Modèles de Markov cachés récents pour la détection et la reconnaissance d'activités à partir d'un capteur IMU placé sur une jambe

Haoyu Li

N^o d'ordre NNT : 2019LYSEC41

THESE

DOCTEUR DE L'ÉCOLE CENTRALE DE LYON

Spécialité: Informatique

Recent Hidden Markov Models for Lower Limb Locomotion Activity Detection and Recognition using IMU Sensors

---

dans le cadre de l'École Doctorale InfoMaths

présentée et soutenue publiquement par

HAOYU LI

December 2019

Directeur de thèse: Stéphane Derrode, PR à l'Ecole Centrale de Lyon

Co-directeur de thèse: Wojciech Pieczynski, PR à Télécom SudParis

JURY

François Septier	Université Bretagne Sud	Rapporteur
Faicel Chamroukhi	Université de Caen	Rapporteur
Latifa Oukhellou	IFSTTAR	Présidente
Lamia Benyoussef	EPITA	Examinatrice
Stéphane Derrode	École Centrale de Lyon	Directeur de thèse
Wojciech Pieczynski	Télécom SudParis	Co-directeur de thèse

Contents

Abstract

The thesis context is that of the quantified self, a movement born in California that consists in getting to know oneself better by measuring data relating to one's body and activities. The research work consisted in developing algorithms for analyzing signals from an IMU (Inertial Measurement Unit) sensor placed on the leg to recognize different movement activities such as walking, running, stair climbing... These activities are recognizable by the shape of the sensor's acceleration and angular velocity signals, both tri-axial, during leg movement and gait cycle.

To address the recognition problem, the thesis work resulted in the construction of a particular hidden Markov chain, called semi-triplet Markov chain, which combines a semi-Markov model and a Gaussian mixture model in a triplet Markov model. This model is both adapted to the nature of the gait cycle, and to the sequence of activities as it can be carried out in daily life. To adapt the model parameters to the differences in human morphology and behavior, we have developed algorithms for estimating parameters both off-line and on-line.

To establish the classification and learning performance of the algorithms, we conducted experiments on the basis of recordings collected during the thesis and on public dataset. The results are systematically compared with state-of-the-art algorithms.

Keywords: Quantified-self, activity recognition, inertial measurement unit, triplet Markov model, semi-triplet Markov chain, on-line estimation.

Résumé

Le contexte de la thèse est celui du quantified-self, un mouvement né en Californie qui consiste à mieux se connaître en mesurant les données relatives à son corps et à ses activités. Les travaux de recherche ont consisté à développer des algorithmes d'analyse des signaux d'un capteur IMU (*Inertial Measurement Unit*) placé sur la jambe pour reconnaître différentes activités de mouvement telles que la marche, la course, la montée d'escalier.... Ces activités sont reconnaissables grâce à la forme des signaux d'accélération et de vitesse angulaire du capteur, tous triaxiaux, pendant le mouvement des jambes lors du cycle de marche.

Pour résoudre ce problème de reconnaissance, les travaux de thèse ont permis la construction d'un modèle de chaîne de Markov cachée particulier, appelé chaîne triplet semi-Markov, qui combine un modèle semi-Markov et un modèle de mélange gaussien dans un modèle de Markov triplet. Ce nouveau modèle est adapté à la fois à la nature du cycle de marche et à l'enchaînement des activités que l'on peut réaliser dans la vie quotidienne. Pour adapter les paramètres du modèle aux différences de morphologie et de comportement humain, nous avons développé des algorithmes d'estimation des paramètres en ligne et hors ligne.

Pour établir les performances d'apprentissage et de classification des algorithmes, nous avons mené des expériences sur la base d'enregistrements recueillis pendant la thèse et d'un ensemble de données publiques. Les résultats sont systématiquement comparés aux algorithmes de reconnaissance actuels.

Mots-clés: Quantified-self, reconnaissance d'activités, capteur inertiel, modèle de Markov triplet, chaine semi-markovienne, estimation on-line.

