are also studied. In the background theory section, we introduce the mechanism and application of Long Short-Term Memory (LSTM).



2.1 Step Events Detection Methods

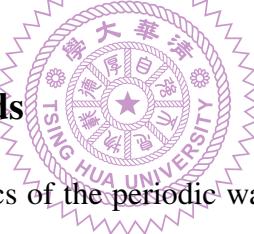
The *timing of step events* refers to the moment when a step event happens, and an example is a “heel-strike” event, which is the moment when the shoe touches the ground. Step-event detection has been done usually by using the data from shoe-mounted or shin-mounted IMUs. The data is first preprocessed to remove noise using signal-process techniques such as low-pass filters. Then, the relationship between the IMU data and the corresponding gait events are found for step event detection, which can be divided into accelerometer-based and gyroscope-based methods.

2.1.1 Accelerometer-based Methods

When using accelerometers to detect step events, researchers find several ways to analyze the waveform of accelerometer data. Khandelwal et al. [6] propose an algorithm based on Continuous Wavelet Transform, which transforms time series into multiple wavelets that can be used for peak detection, to detect heel strike (HS) and toe off (TO) events. They achieve high timing accuracy with a mean

absolute error (MAE) of 0.03 and 0.01 seconds for HS and TO events, respectively. Threshold-based approaches are also simple but efficient ways to detect step events [3, 7]. Selles et al. [7] determine the timing of HS and TO from two accelerometers attached below the knees by applying different thresholds for fast and slow walking. The accelerometer data is first smoothed by a low-pass filter before their relationship with the gait events is found. The mean error (standard deviation) is between 0.013 to 0.034 seconds. [2] use an accelerometer placed on the trunk and determine HS events by finding the negative-positive changes of horizontal acceleration. The time interval between the timing of HS and sign changing is approximately 150 ms. They achieve good timing accuracy with a mean error between 0.024 to 0.086 seconds for different subjects.

These algorithms are developed based on the observation of waveforms and the knowledge about the characteristics of human gaits. They find features in the accelerometer's waveform from different aspects and achieve good accuracy, while their performance may be affected when the variance of subjects' age or walking speed becomes large. In [7, 2], certain rules and hand-optimized parameters are involved, so they may need to switch between parameter settings under different circumstances.



2.1.2 Gyroscope-based Methods

Researchers find that some characteristics of the periodic waveform are helpful for determining the timing of gait events using gyroscope data. In [8], a rule-based algorithm is applied to the signal from a shin-mounted gyroscope. It is observed that two local minima in the sagittal plane indicate the timing of HS and TO. The condition and thresholds applied to these events are determined empirically, and they achieve standard deviations of 0.017 and 0.035 seconds for HS and TO events, respectively. There exist similar algorithms that input data from gyroscopes on shins and detect step events based on the local minima above [9, 10, 11]. The differences between [10] and [11] are their design of Butterworth low-pass filter and the way to search for local minima. Gouwanda et al. [9] further reduce the computational cost for local search by identifying potential points, which are possibly being determined as an HS, TO, or mid-swing (MS) event, using a zero crossing method.

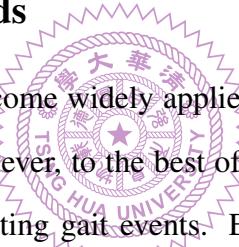
2.1.3 Comparison Between Gyroscope and Accelerometer

Among the algorithms that utilize accelerometers and gyroscopes, we find that gyroscope data usually require less signal preprocessing before it can be used for motion analysis. In contrast, accelerometer

data need more processing to filter out gravity, and linear acceleration can be sensitive to the sensor placements. For example, the measured moving accelerations of a rigid limb such as a shin may differ when a sensor is attached close to the knee or the ankle, while gyroscopes can be attached anywhere on a segment with almost identical signals [12]. In addition, acceleration data often involves high-frequency components that make it noisier than angular rates measured by a gyroscope [13].

Gyroscopes also have their limitations. They are more sensitive to temperature and shock due to the piezoelectric material commonly used in vibrating-structure MEMS gyroscopes [13]. Another problem that needs to be handled for gyroscopes is drift, the long-term noise that causes the gyro data to deviate from its true value over time. Drift is caused by many factors and has complex non-linear characteristics. Many methods have been proposed to compensate for gyroscope drift [14, 15]. A simple way to do this is zeroing the reading of gyroscope when it is stationary so that the error caused by drift will stop accumulating.

2.1.4 Machine-learning Methods



In recent years, machine learning has become widely applied to activity classification, thanks to advances in algorithms and hardware. However, to the best of our knowledge, little research has been published on machine learning for detecting gait events. Edel et al. [16] propose a method using Bidirectional Long Short-Term Memory Recurrent Neural Networks (BLSTM-RNNs), which is able to uncover contextual information, to do step detection. This algorithm outputs a signal when a step is detected, and the class of this step is also determined at the same time. They categorize steps into different classes by walking speeds and directions, and the influence of different placements of IMU is evaluated. Their results show that BLSTM-RNNs are able to improve the robustness in the aspect of IMU placements, and the estimated step count is also improved when compared to other competing algorithms. It should be noted that the timing accuracy of gait events are not evaluated in this research.

In this thesis, we apply a simple RNN on IMU data to obtain the timing of gait events more accurately. One advantage of applying deep learning in this task is that the deep learning model can be adaptive to different walking speeds and subjects, provided that training data set is sufficiently large. Compared to the step event detectors described above, hand optimization of thresholds and rules are not required in the proposed method.

2.2 Stride Length Estimation Methods

For stride length estimation, various kinds of sensors and the corresponding algorithms are proposed. Most of the algorithms described in this section need to split each stride from continuous walking data by some methods, which have been discussed in the previous section, while approaches using cameras may skip this part and calculate the stride length directly by image processing technologies. We categorize the algorithms by the sensor they use and their characteristics, and then we discuss the use of machine learning in estimating stride length.

2.2.1 IMU-based Methods

A typical IMU contains a triaxial accelerometer and a triaxial gyroscope. Many works using IMUs to do stride length estimation have been proposed to date. They can be further divided into two classes based on how they use IMU data.

The first class of methods is based on mechanical models, which utilize natural constraints on human legs. Aminian et al. [13] use wavelet analysis to detect step events and model one stride with two pendulums for the swing phase and one pendulum for the stance phase. They obtain angles at joints by integrating gyroscope data captured at shin and thigh on one leg and then calculate stride length by the length of leg and angles at the joints. Salarian et al. [17] further reduce the number of sensors by predicting thigh angles from shin angles. One limitation of these algorithms is that these models simplify walking as a 2D motion, but in fact the orientation of legs and shoes change slightly on each axis when we are walking. Furthermore, we cannot get the trajectory of legs and shoes, which are important features for the diagnosis of gait characteristic related diseases.

The second class of methods is to calculate stride length by integrating accelerometer data. These algorithms apply a mechanism called zero-velocity update (ZUPT) that manually stops integrating accelerometer data during the stance phase. The reason why ZUPT is needed is that the accelerometer as a low-cost sensor usually outputs noise even when it is not being moved. The noise is amplified after double-integration, which transforms acceleration into velocity and then to displacement. The works in this class [18, 19, 20] all use two foot-mounted IMUs and introduce a similar process chain. First, they maintain the orientation of the sensor by some kind of filters or quaternion calculation. A transformation of accelerometer data from their local frame to the global coordinates is then performed, followed by gravity cancellation. The displacement can then be conducted by integrating

acceleration data twice. In [18], a Kalman filter combined with ZUPT algorithm is proposed. They achieve precisions of 6.1% and 3.7% at a sampling rate of 100 Hz and 200 Hz, respectively. Rebula et al. [19] maintain the trajectory of shoes by estimating the orientation and foot velocity, and then the stride length is computed. The rotational orientation is obtained by integrating gyroscope data and using accelerometer data to correct the tilt base on the reading of gravity, and the velocity is calculated by simply integrating the accelerometer data. They reach a 3.2% RMS error of stride length. Trojaniello et al. [20] reduce the effect of accelerometer drift by applying a kind of high-pass filter, and they obtain less noisy linear velocity in the anterior-posterior axis that is then used to calculate stride lengths.

2.2.2 Camera-based Methods

For methods that use cameras to calculate stride length [21, 22], usually one or two stationary monitors are needed to obtain gait parameters including stride length, frequency, and speed. What they have in common is that they separate targets from the background in a video and then count the number of steps by algorithms that can find repeat pattern of subjects. The stride length is calculated by dividing the distance by step count. Stone et al. [23] further explore stride-to-stride gait parameters from Kinect and a two web-cameras system, separately. These approaches are useful in human identification since no device needs to be installed close to the target. For elderly care, camera-based algorithms are also suitable in environments such as home and hospital where subjects move around within a specific area.

2.2.3 Other Kinds of Methods

Different types of sensors can be used for stride-length estimation. Electromyography (EMG) mounted on the shin [24] can record muscle activities that can be used to infer strides and stride lengths with a linear model. Kinect, which an RGB camera and a depth sensor, is used in [23] for obtaining stride velocity, stride time, and stride length. It is shown that stride-to-stride variation in stride velocity can be obtained more accurately by Kinect compared to the system composed of two webcams.

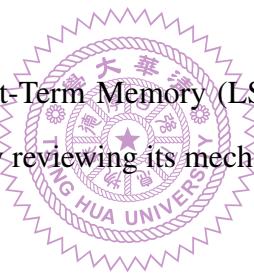
2.2.4 Machine-Learning Methods

Machine learning is relatively new among the algorithms for stride length estimation as discussed in this chapter. Algorithms using machine learning do not need to handle the problems with drift and accumulated error from noise as described above, due to their data-driven characteristic.

To collect data for stride length estimation, multiple smartphones or IMUs are usually attached to the human body and shoes. Edel et al. [16] proposes a method that uses BLSTM-RNNs to separate strides, and then stride lengths are modeled with a linear equation whose input is the variance of the accelerometer data. They achieve good results with a mean error of 1.45% and precision of 1.3% to 3.0% for different sensor placements. Hannink et al. [25] use Convolutional Neural Network (CNN) on accelerometer and gyroscope data as input to estimate stride length, and they achieve an accuracy of 0.01 cm and precision of 5.34 cm.

2.3 Background Theory

In this section, we introduce Long Short-Term Memory (LSTM), the network architecture used to construct our models for gait analysis, by reviewing its mechanism and advantages.



2.3.1 Long Short-Term Memory

LSTM is a unit that can be placed in a Recurrent Neural Network. It is first introduced by Hochreiter et al. [26] in 1997. It has been applied to speech processing, computer vision, and handwriting recognition. The idea of this network architecture is to uncover contextual information of time-series data. Conventional Recurrent Neural Network (RNN) that uses Back-Propagation Through Time (BPTT) to update its weights is also a commonly used network architecture, but it suffers from vanishing and exploding gradients problems. LSTM makes up for these problems by its specially designed architecture that enforces constant error flow and clips gradients at certain points.