Chapter 1

Introduction

Quantified self is gaining interest in our modern life. It is a movement started from 2007 at California, which refers both to the self-tracking with technology and to a community of users and makers of self-tracking tools. Self-tracking is related to 'monitoring, measuring and recording elements of one's body and life as a form of self-improvement and self-reflection', commonly using digital technologies [1]. The objective of this practice is to obtain a precise account on particular elements of human's life, and keep tracking of them over time. The quantified self movement begun to emerge in recent years, perhaps the most important factor is the reduction in the development costs various techologies. Another reason is that more and more people are concerned about their condition and their health marks. One user from Finland explains why quantified self technology is such important, he used a body wearing sensor network to track his weight, fat percentage, sleeping time for years. And he comments: 'I want to improve myself. I want to know where I am and where I'm going. This device has taught me how to exercise and eat right' [2]. As it suggests, via quantified self technologies, people can generate their own expert knowledge about themselves.

The related practices aim at tracking the physical, mental, and emotional performances of our daily life. For example, using GPS to locate where we are and track the routine of our daily life, using smart watch or smart phone to monitor activity and sleep... And the generated time series data could be various, such as position, motion related signals, electromyography (EMG) signals, electrocardiography (ECG) signal... There are many research directions in the field of quantified self. They can be commonly classified into three categories: (i) the devices used for collecting quantified self data; (ii) the identification methods used for extract-

ing data of interest; (iii) the assessment and utilization of the identified data. In this thesis we were interested in using IMU (inertial measurement unit) sensor to recognize the lower limb locomotion activities.

Our daily life can be described by a series of activities, such as sleeping, walking, sitting, eating food, *etc.* The order and sequence of the activities can be very different between two people, and detecting the kind of activity and their duration can be very interesting. Indeed, individuals can obtain a general view of their life over a long period of time, which is beneficial for them to manage their schedule, such as making fitness program or rehabilitation plan. At the people level, knowing what kinds of activity have been done by a large number of people gives an insight view of the statistical inference of health markers such as obesity, physical inactivity... However, the main activity monitoring concerns its ability to provide an according daily report of our activity. By knowing the amount of sports we have done and the regular time we sleep, we will be aware of of the extent to which we lead a healthy life. Based on these information, specialists can provide advice to reduce anxiety or insomnia for example, or even assist the doctors to diagnose the sickness of their patients. More than that, abnormal activity monitoring can also prevent accident. Fall detection techniques can help the elderly to quickly access to an emergency center. Of course these activities are performed by people very often, if looking at all the activities that people perform every day, the lower limb locomotion activity is one of the most frequent activity. Lower limb locomotion activity is the motion that drives a person moving from one place to another place by using lower limb, including walking, running, cycling... This kind of activity is inevitable in our daily life. As mentioned above, the great amount of quantified self practices shows the importance of activity monitoring, while the fundamental of activity monitoring is the correctness of automatic activity recognition.

When a person is performing motion activity, various data can be acquired that relate to the activity, such as ECG related to the heart, EMG of a specific muscle, kinematic information of some body parts, *etc.* A lot of commercial products are available to acquire these data. Among them, an IMU sensor is able to acquire the kinematic information, generally it measures acceleration, angular rate and

magnetic fields. Since an IMU sensor is very small and light—also it is able to transfer data by wireless connection—, it can be worn on the body without affecting the motion. Since the thesis investigates the lower limb locomotion activity, we assume that the IMU sensor is placed on the lower limb to acquire the kinematic information of the motion. As we know, when people carry out different activities, the motion of the body will also be different. For example, walking is more moderate than running, which means the two legs are alternating slower in walking than in running. Additionally, the feet in walking are not lifted as high as running. Also, the same activity is not performed exactly the same way by different people, due to the morphology and attitude difference. As a result, the measured kinematic information of one activity will differ from the others. This property provides us a way to recognize different activities from the signals, if it is possible to find a mechanism able to distinguish each pattern of the activity.

In most of the lower limb locomotion activities, continually moving from one place to another place needs our two legs to repeatedly alternate a specific pattern. Normally the two legs cannot lift or make forward at the same time, except for some special motions like distance jump. Therefore, one cycle of the alternative pattern is called gait cycle, it consists of the phases that leg starts lifting (or making forward) and waits the other leg lifting (or making forward). Consequently, most of the lower limb locomotion activities share a similar periodic pattern and have similar phases inside one cycle. This kind of periodic cycle and the phases may provide a chance to tell the activities apart.

In this context, the thesis mainly focuses on developing specific Markov chain models to detect gait cycles and recognize the related activities. Because hidden Markov chains are very suitable for handling time series data and state. The Markov models are very simple, with robust ways to learn parameter both off-line and on-line. Recently, new advances in Markov modeling have been reported (*i.e.* pairwise and triplet Markov models [3–5]), and it appears to be of great interest to evaluate and compare those models with the state-of-the-art. So, we propose some specific Markov models that are suited for recognizing lower limb locomotion activity and detecting gait cycle. After proper model construction and parameter learning, hid-

den states can be recovered from the observed data only. Hence, the goal of the thesis is to establish some specific Markov chain models to represent the real lower limb locomotion activity properly. Besides, the established models should be suitable for on-line recognition. Therefore, the problem of parameters updating is also studied.

Outline of the thesis

The thesis is divided into six Chapters, the remaining five Chapters are organized as follows:

Chapter 2 introduces the recent literature work of quantified self researches. The review introduces several aspects separately. Firstly, different types of sensors that can be used for quantified self practices are presented. Comparison of IMU with other sensors is described, and the advantages of IMU explain the reason why it is chosen as the source of data for lower limb locomotion activity. A view of the practices is given to show the wide usage of quantified self, and prevalent used algorithms are described. Finally, some useful public datasets are described and compared according to their characteristics and specificities.

Chapter 3 focuses on developing a specific left-to-right hidden Markov chain (LR-HMC) to detect the gait phases of one individual activity at constant speed. The four considered activities are: walking, running, stair ascent and stair descent. In LR-HMC, a specific transition graph of hidden states is proposed to mimic the real transition of the gait phases. The class-conditional observation density is assumed to be Gaussian, where the observation of the model is the feature extracted from the kinematic information of IMU sensor. Then an Expectation-Maximization (EM)-based algorithm is introduced to automatically learn the parameters of the LR-HMC.

Chapter 4 depicts a specific triplet Markov chain (TMC) to simultaneously recognize activity and gait phases, by introducing an auxiliary hidden state to represent activity. To overcome the disadvantages of the class-conditional observation density of Gaussian type, used in LR-HMC, this Chapter introduces a non-parametric rep-

resentation by histogram density. This specific TMC is then referred as TMC-HIST. Because of the non-parametric density, EM algorithm is no longer applicable for TMC-HIST, so we introduce a method called "Iterative Conditional Estimation" (ICE) to learn the parameters. Based on the developed TMC-HIST model, an adaptive on-line recognition algorithm is then proposed. It allows to recognize the activity and gait phases at run time, and also to adjust its parameters gradually, to approach the activity pattern of the person. An experiment containing four activities —walking, running, stair ascent and stair descent— is performed by ten subjects at their preferred speed, the IMU sensor being placed on a shoe.

Chapter 5 also aims a developing an algorithm that recognizes activity and gait phases simultaneously, and on-line applicable as well. To overcome the disadvantages of TMC-HIST proposed in Chapter 4, this Chapter proposes a specific parametric semi-Markov chain to fulfill the task. Firstly, the semi-Markov structure is introduced into the TMC through a new auxiliary hidden state. Semi-Markov structure allows the hidden state to keep the same in a period of time, which is naturally consistent with the real situation. Gaussian mixture model (GMM) is utilized to form the class-conditional observation density. GMM density helps to approximate the non-Gaussian like class-conditional observation density. At the meantime, it involves the correlations among each dimension of class-conditional observation density, whereas histograms in previous chapter only modelize the marginal density of each dimension. The TMC equipped with the semi-Markov structure and GMM density is referred as SemiTMC-GMM. In this model, EM-based parameter learning algorithm is applicable, and an on-line EM parameter learning algorithm is adapted to the model. The experimental results show that, in batch mode recognition, SemiTMC-GMM obtains better performance than TMC, and also outperforms other comparative methods. While in on-line mode recognition, SemiTMC-GMM outperforms the TMC-HIST.

Finally, Chapter 6 summarizes the main contributions of this thesis. The limitations of the proposed model are presented, the corresponding possible solutions are also discussed, which give an outlook of possible future work.

Chapter 2

Background

The quantified self related technologies can benefit our lives in many ways, if personal data can be obtained properly. For example, smart devices such as smart watch and smart suit are able to collect medical information, which suggests the health status of the user and helps him/her to manage his/her life; navigation devices are able to know where the user is and show him/her the way to their destination. Because of the great benefits of the quantified self related technologies that can bring to our daily life, great research interest in various areas has been raised for exploiting the potential of it.