Fig. 2.1 shows the arrangement of gates and the memory cell in a single LSTM unit. It inputs time-series data x_t and outputs a_t at time t . The LSTM unit combines x_t and a_{t-1} together and multiplies it by certain weights to generate the inputs Z , Z_i , Z_f , and Z_o to the LSTM unit at time t . Three of them, namely Z_i , Z_f , and Z_o , are used to determine whether to activate the corresponding gates to memorize, forget, and expose its content, respectively. This mechanism enables LSTM to

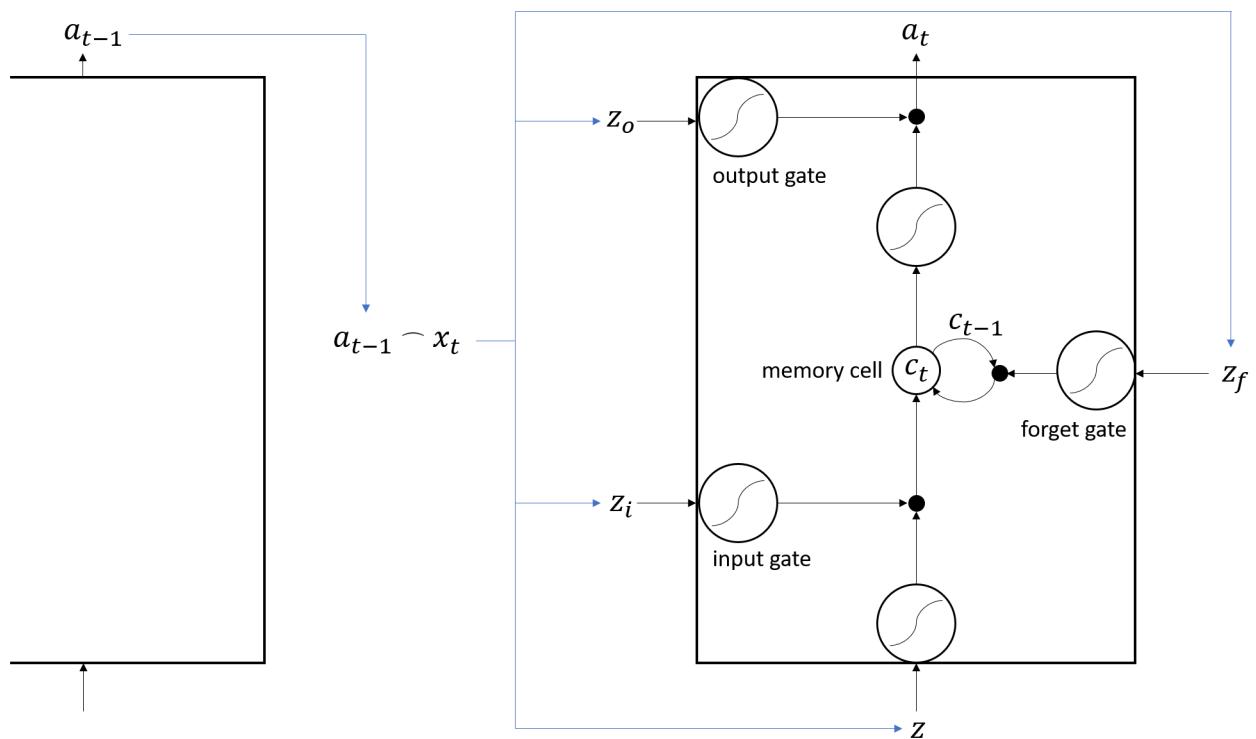
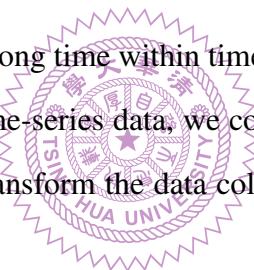


Figure 2.1: LSTM units

carry critical information in sufficiently long time within time series data.

Due to LSTM's ability to analyze time-series data, we consider it to be a suitable network architecture for modeling our tasks, which transform the data collected from IMU sensors to the desired gait events and stride lengths.



Chapter 3

Technical Approach

In this chapter, we describe the methods we apply to obtain gait parameters, including an LSTM model for step event detection and two models for stride length estimation.

3.1 Step Event Detection

To detect the timing of desired gait events, we use an LSTM model with input data obtained from leg-and-shoe-mounted IMUs. The idea is to utilize an LSTM model to help us find the relationship between waveforms of IMU data and the corresponding step events. We choose toe-off (TO) and heel-strike (HS) events as our targets because these two events provide helpful information for determining other gait parameters such as stride length and gait cycle time.

3.1.1 Data Preparation

The data we feed to an LSTM model has been preprocessed by a low-pass filter and normalized to $[0, 1]$ range. A Butterworth filter with ten poles is used to filter out high-frequency components in accelerometer and gyroscope signals. Then, we perform normalization feature by feature. The maximum and minimum of each feature in training data are found, and test data is scaled according to these extreme values. At last, data from five IMUs are aggregated to form a sequence of feature vectors that are ready to input to LSTM models. Fig. 3.1 shows an example of time-series data: from a single IMU: Z-axis of the gyroscope data and X-axis of the accelerometer. The red and blue dashed lines represent the ground truth timings of TO and HS events, respectively.

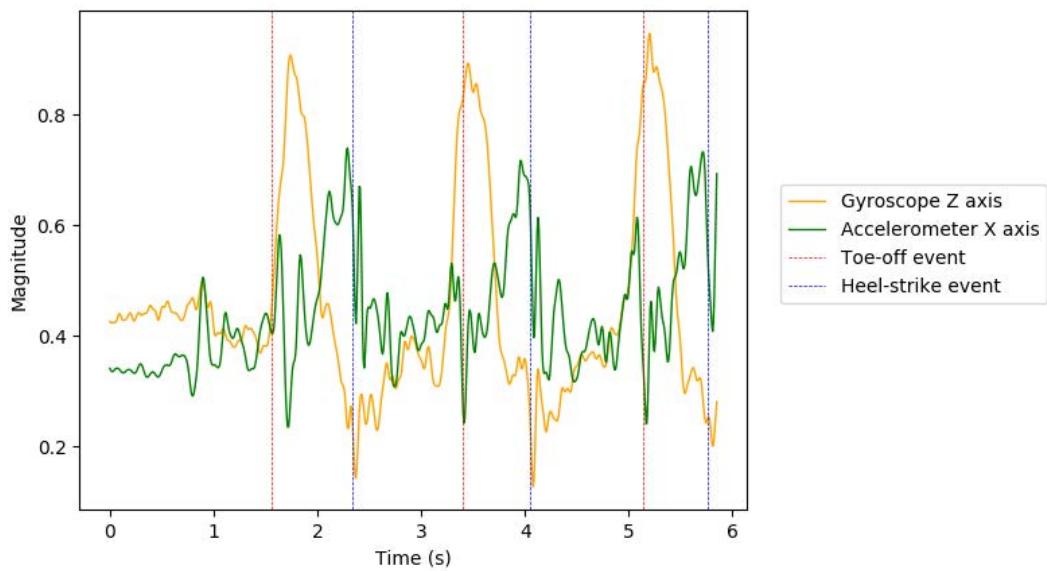


Figure 3.1: An example of LSTM's input data and the ground truth of TO and HS events

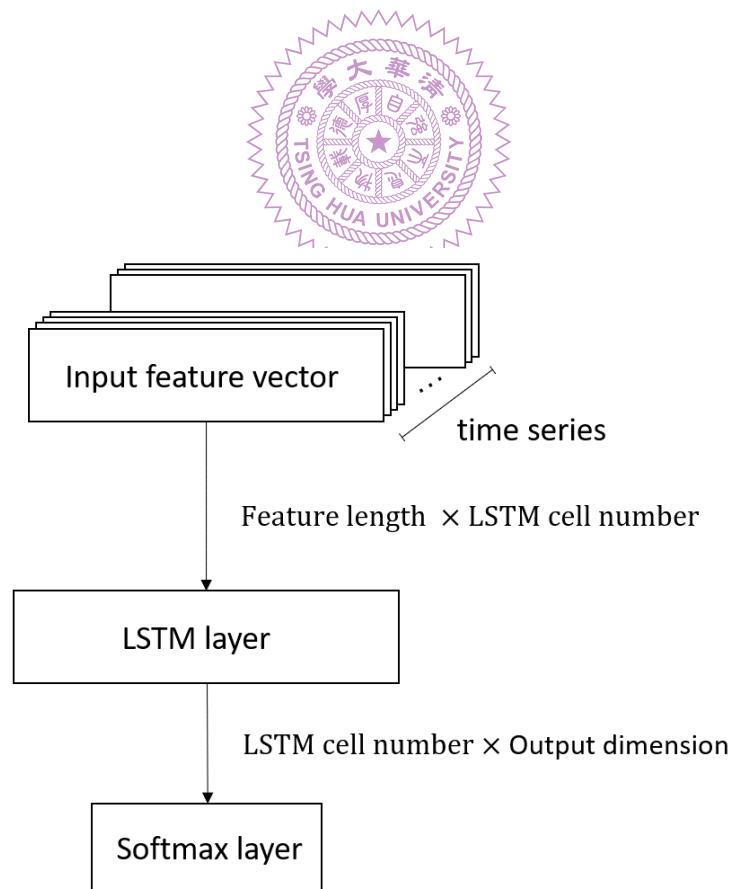


Figure 3.2: The architecture of network

3.1.2 LSTM Classification

We model the task of detecting the timing of gait events as a classification problem. For a sequence of time-series data, we classify the data at each time step as either a “normal point” or an “event point.” An event point can be further divided into a TO or HS point. We use a simple network architecture shown in Fig. 3.2.

An input layer is followed by an LSTM layer that contains a certain number of LSTM units. The LSTM layer is responsible for finding the contextual information and determining the most possible moment the gait event happens. The third layer is a softmax layer that rescales each entry of the classification result to the range $[0, 1]$, and the summation of all entries is one. The output entries represent a probability distribution over the three possible outcomes.

3.1.3 Post-processing

The output of the LSTM classifier is a sequence that represents the classification result at each timestep. Around a step event, there may exist several points all classified as that step event. Therefore, we need some post-processes to find the most possible point that represents the step event among these points. We find local maxima in the output sequence and see the timing of these local maxima as the timing of step events. When the time difference between two local maxima is less than 0.4 seconds, we take the average of their timings as the moment when step event happens.

3.2 Stride Length Estimation

We propose two methods to estimate the stride length. The first is a mechanical model that makes use of the natural constraint of human legs and joints. The other is based on LSTM models that use two kinds of input data to find stride lengths.