Given the extremely wide range of quantified self related researches, this Chapter mainly introduces some basic technique issues related to quantified self matters. The involved aspects are: (i), the numerous sensors that may be used for quantified self; (ii), the wide related research fields and applications; (iii), the prevalent algorithms used in recent years; (iv), some public datasets related to the work conducted in this thesis.

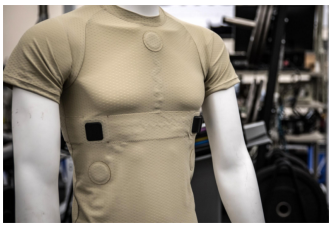
2.1 Available sensors for collecting data

As we know, human can be 'quantified' by different kinds of data, including but not limited to weight, calorie consumption, position, heart rate, respiration volume, motion kinematics, *etc.* Thus, in order to measure different data, many types of sensors should be used. The sensors related to quantified self can be categorized into two categories: the group of wearable sensors, and the group of non-wearable sensors.

2.1.1 Wearable sensor

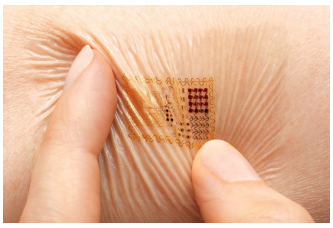
Commonly, wearable sensors are manufactured in small size and light weight, thus they can be worn on human body without affecting mobility of people. As shown in Figure 2.1, wearable sensors can be placed on different body parts, and their functions are various based on the usage.

(a) Smart watch.



(b) ECG sensor.

(c) Gesture control glove.



(d) Flexible sensor.

(e) VR or AR headset.

(f) Portable camera.

Figure 2.1: Some wearable sensors[1].

Smart watch, Figure 2.1a, is a highly integrated commercial product that contains several sensors inside. Commonly there are at least three kinds of sensors in a smart watch: IMU module, GPS and optical heart rate monitor. As a matter of fact, IMU [6–9] is not a single sensor, but is a group of inertial sensors made of three accelerometers and three gyroscopes. Sometimes it also contains a 3D magnetic sensor. It measures the kinematic information, *i.e.* acceleration, angular rate and magnetic field readings, in the tri-axial coordinate system of the sensor. So IMU sensor is very suitable for analyzing the motions since it provides the kinematic information. Also, IMU is used as an assistance of navigation in some industrial conditions, because the kinematic information can be used for estimating position.

[1]The products in Figure 2.1: (a), Apple watch series 4, site in `https://www.apple.com/apple-watch-series-4/`; (b), Samsung smart suit, site in `https://mygoodplanet.com/smart-clothing/`; (c), Gest, site in `https://gest.co/`; (d), MC10 Biostamp, site in `https://www.mc10inc.com/`; (e), Vuzix AR headset, site in `https://www.vuzix.com/Vuzix-Remote`; (f), AXON body camera, site in `https://www.axon.com/`.

GPS [10, 11] is a positioning sensor that generates the latitude and longitude were the user is, by receiving the signals from synchronous orbit satellites. GPS is able to give accurate position and track the routine of the user. But unfortunately, GPS is very vulnerable to the satellite signal, so it is commonly only available outdoor without barrier (street, mountain but not tunnel). The optical heart rate monitor uses a methodology called photoplethysmography (PPG) to measure heart rate. A PPG is often obtained by using a pulse oximeter which illuminates the skin and measures changes in light absorption [12]. Then, the heart rate is obtained by computing the changing rate of PPG. Apart from these three sensors, some other types of sensors can also be found in a smart watch. For example, the embedded barometer measures air pressure. The air pressure value can be used for estimating the elevation, while the value tendency can forecast short term changes in the weather. This functionality is extremely useful during mountain climbing and parachuting.

As the name suggests, ECG sensor [13] records the electrocardiogram. In the past, ECG was commonly seen in hospital and was used by patients to monitor the status of their heart. The usage of ECG was limited because it needed cables to connect the electrodes and the processor. But nowadays, as shown in Figure 2.1b, ECG sensor can be packaged in a small case and the data is transferred through wireless connection. The improvement allows ECGs to be worn on the body. Unlike the optical heart rate monitor only provides the heart rate, ECGs provide richer data related to electrocardiogram. Thus, the applications based on ECG are able to monitor our heart at any time and anywhere, medical advice can then be deduced based on the ECG signals.