3.2.1 Mechanical Model

As depicted in Fig. 3.3, we model a stride by the motion of the legs at TO and HS moments. The stride length is then segmented into D_1 , D_2 , and D_3 . This idea and a similar model are first proposed by Aminian et al [13]. In their work, two gyroscopes are attached to the thigh and shin of one leg. To simulate the motion of walking closer, we attach four IMUs on the thighs and shins of two legs

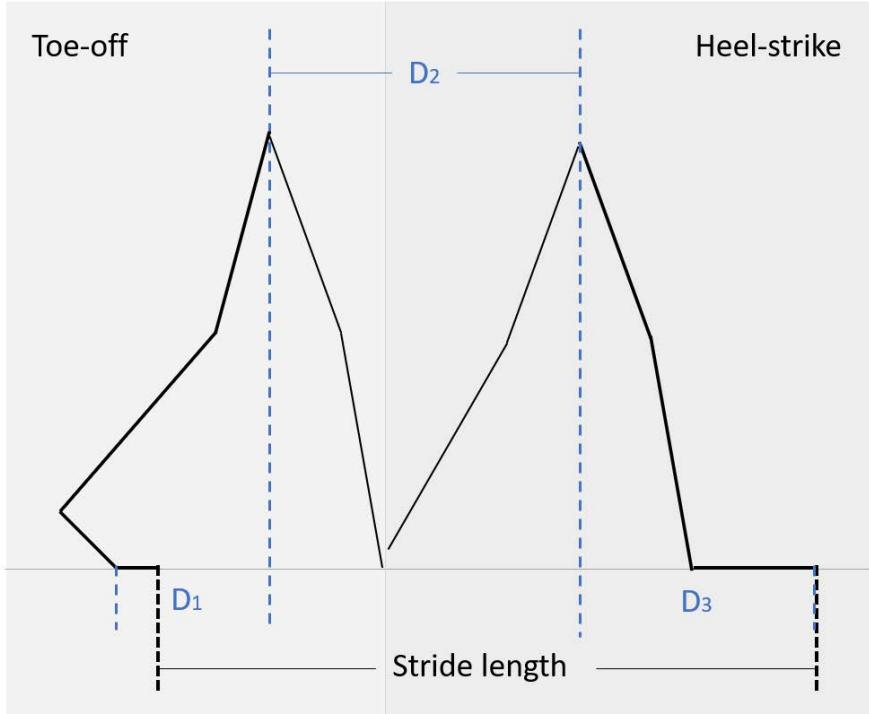


Figure 3.3: The mechanical model of a single stride

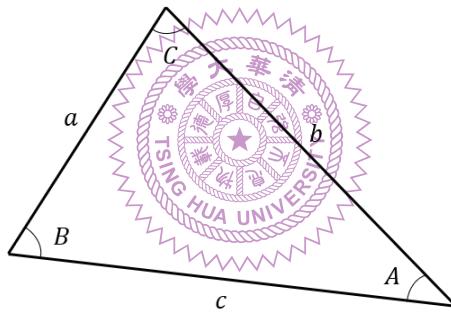


Figure 3.4: The angles and sides of a triangle

and one IMU on the left shoe. Totally five IMUs are used, and we further modify the mechanical model. In conventional double pendulum model, we assume that legs are straight at the moment of HS. Besides, the length of shoes are ignored, but human shoes are lifted up at TO time. Therefore, we add the motion of the shoes into the model instead of adopting the straight-leg assumption.

This model requires the lengths of a subject's thigh, shin, and shoe as its inputs, and the angles at knees and ankles are obtained by integrating the angular rate of the five IMUs. Fig. 3.5 illustrates the required angles and lengths to be found. The calculation of these components are mainly based on Eq. (3.1) and Eq. (3.2). The detailed process is described by Eq. (3.3). At last, the stride length is obtained by Eq. (3.6).

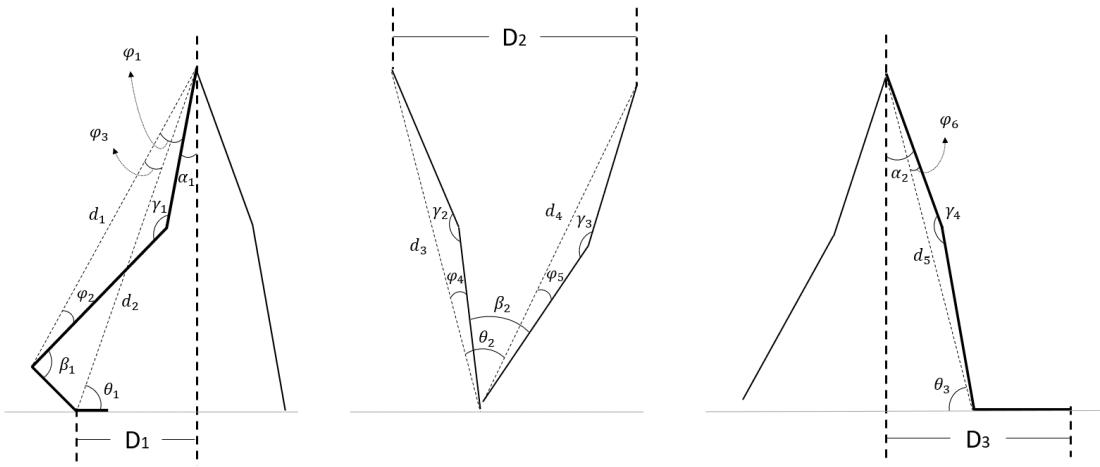


Figure 3.5: Joint angles and lengths to be calculated

$$\text{Law of sines : } \frac{a}{\sin(A)} = \frac{b}{\sin(B)} = \frac{c}{\sin(C)} \quad (3.1)$$

$$\text{Law of cosines : } c^2 = a^2 + b^2 - 2ab \cos(C) \quad (3.2)$$

Known variables:

l_1 = thigh length

l_2 = shank length

l_3 = shoe length – toe length

α_1 = angle of left thigh at TO

α_2 = angle of left thigh at HS

β_1 = angle of left ankle at TO

β_2 = angle of right shin betweewn TO and HS

γ_1 = angle of left knee at TO

γ_2 = angle of right knee at TO

γ_3 = angle of right knee at HS

γ_4 = angle of left knee at HS

Calculation of D1:

$$\begin{aligned}
 \phi_1 &= \arcsin\left(\frac{\sin(\gamma)l_2}{d_1}\right) & d_1 &= \sqrt{l_1^2 + l_2^2 - 2l_1l_2 \cos(\gamma)} \\
 \phi_2 &= 180^\circ - \gamma - \phi_1 & d_2 &= \sqrt{d_1^2 + l_3^2 - 2d_1l_3 \cos(\beta + \phi_2)} \\
 \phi_3 &= \arcsin\left(\frac{\sin(\beta + \phi_2)l_2}{d_1}\right) & \theta_1 &= 90^\circ - ((\phi_1 - \phi_2) + \alpha) \\
 D_1 &= d_2 \cos(\theta_1) & & (3.3)
 \end{aligned}$$

Calculation of D2:

$$\begin{aligned}
 \phi_4 &= \arcsin\left(\frac{\sin(\gamma_2)l_1}{d_3}\right) & d_3 &= \sqrt{l_1^2 + l_2^2 - 2l_1l_2 \cos(\gamma_2)} \\
 \phi_5 &= \arcsin\left(\frac{\sin(\gamma_3)l_1}{d_4}\right) & d_4 &= \sqrt{l_1^2 + l_2^2 - 2l_1l_2 \cos(\gamma_3)} \\
 \theta_2 &= \beta_2 - \phi_5 + \phi_4 & D_2 &= \sqrt{d_2^2 + d_3^2 - 2d_2d_3 \cos(\theta_2)} & (3.4)
 \end{aligned}$$

Calculation of D3:

$$\begin{aligned}
 \phi_4 &= \arcsin\left(\frac{\sin(\gamma_4)l_2}{d_5}\right) & d_5 &= \sqrt{l_1^2 + l_2^2 - 2l_1l_2 \cos(\gamma_4)} \\
 \theta_3 &= 90^\circ - (\alpha_2 - \phi_4) & D_3 &= d_5 \cos(\theta_3) + \text{Shoe length} & (3.5)
 \end{aligned}$$

$$\text{Stride length} = D_1 + D_2 + D_3 - \text{Toe length} \quad (3.6)$$

3.2.2 LSTM Regression

Estimating stride length from IMU data can be seen as a regression problem. We again use LSTM to find the relationship between the time-series data and the target value. We use two kinds of input, including the raw data of these sensors and features extracted from the mechanical model described in Section 3.2.1.

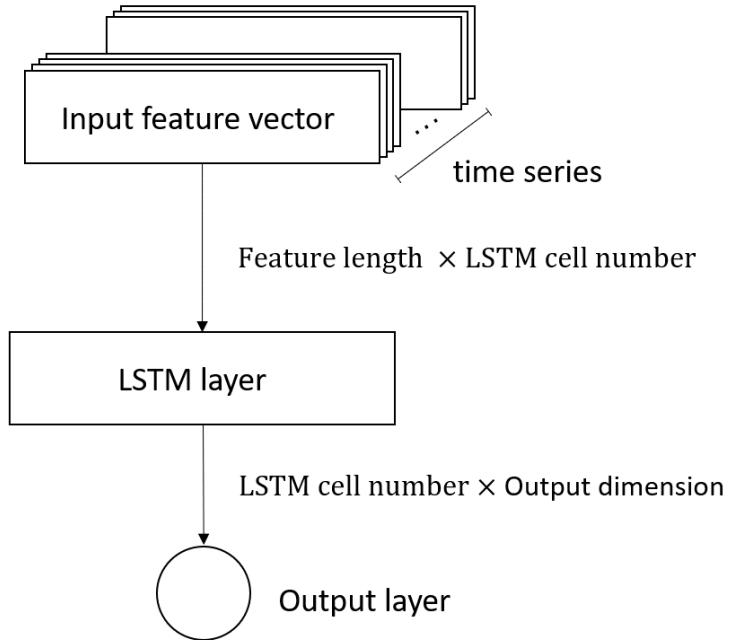


Figure 3.6: The architecture of network

Data Preparation

Continuous walking data is first separated stride-by-stride using TO and HS timings obtained from our gait event detector as described in Section 3.1. The event timings measured by the sensor on the left shoe are used to segment data because we measure stride lengths by the distance traveled by our left leg. We pick data from each TO time to its next HS time, and the data from each HS time to the next TO time is discarded because the left shoe is on the ground and practically not moving during this time. After the segmentation, the data of each stride is then passed through a tenth-order Butterworth filter and normalized into the range [0, 1].

Besides the raw data from the five IMUs, the angles and segments' lengths, which are generated during the process of calculating stride lengths in our mechanical model, are used as input. These features are labeled and shown in Fig. 3.5. We calculate the angles and segments' lengths at each timestep, and then these features become new time-series data that serve as input to LSTM models.

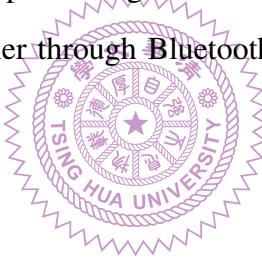
Network Architecture

Fig. 3.6 illustrates the network architecture, which is similar to the one we use to detect gait events. The first two layers are Input and LSTM layers. The difference is that we replace the Softmax layer with a simple output layer that contains a single neuron. By this change, we transfer the classification result to a single value that represents the regression result.

Chapter 4

System Architecture and Implementation

The system architecture of the proposed system can be divided into the node subsystem and the host subsystem. The node subsystem is composed of a number of sensors, which are responsible for collecting 6-DoF data, ultrasonic sensor data, and Force Sensitive Resistor (FSR) data. The host subsystem is a laptop that does data preprocessing and runs the trained LSTM models. These two subsystems communicate with each other through Bluetooth Low Energy (BLE). The overview of our system is shown in Fig. 4.1.



4.1 Node Subsystem

The node subsystem is composed of five sensing nodes attached on both legs and the left shoe. We implement our system on EcoMini, a $9 \times 12 \text{ mm}^2$ motion sensing node developed by the Embedded Platform Lab (EPL) at National Tsing Hua University (NTHU) in Taiwan. EcoMini is based on TI CC2541 microcontroller unit (MCU) that contains an 8051 core and a BLE radio transceiver. A 9 degree-of-freedom (9-DoF) inertial sensor, MPU9250, is also embedded in EcoMini. MPU9250 is composed of a digital triaxial accelerometer, a gyroscope, and a compass. Fig. 4.2(a) shows the front side of EcoMini. For convenience of debugging and powering, we mounted EcoMini on an adapter and use EcoMini with the adapter to implement our system, as shown in Fig. 4.2(b).

The nodes are attached to the outer side of the subjects' two legs by velcro straps. Fig. 4.3 shows the direction of EcoMini's gyroscope and accelerometer. When walking, the subject's legs move along the front-back plane, so the bearing angle of each node can be obtained by integrating the Z-axis of its gyroscope data.

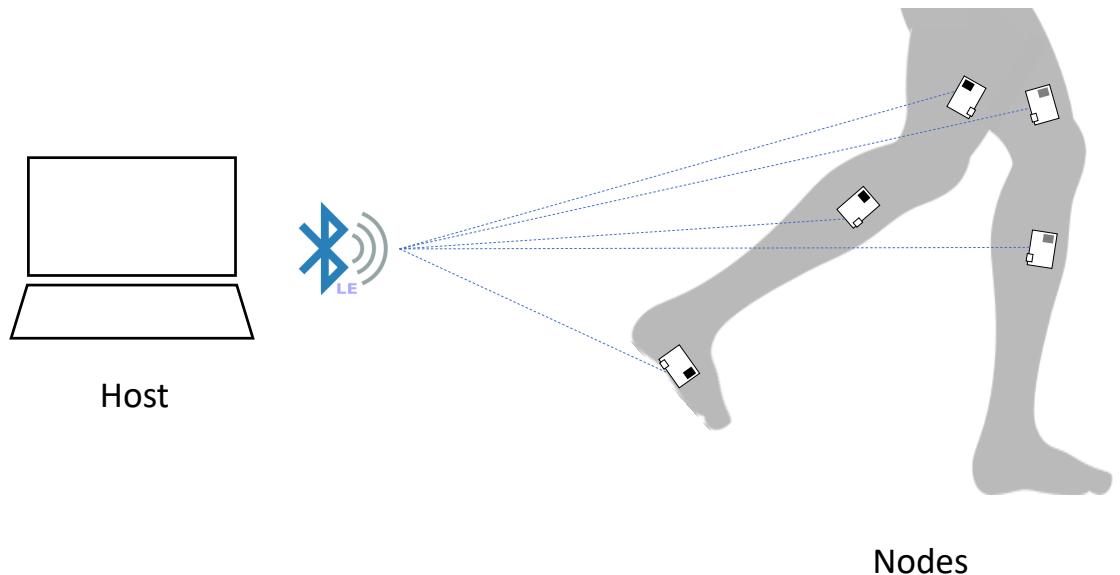


Figure 4.1: System Overview

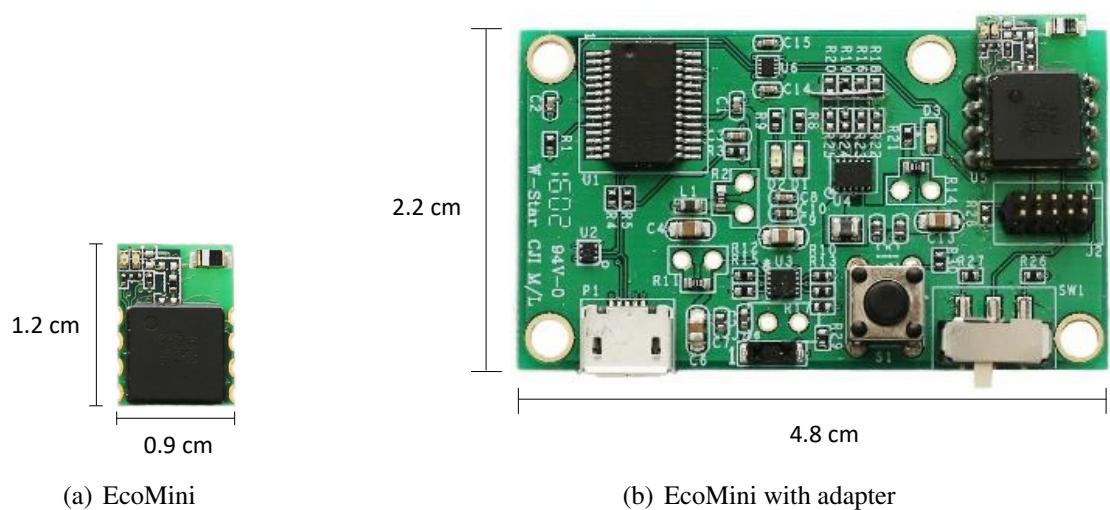
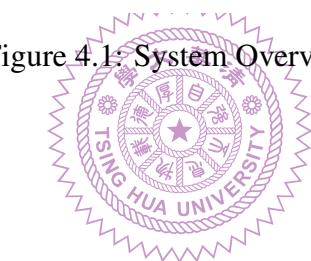


Figure 4.2: EcoMini

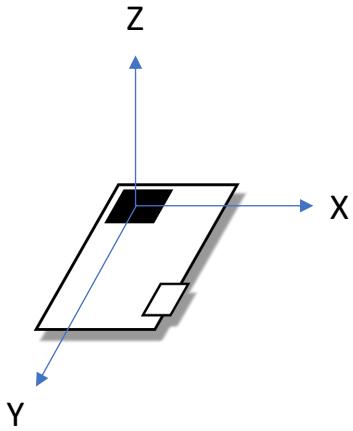


Figure 4.3: Axes of the gyroscope and the accelerometer in EcoMini

4.1.1 Sensor Calibration

Our node can perform calibration upon powering on. We keep it stationary and calculate the average value of the first 3000 data samples in each axis. These average values are considered biases that are later subtracted from the readings of the corresponding axes.

4.2 Host Subsystem



The host is a laptop with a Bluetooth dongle that builds connections with the nodes and receives notifications from them. After collecting data, the host does a sequence of pre-processing and then estimates gait parameters through the proposed mechanical model and the trained LSTM models.

4.2.1 Handling BLE Packet Loss

The data from EcoMini is transmitted to the host through BLE’s *notification* mechanism, where a slave “pushes” data to its master once the data in a characteristic being changed. However, a slave does not know if a notification packet is successfully received by the master. In contrast, regular paired BLE communication is fully acknowledged.

Due to the problem of packet loss, the host does not necessarily receive the same number of packets from all five EcoMinis in the same time duration. On the host side, we record the timestamp when receiving each packet. These timestamps are called receiving timestamp. To align the packets from the five nodes, we use the receiving timestamps from an additional node that provides the ground truth of step events. Fig. 4.4 shows an example of data length alignment. Between two successive

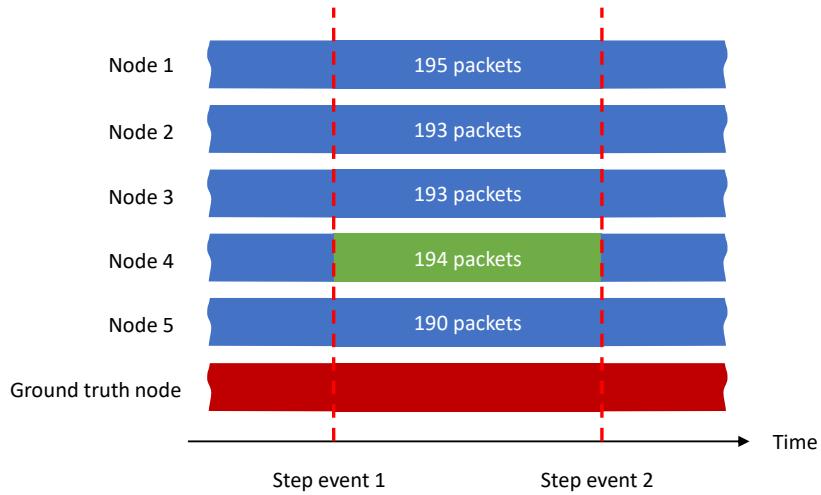


Figure 4.4: Data length alignment

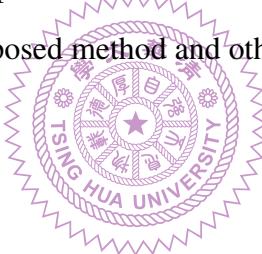
step events, the number of received packets from each node can be different. We find the minimum received packet number, which is marked green, among all nodes, and then downsample the data from each node using this minimum. Afterward, we can obtain the aligned data by aggregating the downsampled data.



Chapter 5

Evaluation

This chapter explains our method for performance evaluation. We first describe the collection of walking data and ground truth. For our gait event detector and stride length estimator, we examine their user dependency and their stabilities toward different walking speeds. The estimation results of the LSTM models using two different input combinations are also compared. At last, we compare the accuracy and precision between the proposed method and other works.



5.1 Data Collection

The walking data used in this work is collected from six volunteers. Four sensors are attached to the shins and thighs of their two legs. Two additional sensors are attached to the left shoe. Among the six sensors attached to subjects, five of them collect data from their IMU sensors and send it to the host through BLE at a data rate of 125 samples per second. Each sample contains 12 bytes of IMU data. The sixth sensor, which is attached to the left shoe, is responsible for collecting ground truth from the FSRs and the ultrasonic sensor. It uses a faster data rate of 200 samples per second to ensure that the host can get FSR signals as soon as possible. Each sample contains one byte of data that represents whether the FSRs have been triggered. Totally, 369 strides are collected, including 213 strides from six subjects with their comfortable walking speed and 156 strides from a single subject with fast and slow walking speeds.