Gesture control [14, 15] is a new direction of tracking the motion of human and controlling the machine, such as using gesture control glove to control computer, as shown in Figure 2.1c. The idea of a gesture control system is to track the motion of human with appropriate sensors, and then to capture a pre-defined motion gesture. The position where the sensor is put on the body depends on the requirement of the gesture, but normally the sensors are placed on the arm or hand since people can make rich and agile gestures with upper limb. There are many kinds of sensors that can be used for tracking motion, while particularly for the wearable sensors, IMU

and EMG are preferably used for gesture control [16,17]. An EMG sensor measures the electricity signal of the muscle through electrode, and by analyzing the signal we are able to know whether the muscle is in contraction or stretching. Then, the motion is tracked by knowing the behavior of a set of muscles. But the electrode of EMG needs to be tightly attached to the skin, whereas an IMU sensor does not have this restriction. Some other sensor-based gesture control devices can also be found, such as using cameras to track the motion, which belongs to non-wearable sensors and will be described in the following section.

Unlike the rigid shape of traditional sensors, flexible sensor [15] can be bent to some extent. This property enables the sensor to be placed on non-flat surfaces. Particularly for quantified self use, flexible sensor can be extremely small and light. As shown in Figure 2.1d, the thin flexible circuit is highly integrated with sensors such as ECG, EMG, temperature sensor, strain gauges, wireless communication oscillator. Thus, it measures a wide range of people vitals, including the temperature, hydration levels, strain, muscle, heart rate, *etc.* Then the collected data can be transferred to the server for further analysis. Because of the flexible property and extremely light weight, this kind of sensor is quite ideal for monitoring health status at any time, no matter whether the user is sleeping or doing something else.

Virtual reality (VR) headset, Figure 2.1e, is a device that presents built view to the wearer, while augmented reality (AR) headset overlays digital content on top of the real world. The two kinds of headsets are not just displayers, they are also integrated with motion sensing sensors, some headsets even have additionally an eye tracking sensor [18]. Therefore, the headsets are able to track the head motion of the wearer, and record the first-person view. This characteristic is quite useful in the fields of communication, remote control, entertainment and education, *etc.* And it brings a new way of seeing the world.

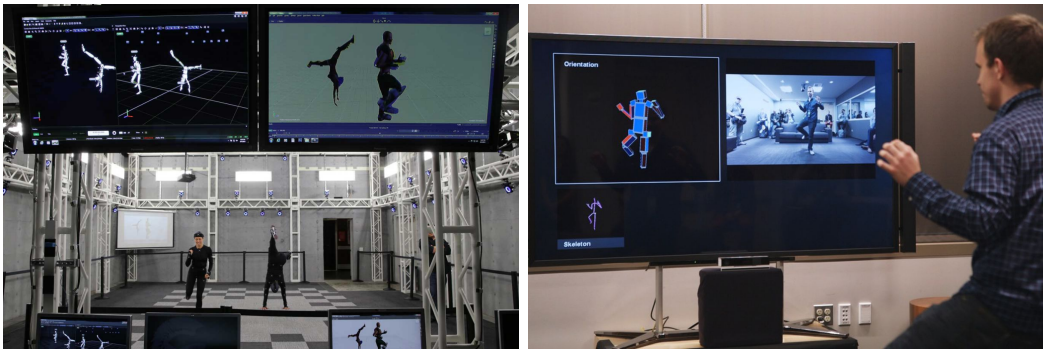
Cameras have existed for more than one century. With the development of the technology, nowadays cameras can be manufactured in a very small size and be placed on human body, as shown in Figure 2.1f. The small size camera placed on the body is generally regarded as a device to record life in first-person view [19]. The portable cameras are usually utilized in the fields of extreme sports and criminal

prosecutions.

Presently, there are many other types of wearable sensors used for quantified self, such as the pressure sensor [20, 21] that measures the forces produced by feet. Besides, we believe that new sensors and integrated products will come out in the future, which may probably enrich the field of quantified self further.

2.1.2 Non-wearable sensor

The non-wearable sensors for the use of quantified self are limited because the sensors are not attached to human body. Unlike the wearable sensors that measure the signals like heart rate, muscle strain, skin humidity, non-wearable sensors mainly detect the behavior of human. Normally these non-wearable sensors are installed at some places and have a fixed detection area If a large detection area is required, then it needs several sensors to cooperate together. In this Section we mainly describe some most commonly seen sensor systems that are used for quantified self.