5.1.1 Ultrasonic sensor

To correctly obtain the distance of each stride, we use a shoe-mounted ultrasonic sensor, as shown in Fig. 5.1(a). In the past, two kinds of systems are often introduced to obtain stride parameters. One type is motion-capture systems based on cameras and markers that can capture 3D motion with high spatial and temporal resolution, but they are relatively costly and need a dedicated location of installation. The other sensing modality is walkways with pressure sensors that provide 2D stride parameters while able to handle higher mobility than stationary camera systems can. These two systems can efficiently and conveniently provide ground truth data once they are set up. Comparing to these two systems, ultrasonic sensors that we use are low-cost and still provide comparable accuracy to walkways. We can measure distance by a shoe-mounted ultrasonic sensor as long as there exists a wall or a barrier that can reflect sound waves, while it requires in the front to generate reflection.

In our experiments, we use HC-SR04 that provides accuracy of ± 3 mm to measure the distance of each stride. HC-SR04 can be used to measure distance by sending out ultrasonic pulses and receiving the echo. One interrupt is generated when HC-SR04 emits the sound and another interrupt on detecting the echo, allowing the MCU to do the time measurement. Our EcoMini measures the time between these two interrupts and calculates the distance by Eq. (5.1). The speed of sound in air is changing under different temperatures, as shown in Eq. (5.2). The term T represents the temperature in Celsius. We assume that the temperature is 25°C in our experiments.

$$distance = \frac{velocity\ of\ sound \times time}{2} \quad (5.1)$$

$$velocity\ of\ sound = 331.5 + 0.607 \times T \ (m/s) \quad (5.2)$$

5.1.2 Force-Sensitive Resistor (FSR)

To collect the ground truth for when shoe hits and leaves the ground, which are also known as heel-strike and toe-off events, we attach force sensitive resistors (FSR) under the shoes. Two FSRs are installed in the toe-end and heel-end of the left shoe, respectively, as shown in Fig. 5.1(b). FSR is a resistor that initially has a large resistance without force. Once some force is applied to it, its resistance drops from larger than 100 k Ω to hundreds of Ω depending on the magnitude of the force. We connect the FSR to an interrupt pin of EcoMini, which sends the corresponding data to the host system.



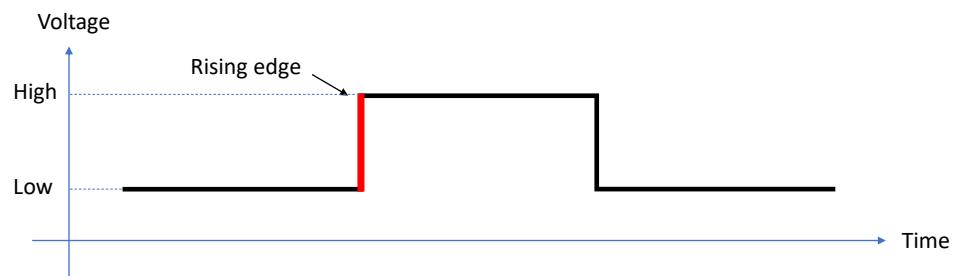
(a) Ultrasonic sensor

(b) Two FSRs under the left shoe

Figure 5.1: The shoes with a ultrasonic sensor and FSRs



The trigger of heel-strike event (use a pull-down resistor) :



The trigger of toe-off event (use a pull-up resistor) :

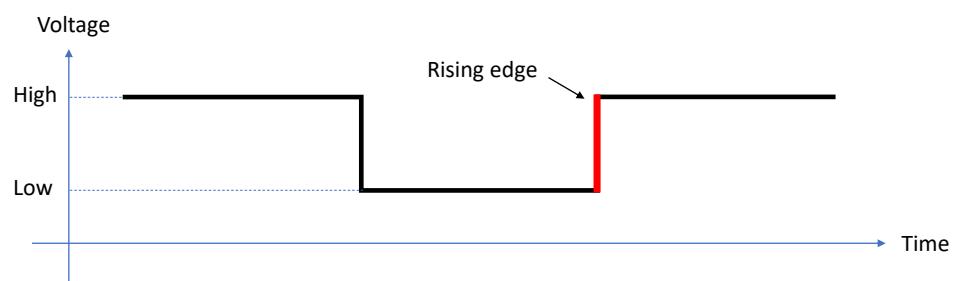


Figure 5.2: The scene of triggering FSRs for HS and TO events

These two FSRs need slightly different circuits to determine the correct time to trigger the interrupt. We configure the corresponding interrupt pins on EcoMini to trigger on rising edge, as shown in Fig. 5.2. For heel-strike events, we use the FSR attaching at the back end of the sole to detect the start of the force. Upon touching the ground, the FSR changes its resistance and causes the pull-down resistor to generate a rising edge. For toe-off events, we use the FSR attaching at the front end of shoes to detect the loss of ground-contact force on FSR, causing a pull-up resistor to generate a rising edge for the interrupt.

5.2 Step Event Detection

To evaluate the performance of the proposed LSTM method for detecting step events, we train and test our LSTM models with two different datasets. The first dataset is composed of the walking data from six volunteers walking in their comfortable speeds. The other dataset contains the walking data from one volunteer walking in three different speeds. These two datasets test the user dependency and walking speed dependency of the proposed algorithm. Then, we reduce the number of IMUs from five to one to see whether the data from one IMU is sufficient for gait detection. At last, we compare the performance of the proposed LSTM method to other algorithms that have been discussed in Section 2.

Mean error and precision are used as metrics to assess the performance of our methods. Mean error is calculated by averaging the differences between the estimated values and the ground truth. Positive and negative components cancel out each other. The resulting mean error indicates the trend that a model tends to underestimate or overestimate the target value. There exist several ways to calculate precision, in our experiments we define precision as the standard deviation of errors, shown in Eq. (5.3), to describe the repeatability of our measurements. x_i is estimation error of the i^{th} stride, \bar{x} is the mean error of all strides, and N is the number of strides.

$$\text{precision} = \sqrt{\frac{\sum_{i=1}^N (x_i - \bar{x})^2}{N - 1}} \quad (5.3)$$

5.2.1 User Dependency

To test the user dependency of the proposed LSTM model, we use cross-validation in our experiment. In the training phase, we use the data from the five subjects and leave the data from the other subject

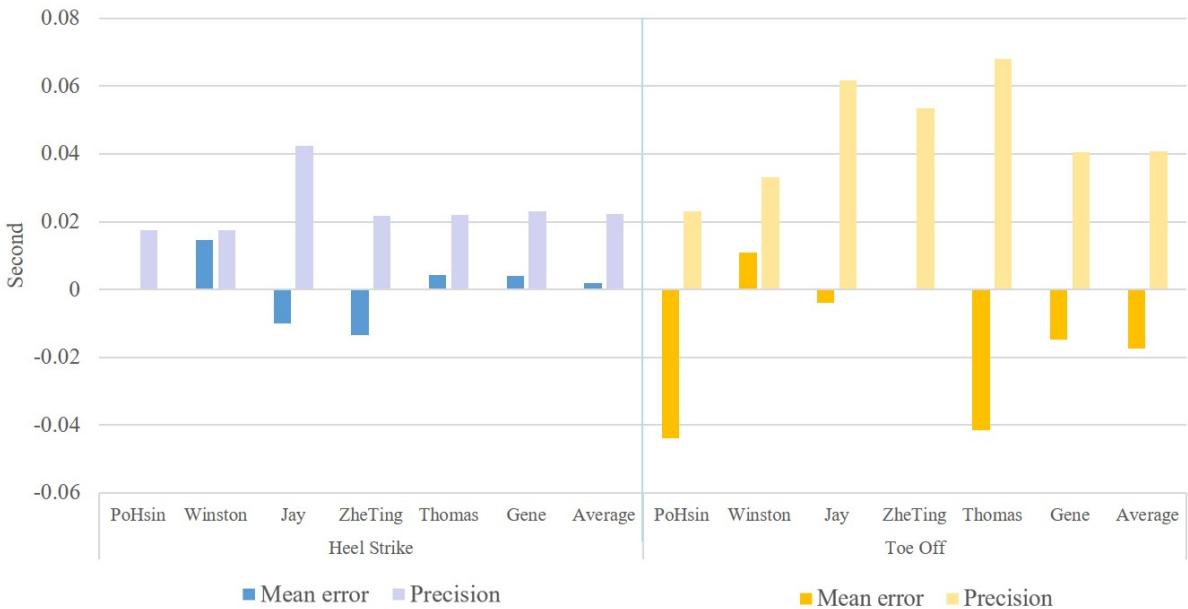


Figure 5.3: The mean error and precision of different testing subjects

for testing. Each subject takes turns being the test subject. The data from the five subjects are further divided into the validation set and the training set in a 1 : 9 proportion. We use cross-entropy as our loss function to be optimized and use Adam [27] as the optimization method. The LSTM layer contains 20 LSTM neurons, and we train the model for 100 epochs. The model with minimum validation loss is saved for testing.

Fig. 5.3 shows the result. First, we find that HS events are detected more precisely than TO events. This observation can also be found in [2, 7]. The reason is that the accelerometer is much more sensitive to the impact of the shoes on the ground but not so sensitive to shoes being lifted off the ground. For HS events, the average mean error is 0.002 seconds and the precision is 0.022 seconds. The variation of the results between different subjects is small. For TO events, the average mean error and precision are -0.018 and 0.041 seconds, respectively. The variance between different subjects becomes large compared to the result of HS events. We think the cause is that during the time our shoes leave the ground, the gyroscope signals are not fixed enough to indicate the TO events because of the different walking pattern of different subjects. Fig. 5.4 shows the Z-axis of the gyroscope data from the nodes on the left shoes of the six subjects. The yellow lines indicate step events. The rising edges and falling edges represent TO and HS events, respectively. We find that the relationship between waveforms of the gyroscope and the ground truth of TO events (the rising edges) varies from subject to subject. Even for the same subject, the timing of TO events varies slightly around the local minima, while the timing of HS events matches that of the local minima more precisely.