(a) Using markers. (b) Without marker.

Figure 2.2: Non-wearable sensors using cameras[2].

Camera is very prevalently used in the motion analysis, because pictures and videos have been extremely investigated for extracting interested objects, including human bodies and some other salient properties. The camera-based motion capture system is well developed and has been applied in many fields, such as sport sciences, animation and robotics. Based on the detection methodology, the camera-based

[2]The products in Figure 2.2: (a), OptiTrack Prime 41, site in https://optitrack.com/products/prime-41/; (b), Kinect, site in https://developer.microsoft.com/fr-fr/windows/kinect.

motion capture systems can be divided into two categories. The first category tracks trajectory of the optic markers attached on the human body, see Figure 2.2a. Normally people is wearing the markers and performing the motion within an area that is surrounded by at least four well calibrated cameras [22, 23]. Then the cameras will give accurate 3D position of the markers at run-time, through solving the geometric problem. Therefore, if two or more markers on the body part are placed, like lower arm and shank, it is able to obtain the motion by connecting the markers. On the contrary, the second category does not need marker to detect the motion, in Figure 2.2b. In fact, this kind of system uses a set of different cameras to capture the images of the interested human body [24–28], then uses algorithms to segment each body part and estimate the body motion in 3D space. In the system, two cooperating cameras are involved, a conventional color camera is used for recording general images of scene, and an infrared-based depth camera for obtaining depth information by analyzing the infrared reflected from the objects. Then an embedded algorithm extracts human skeleton structure from the general images, consequently the skeleton structure is combined with the depth data to provide how the body is moving in the 3D space.

The two kinds of camera systems have their own advantages and disadvantages. The first system can track accurate position with an error less than 1 mm, and velocity and acceleration can be obtained based on the accurate positions. Besides, it tracks almost all the objects if the marker can be attached on. While the second system normally only detects the motion of human, and the estimated motion is not as accurate as that of the first system, velocity and acceleration are also not available directly. However, the cost of the first system commonly is more than ten thousand of dollars, while a commercial product of the second system only costs about four hundred dollars. Because of that the first system requires several cameras placed at different locations, so it needs a large room to run. And if one camera is moved, the entire system needs to be re-calibrated to ensure the accuracy. Whereas the second system is quite small and generally only needs to be calibrated at the first time of use.

Another system for localization and motion analysis is based on wireless bea-

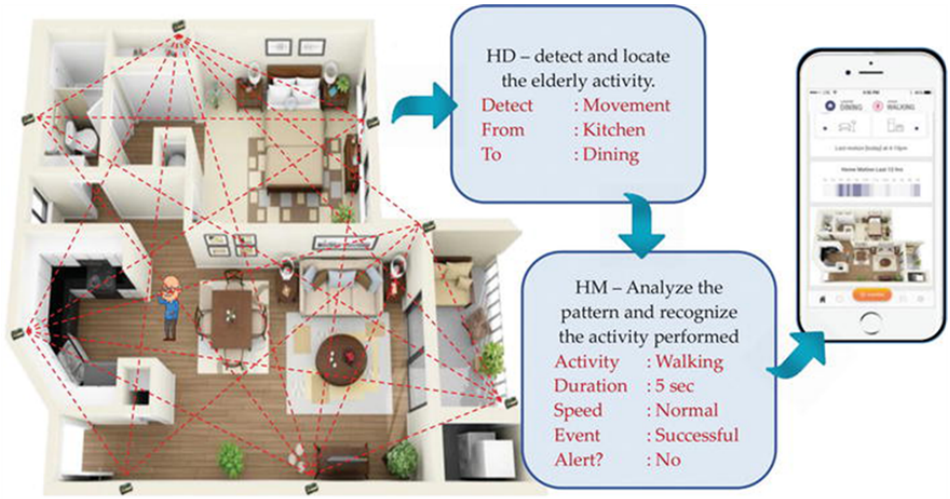


Figure 2.3: ZigBee-based sensor network deployed for elderly care application with the integration of mobile apps visualization (Extract from [29]).

con [30–35], such as WIFI, Bluetooth, ZigBee sensors. A schematic graph of a ZigBee-based localization sensor network [29] is shown in Figure 2.3. Normally, the wireless signals like WIFI, Bluetooth, ZigBee are used for communication. But apart from information of the signals convey, the wireless beacon is also aware of the signal strength. Thus, if a set of wireless sensors are well arranged in an area, the received signal strength (RSS) by each beacon may indicate whether there are obstacles that block the signal. When a human is moving in the area, it may cause fluctuations of RSS in each beacon. By analyzing the RSS fluctuations in the beacons, it is able to obtain the localization of the human. Particularly, if plenty of beacons are placed in an array, more details of the motion can be obtained, which may probably be used for motion analysis. The wireless signal-based localization and motion analysis system is normally installed indoor, used as a surveillance system for the elderly, or for patients in a hospital or nursing home.