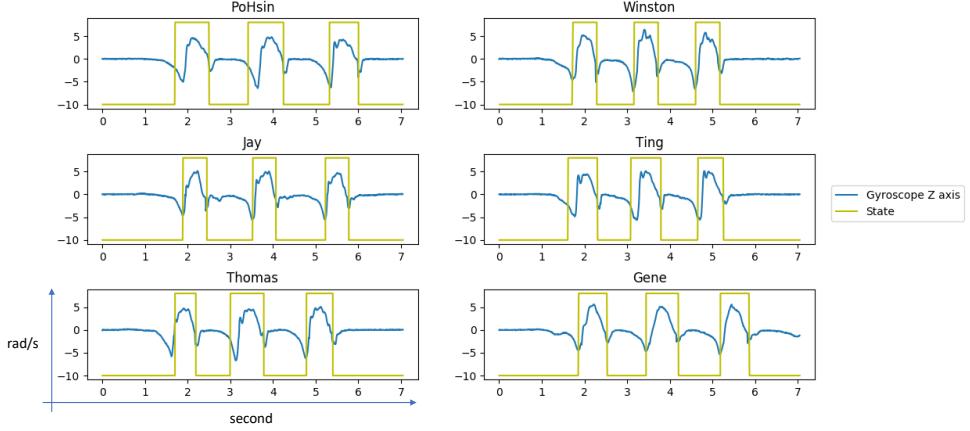


Figure 5.4: The Z axis of gyroscope data from different subjects

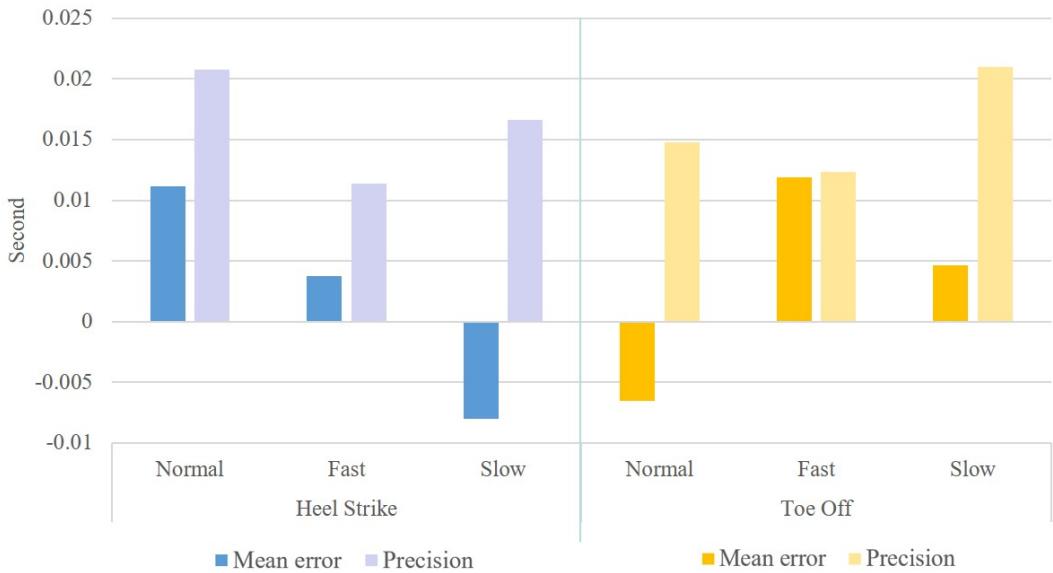


Figure 5.5: Walking under different speeds

5.2.2 Walking Speed Dependency

In this subsection, we evaluate the performance of the proposed LSTM model under three different walking speeds. We collect 54 strides for normal-speed walking, 90 strides for fast walking, and 66 strides for slow walking. The data is collected from a single subject. In the training phase, we use 70% of the data from each speed, and the remaining 30% of data is further divided into validation set and test set using a 1 : 1 proportion.

Fig. 5.5 shows the result. For both HS and TO events, the result of fast walking is better than the result of normal-speed walking and slow walking. We think the reason is that the measured acceleration is higher than usual during fast walking. Thus, the ups and downs of acceleration form a more explicit waveform for the LSTM model to learn.

5.2.3 Sensor Selection

To find how many sensors are needed in gait detection, we do the above described two experiments (user dependency and walking speed dependency tests) with five sensors and one sensor, separately. In the scenario of using one sensor, we use the sensor on the left shoe.

Fig. 5.6(a) and Fig. 5.6(b) show the results. We find that the performance of using these two numbers of sensors are similar. In the test of different walking speeds (see Fig. 5.6(b)), the mean error and precision even improved slightly when we use one sensor. We think the reason is that the HS and TO events are detected by the FSRs under the left shoe. Therefore, the data from the legs are not as critical as the data from the left shoe. This result implies that the data from the shoe-mounted IMU is sufficient for the proposed LSTM model to detect HS and TO events.

5.2.4 Comparing with related works

Table. 5.1 shows the performance of the proposed LSTM method and the works we described in Chapter 2. Compared to related works, the proposed LSTM algorithm achieves good results with the data from different walking speeds, but worse performance in the user-dependency test. In our user-dependency test, the subject's walking data is never seen by the LSTM model, so it is a more challenging task for our LSTM model. In the walking-speed dependency test, parts of the data from three walking speeds are used in the training phase, which enables the LSTM model to detect the timing of gait events more accurately.

Although the precision of the proposed LSTM method is not the best among the methods listed in Table. 5.1, LSTM still has several advantages when solving this problem. The LSTM units help us find useful information from the raw data of the IMU automatically; moreover, thresholds and rules are not needed. Also, due to the data-driven characteristic of deep-learning methods, we can expect the performance to improve when trained with sufficiently large data.

5.3 Stride Length Estimation

In this section, we evaluate the proposed mechanical model and LSTM-regression model in estimating stride length. Similar to the experiments we do for gait events, we do two experiments with the walking data from different subjects and in different walking speeds. The LSTM-regression model is



Figure 5.6: Evaluation of sensor numbers

Table 5.1: The performance of related works in gait event detection. unit: second

			Mean Error	Standard Deviation	RMSE
Accelerometer-based	Khandelwal et al. [6]	HS	–	0.010	–
		TO	–	0.010	–
	Selles et al. [7]	HS	0.034	0.025	–
		TO	0.019	0.036	–
	Mansfield et al. [2]	HS	–	0.065	–
		TO	–	–	–
Gyroscope-based	Maqbool et al. [8]	HS	0.011	0.018	–
		TO	-0.008	0.035	–
	Lee et al. [10]	HS	0.019	–	0.026
		TO	-0.00867	–	0.025
	Catalfamo et al. [11]	HS	-0.008	0.009	–
		TO	0.05	0.014	–
Proposed Method	Proposed LSTM algorithm with different subjects	HS	-0.008	0.028	0.031
		TO	-0.015	0.046	0.052
	Proposed LSTM algorithm with different speeds	HS	0.005	0.015	0.019
		TO	-0.008	0.015	0.022

tested using two different inputs: raw data from five IMUs and features extracted from the mechanical model. In addition, we do a user-dependent test, which trains and tests the LSTM model with the data from the same subject. At last, we compare our performance with that of related works.

5.3.1 Cross-subjects Test

Mechanical Model

Due to the limitation of the ultrasonic sensor (HC-SR04) used for the ground truth, the walking data are collected by walking toward a wall from 4 m away. The number of strides for a human to walk within this distance is around three strides. Thus, the collected walking data is composed of many three-stride units.

When testing the mechanical model, we find that the result of the first stride and the last stride are different from the middle stride. Most of the first strides are underestimated, while most of the last strides are overestimated. Base on this observation, we enlarge the estimation of all first strides by 5%, and shrink the estimation of all last strides by 5%. We believe this difference can be explained by the fact that the starting and ending gait is different from the “middle” strides for continuing walking.

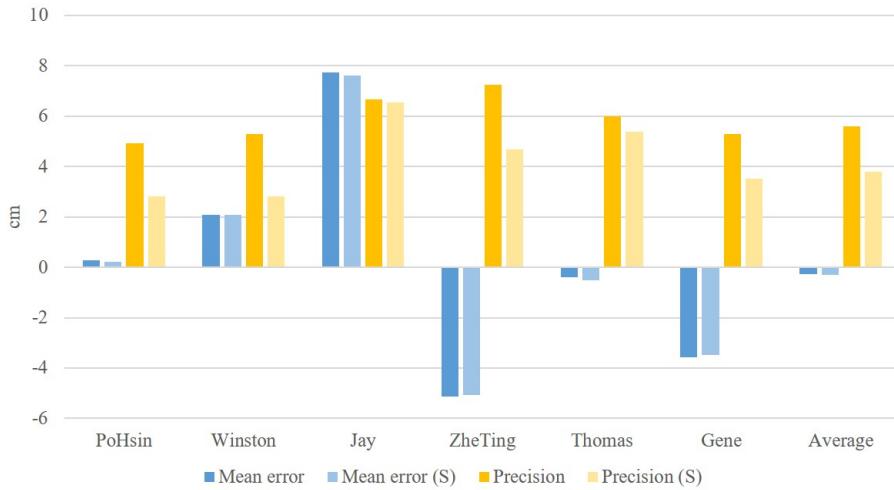
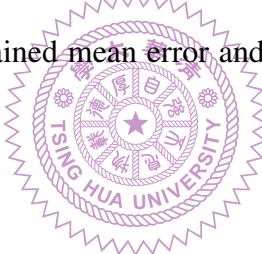


Figure 5.7: Mechanical model - Difference Subjects

Fig. 5.7 shows the mean error and standard deviation of using the proposed mechanical model in stride length estimation. Two blue lines and two orange lines represent the mean error and precision, respectively. The lines with darker color show the original result, and the lines with (S) labels show the result after scaling the first and the last strides. The scaled results show an improvement in precision by around 1.8 cm on average. The obtained mean error and precision on average are -0.3 cm and 3.8 cm, respectively.

LSTM Model



In the training phase of this LSTM model, the way we select the training data is different from that for the LSTM model in gait detection. Walking data of each subject is divided into the training set, validation set, and test set using a 70 : 15 : 15 proportion. Then, we aggregate the training set from all subjects to train our LSTM model.

Fig. 5.8 shows the result. The labels marked with (R) and (M) indicate that the input to the LSTM is either raw data or model features, respectively. It is shown that by using features extracted from the mechanical model, the obtained precision has a 1.87 cm improvement compared to the result of using raw data for training. It implies that if we extract useful features for LSTM models, they have a better chance of achieving better performance. The mean error and precision on average are 0.22 cm and 3.8 cm, respectively.

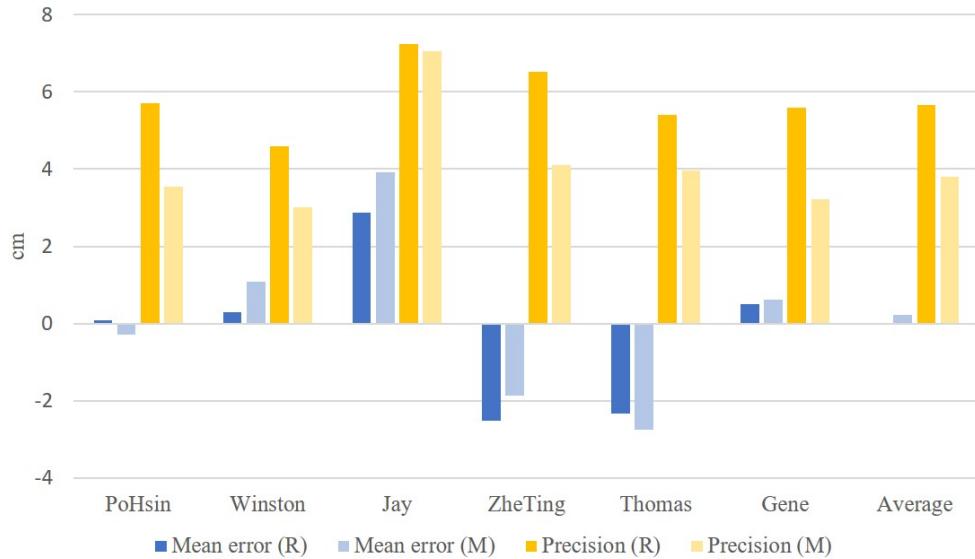


Figure 5.8: LSTM with different inputs - Difference Subjects

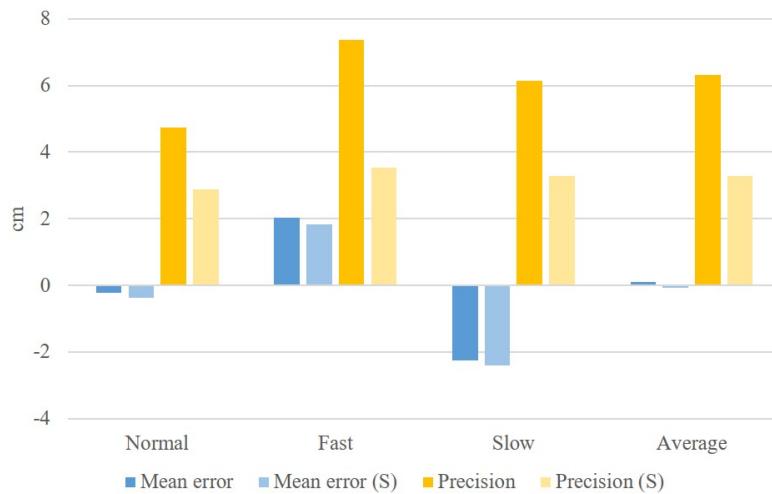


Figure 5.9: Mechanical model - Difference Speeds

5.3.2 Walking Speed Dependency

Mechanical Model

Fig. 5.9 shows the performance of the proposed mechanical model under different walking speeds.

The labels marked with (S) indicate the result after scaling the first and the last strides. Similar to the observation in the cross-subjects test, the scaling improves the precision by 3.02 cm.

LSTM Model

Fig. 5.10 shows the result of using the LSTM model with different inputs. The labels marked with (R) and (M) indicate the input to the LSTM model are raw data and model features, respectively. We

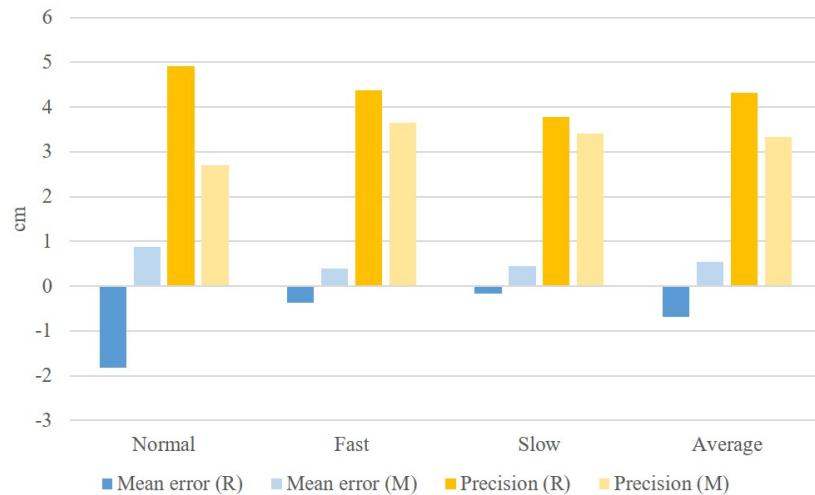


Figure 5.10: LSTM with different inputs - Difference Speeds

get a 0.99 cm improvement in precision by using the extracted features from the mechanical model.

5.3.3 User-dependent Test

In the user-dependent test, we want to know how much training data is needed for the proposed LSTM model to generate a good result. In addition, we want to know whether we can get a better result when we train a dedicated LSTM model for one subject. The performance of using different inputs, i.e., raw data and model features, are also compared. The walking data contains 198 strides of normal-speed walking.

Fig. 5.11 shows the performance with different numbers of training data. Both the mean error and precision are gradually improved when the size of training set increases, while the improvement becomes not so observable when we use more than 100 strides of walking data for training. To obtain a precision under 4 cm, around 70 strides of walking data are needed. In addition, we find that using model features results in an improvement from 1.7 cm to 2 cm in precision compared to that using raw data. With 160 strides of training data, the obtained mean error and precision are 0.08 cm and 3.11 cm, respectively. Compared to the result obtained by using the mechanical model for the same subject, which is 0.21 cm and 2.8 cm, our model shows an improvement in mean error with similar precision.

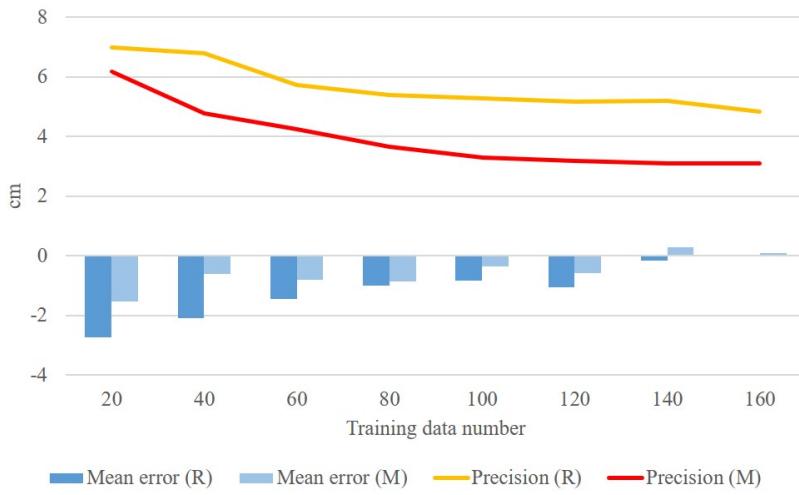


Figure 5.11: Testing with different numbers of training data

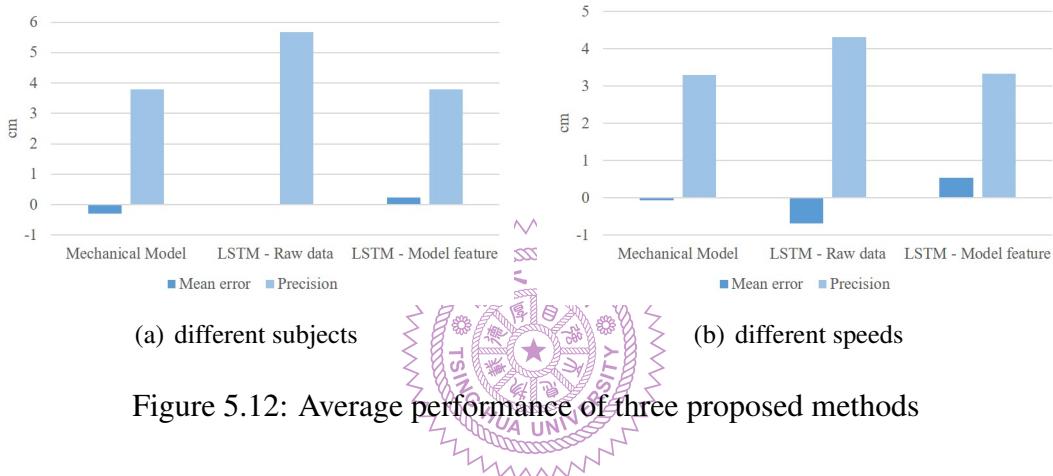


Figure 5.12: Average performance of three proposed methods

5.3.4 Performance Comparison Between Mechanical Model and LSTM Model

We propose three ways to estimate the stride length, and their performance is shown in Fig. 5.12. The mean error and precision are calculated by averaging the result from all six subjects and three walking speeds, respectively. The result of the proposed mechanical model and the LSTM model, which use the model feature as their inputs, are nearly the same. For the mechanical model, the error may come from BLE packet loss that causes incorrect integration of the angular rate. Another source of error is the placement of sensors. We use the Z-axis data of the gyroscope for integration, but we cannot guarantee that the sensor is placed exactly on the sagittal plane. Each node may tilt slightly to cause the angular rate to scatter to other axes. The LSTM models are also adversely affected by these two factors. Furthermore, the structure of the network and the parameters used in the training phase also influence the result of regression. In the three experiments described in this section, we find that by using model features as input, the LSTM model makes better estimation compared to the LSTM model using raw data as input, but it does not do better than the mechanical model in our tests.

5.3.5 Performance Comparison with Related Works

Table. 5.2 shows the performance of related works surveyed in Chapter 2. These results are directly taken from the corresponding papers, which means the walking data used in each work is different from others. The average stride lengths in related works range from 80 cm to 124.5 cm. In our experiments, we get a similar average stride length, which is 114.6 cm, in the collected walking data. Compared to the model-based approaches that use angles at joints and leg lengths to calculate the stride length, the proposed mechanical model improves both the mean error and precision. It implies that with more nodes and more dedicated model, we can get better results in this task using gyroscopes. For the methods that estimate stride lengths by integrating acceleration, we get similar mean error and precision with them, while the complicated signal processing and sensor orientation maintenance are not needed in the proposed LSTM method. Two methods that use machine learning in stride length estimation are also listed in Table. 5.2. Hannink et al. [25] estimates stride lengths using CNN with raw data from two ankle-mounted IMUs. It should be noted that their test is user-independent, which means the test data is composed of the walking data from subjects that the model has never seen in the training phase. The data used in their research is collected from 101 patients, and 12 strides are collected from each patient on average. Due to the difference in dataset composition and the difference in the testing method, it is difficult to compare the proposed LSTM models to the CNN-based method. Edel et al. [16] achieve good performance by first classifying the walking type (the speed and the direction) and then using dedicated parameters in an equation to estimate stride length. The two-level structure, which first analyzes the walking type and then uses a specific model to estimate stride length, may be able to improve the performance of the proposed LSTM model.