Compared with the camera-based system, a wireless beacon-based system costs much less money. Also the radio signal is much less influenced by the obstacles, which means a wireless beacon-based system covers larger area and uses fewer sensors. However, the system may fail when the layout of the room is changed, and it is very difficult to track two or more persons moving in the area at the same time.

[3]Figure from site: http://cm.jo/portfolio/smarthouse-arduino/.

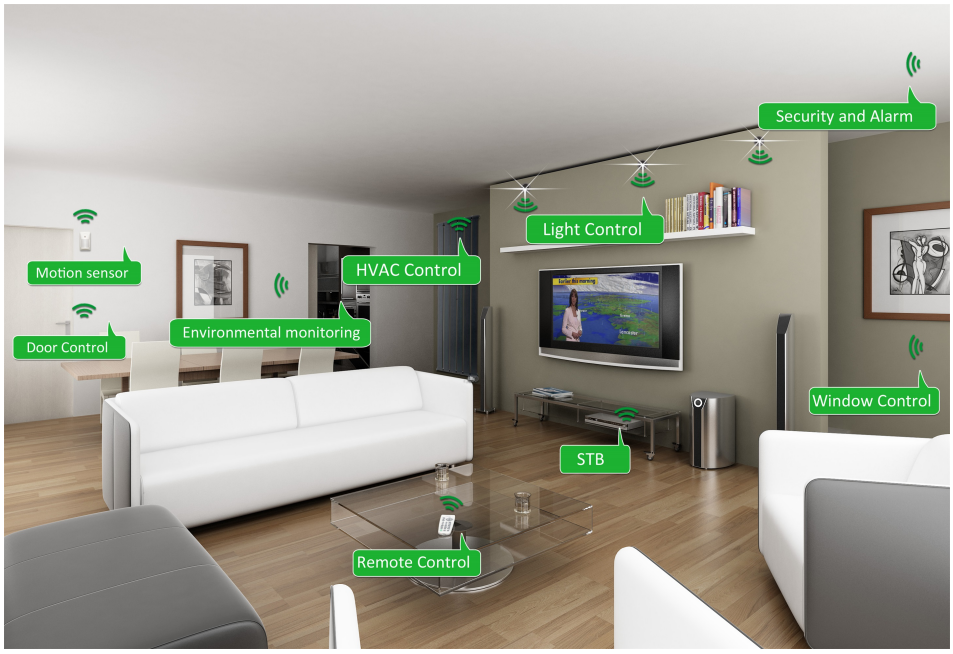


Figure 2.4: IoT sensor network for smarthome.[3]

Nowadays, given the development of wireless connections, low cost and low latency connections are available, which significantly raises interest of the Internet of Things (IoT) [36–38]. IoT allows physical devices and everyday objects to join in the Internet. These things can communicate and interact with others over the Internet, and can be remotely monitored and controlled. IoT has been used in many field, such as industrial factory, transportation. As for quantified self applications, IoT brings the possibility of involving various kinds of sensors into the Internet, then the sensors can cooperate together to enrich the functionality of analyzing our daily life. For example, in a smart home as shown in Figure 2.4, the IoT collects and analyzes the data from the host and the house. By knowing the motion, health and emotion status of the host, IoT can control the environment of the house to fit the demands, such as switching off the light when the host goes to sleep, adjusting the temperature through air conditioner when the host feels hot or cold. As a matter of fact, IoT is not a kind of sensor, on contrary it provides a platform for various sensors and bring them together to accomplish complex tasks.

2.2 Applications of quantified self

After introducing the sensors that can be utilized for quantified self, now we are going to describe what kind of quantified self applications could be developed based on these sensors. To our knowledge, the applications can be mainly divided into four categories: (i), activity recognition and analysis; (ii), robotics; (iii), localization and navigation; (iv), health monitoring and caring. In each category, many applications are developed on various sensors.