Table 5.2: The performance of related works in stride length estimation

Method		Mean Error	Standard Deviation	RMSE
Model-based	Aminian et al. [13]	–	–	7 cm
	Slarian et al. [17]	3.8 cm	6.6 cm	–
Accelerometer integration	Ferrari et al. [18]	0.66 - 0.99 cm ^{†1}	4.24 - 6.99 cm ^{†1}	–
	Rebula et al. [19]	–	–	4.58 cm [†]
CNN	Hannink et al. [25]	0.01 cm	5.37 cm	–
Linear equation	Edel et al. [16]	0.34 - 5.5 cm ^{†2}	1.72 - 4.10 cm ^{†2}	–
Proposed mechanical model	(different subjects)	-0.3 cm	3.8 cm	4.83 cm
	(different speeds)	-0.06 cm	3.29 cm	3.71 cm
Proposed LSTM model	(different subjects)	0.22 cm	3.8 cm	4.46 cm
	(different speeds)	0.54 cm	3.33 cm	3.61 cm

[†] The values are calculated from percentage to centimeters using the average stride length of the walking data in our experiments, which is 114.6 cm.

¹ The result differs with different sampling rates.

² The result differs with different sensor placements.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

We present a gait-detection method and a stride-length estimation method using recurrent neural networks (RNN) with LSTM units. These two methods both achieve good accuracy and precision for the datasets composed of the walking data from different subjects and different walking speeds.

The step-event detector uses the raw data from one shoe-mounted IMU as the input to its LSTM model. In the user-dependency test and the walking-speed dependency test for the step-event detector, the proposed method is shown to be compatible with different users and different walking speeds. Experimental results are -0.0008 s to 0.015 s in mean error and 0.015 s to 0.046 s in precision.

The stride-length estimator contains a mechanical model and two LSTM models. The mechanical model utilizes the natural constraint on human legs and calculates stride lengths by the angles at joints and leg lengths. We add the motion of shoes into the traditional mechanical model and obtain better results. It implies that with a more dedicated model, we can estimate stride length more precisely in a simple way. The obtained mean error and precision are -0.3 cm and 3.8 cm, respectively.

The two LSTM models take the data from five IMUs attached on legs and the left shoe. Their respective inputs are the raw data and the features extracted from the mechanical model. We evaluate their performance under different subjects and different walking speeds, and using model features improves the precision by 1.8 cm and 1 cm under these two tests compared to the model that uses raw data.

6.2 Future Work

While the proposed mechanical model improves the estimated stride lengths, we still get a 3.8 cm standard deviation of errors. To further improve the performance, we need to reduce the BLE packet loss that causes decreased integration of the angles at joints. This can be done by compressing the data before transmitting and then using a lower data rate to avoid the packet loss during high-speed transmission. The performance of IMU-based methods is also influenced by the signal-to-noise ratio (SNR) of the IMU, so we can expect improved estimation results using the IMU with higher SNR.

A two-level method may help the LSTM model. If we first identify the walking speed or identify the user from the walking data, we can use a dedicated model, which is trained for that specific subject, to estimate the stride length. A combination of the mechanical model and this two-level LSTM model can also be integrated. When we cannot identify the user, we switch to the mechanical model and use average leg lengths to estimate the stride length. Another possible improvement is to use a more complex network architecture that contains more layers, such as adding an autoencoder in the proposed network to help find critical features from the raw data.

The proposed gait parameters analysis system is used in straight-line walking. When applied to pedestrian dead-reckoning (PDR), our system needs to estimate the turning angles as well. The turning angle can be calculated by integrating the readings of gyroscopes, and more complex algorithms that deal with the drift of gyroscope such as Madgwick's filter [28] and Mahony's filter [29] may also need to be applied.

Bibliography

- [1] Antoon Th. M. Willemsen, Fedde Bloemhof, and Herman B.K. Boom. Automatic stance-swing phase detection from accelerometer data for peroneal nerve stimulation. *IEEE Transactions on Biomedical Engineering*, 37(12):1201–1208, 1990.
- [2] Avril Mansfield and Gerard M. Lyons. The use of accelerometry to detect heel contact events for use as a sensor in FES assisted walking. *Medical Engineering and Physics*, 25(10):879–885, 2003.
- [3] R.R. Torrealba, J. Cappelletto, L. Fermín-León, J.C. Grieco, and G. Fernandez-Lopez. Statistics-based technique for automated detection of gait events from accelerometer signals. *Electronics Letters*, 46(22):1483–1485, 2010.
- [4] Eric Watelain, Jérôme Froger, Franck Barbier, Ghislaine Lensel, Marc Rousseaux, François-Xavier Lepoutre, and André Thevenon. Comparison of clinical gait analysis strategies by French neurologists, physiatrists and physiotherapists. *Journal of Rehabilitation Medicine*, 35(1):8–14, 2003.
- [5] Cristian F. Pasluosta, Heiko Gassner, Juergen Winkler, Jochen Klucken, and Bjoern M. Eskofier. An emerging era in the management of Parkinson’s disease: wearable technologies and the Internet of Things. *IEEE Journal of Biomedical and Health Informatics*, 19(6):1873–1881, 2015.
- [6] Siddhartha Khandelwal and Nicholas Wickström. Identification of gait events using expert knowledge and continuous wavelet transform analysis. In *7th International Conference on Bio-inspired Systems and Signal Processing (BIOSIGNALS 2014), Angers, France, March 3-6, 2014*, pages 197–204. SciTePress, 2014.

- [7] Ruud W. Selles, Margriet A.G. Formanoy, Johannes B.J. Bussmann, Peter J. Janssens, and Henk J. Stam. Automated estimation of initial and terminal contact timing using accelerometers; development and validation in transtibial amputees and controls. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 13(1):81–88, 2005.
- [8] Hafiz Farhan Maqbool, Muhammad Afif Bin Husman, Mohammed I. Awad, Alireza Abouhossein, Nadeem Iqbal, and Abbas A. Dehghani-Sanj. A real-time gait event detection for lower limb prosthesis control and evaluation. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 25(9):1500–1509, 2017.
- [9] Darwin Gouwanda et al. A robust real-time gait event detection using wireless gyroscope and its application on normal and altered gaits. *Medical Engineering and Physics*, 37(2):219–225, 2015.
- [10] Jung Keun Lee and Edward J. Park. Quasi real-time gait event detection using shank-attached gyroscopes. *Medical & Biological Engineering & Computing*, 49(6):707–712, 2011.
- [11] Paola Catalfamo, Salim Ghousayni, and David Ewins. Gait event detection on level ground and incline walking using a rate gyroscope. *Sensors*, 10(6):5683–5702, 2010.
- [12] Kaiyu Tong and Malcolm H. Granat. A practical gait analysis system using gyroscopes. *Medical Engineering and Physics*, 21(2):87–94, 1999.
- [13] Kamiar Aminian, B. Najafi, C. Büla, P.-F. Leyvraz, and Ph. Robert. Spatio-temporal parameters of gait measured by an ambulatory system using miniature gyroscopes. *Journal of Biomechanics*, 35(5):689–699, 2002.
- [14] XunSheng Ji, Shourong Wang, Yishen Xu, Qin Shi, and Dunzhu Xia. Application of the digital signal procession in the MEMS gyroscope de-drift. In *Proc. 1st IEEE International Conference on Nano/Micro Engineered and Molecular Systems, 2006. NEMS'06*, pages 218–221. IEEE, 2006.
- [15] Li Qiang, Teng Jianfu, Wang Xin, Zhang Yaqi, and Quo JiChang. Research of gyro signal de-noising with stationary wavelets transform. In *Proc. Canadian Conference on Electrical and Computer Engineering, 2003. IEEE CCECE 2003.*, volume 3, pages 1989–1992. IEEE, 2003.

- [16] Marcus Edel and Enrico Köppe. An advanced method for pedestrian dead reckoning using BLSTM-RNNs. In *Proc. 2015 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–6. IEEE, 2015.
- [17] Arash Salarian, Pierre R. Burkhard, François J.G. Vingerhoets, Brigitte M. Jolles, and Kamiar Aminian. A novel approach to reducing number of sensing units for wearable gait analysis systems. *IEEE Transactions on Biomedical Engineering*, 60(1):72–77, 2013.
- [18] Alberto Ferrari, Pieter Ginis, Michael Hardegger, Filippo Casamassima, Laura Rocchi, and Lorenzo Chiari. A mobile Kalman-filter based solution for the real-time estimation of spatio-temporal gait parameters. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 24(7):764–773, 2016.
- [19] John R. Rebula, Lauro V. Ojeda, Peter G. Adamczyk, and Arthur D. Kuo. Measurement of foot placement and its variability with inertial sensors. *Gait & posture*, 38(4):974–980, 2013.
- [20] Diana Trojaniello, Andrea Cereatti, Elisa Pelosi, Laura Avanzino, Anat Mirelman, Jeffrey M. Hausdorff, and Ugo Della Croce. Estimation of step-by-step spatio-temporal parameters of normal and impaired gait using shank-mounted magneto-inertial sensors: Application to elderly, hemiparetic, Parkinsonian and choreic gait. *Journal of Neuroengineering and Rehabilitation*, 11(1):152, 2014.
- [21] Nicolas Saunier, Ali El Husseini, Karim Ismail, Catherine Morency, Jean-Michel Auberlet, and Tarek Sayed. Estimation of frequency and length of pedestrian stride in urban environments with video sensors. *Transportation Research Record: Journal of the Transportation Research Board*, (2264):138–147, 2011.
- [22] Chiraz BenAbdelkader, Ross Cutler, and Larry Davis. Stride and cadence as a biometric in automatic person identification and verification. In *Proc. Fifth IEEE International Conference on Automatic Face and Gesture Recognition*, pages 372–377. IEEE, 2002.
- [23] Erik E. Stone and Marjorie Skubic. Passive in-home measurement of stride-to-stride gait variability comparing vision and Kinect sensing. In *Proc. 2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 6491–6494. IEEE, 2011.

- [24] Wei Chen, Ruizhi Chen, Xiang Chen, Xu Zhang, Yuwei Chen, Jianyu Wang, and Zhongqian Fu. Comparison of EMG-based and accelerometer-based speed estimation methods in pedestrian dead reckoning. *The Journal of Navigation*, 64(2):265–280, 2011.
- [25] Julius Hannink, Thomas Kautz, Cristian F. Pasluosta, Jens Barth, Samuel Schülein, Karl-Günter Gaßmann, Jochen Klucken, and Bjoern M. Eskofier. Stride length estimation with deep learning. *arXiv preprint arXiv:1609.03321*, 2016.
- [26] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.
- [27] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [28] Sebastian Madgwick. An efficient orientation filter for inertial and inertial/magnetic sensor arrays. *Report x-io and University of Bristol (UK)*, 25:113–118, 2010.
- [29] Robert Mahony, Tarek Hamel, and Jean-Michel Pflimlin. Nonlinear complementary filters on the special orthogonal group. *IEEE Transactions on automatic control*, 53(5):1203–1218, 2008.