2.2.1 Activity recognition and analysis

In this kind of application, the sensors need to measure the data related to human motion. Joshi *et al.* proposed EMG-based method to collect leg muscle signals and recognize the walking and stair ascent activities. In [6–8], the authors used IMU sensors placed on the lower limb to detect activity and gait phase simultaneously. While Derawi *et al.* [9] used the IMU module in smartphone to detect gait cycle and activity. They found that introducing gait phase can improve the performance of recognizing the lower limb locomotion activity. Apart from the lower limb locomotion activity recognition, the gait alone can be applied to clinical research. Liparoti *et al.* [22] presented that gait analysis can be used in detecting gait abnormalities of disabled people. TunçAşuroğlu *et al.* [20] used a foot-worn pressure sensor to analyze the detected gait from Parkinson's patients, their method provided a way for understanding the disease evolution in the long term and simplify the detection of precipitous changes in gait on a daily basis in the short term. On the other side, the wearable sensors can also be used for the motion of upper limbs. Moschetti *et al.* [39] used IMU sensors placed on the finger and wrist to recognize daily activities, such as eating with hand, eating with fork, eating with spoon, drinking with glass, drinking with cup, brushing teeth, brushing hair, *etc.* An armband integrated with an IMU module and EMG sensors are used in [16, 17] to identify the gestures and daily activities, through combining the kinematics data and muscle signals.

On the other hand, activity of the entire body can also be tracked. For wearable sensors, people need to put on more than one sensors, sometimes even a sensor

network. Hsu *et al.* [40] used two IMU sensors that placed on wrist and ankle respectively for recognizing 10 domestic daily activities and 11 sport activities. Ayachi *et al.* [41] proposed an algorithm that automatically selected specific sensors for recognizing daily activities in a group of 17 IMU sensors placed on the body. Their algorithm does not use all the 17 sensors, but selects the most likely sensors related to the outgoing activity. The authors in [42] developed a wearable biosensor network towards daily life emotion recognition, the sensor network consists of electroencephalography (EEG) headset, temperature sensor, blood sensor, pulse sensor, *etc.* By analyzing the collected data, their work can evaluate four emotion status: happiness, horror, boredom and relaxation. Garcia-Ceja *et al.* [43] proposed a multi-view stacking method to fuse the data from heterogeneous types of sensors for activity recognition. Specifically, they utilized a wearable acoustic and inertial sensor system, the sound and acceleration are collected to recognize seven daily activities: mop floor, sweep floor, type on computer keyboard, brush teeth, wash hands, eat chips and watch television.

Using non-wearable sensors may have a global view for recognizing the activity of the entire body. Camera is widely used for analyzing activities through videos or pictures, and numerous researches have been investigated in activity recognition. Wang *et al.* [24] proposed a pedestrian recognition algorithm with multi-camera networks, the algorithm is very suitable for object recognition across cameras with disjoint views, especially for real-time long distance object tracking. In the survey [44], a lot of researches based on cameras were investigated for the recognition of abnormal human activity. The applications such as pedestrian recognition and abnormal activity recognition are quite useful in video surveillance system, homeland security, crowed analysis... A lot of efforts can be saved with the help of these applications. Gesture analysis is another important field and it can be accomplished by cameras. The researches conducted in [27, 28, 45] investigated how to use hand gesture recognition to interact with machines, such as computer, robotics. Gesture recognition is also of great interest in the translation between sign language and conventional language [46–48], which brings conveniences when communicating with the deaf or the mute. Sports analysis [49] is another potential application,

because non-wearable camera does not affect people when doing sports, which results in a non-biased motion. Another widely used non-wearable sensor for activity recognition is the wireless beacon. In the indoor scenario, wireless signal can be used for activity recognition [30–32]. A smart house is established in [36], which can recognize many sophisticated activities. A lot of works showed that wireless system is an effective way for activity recognition, however the limitations are obvious, as described in Section 2.1.2.

2.2.2 Robotics

The link between quantified self and robotics is how to use the collected quantified self data to communicate with and control the robotics. To be more precise, it normally can be divided into two categories, one is remote controlling of the robotics, another one is using robotics to assist people's life.

A potential application of remote control and quantified self is for remote surgery. Remote surgery can assist the doctor to conduct the surgery when he or she is even not in the operating room. The researches [50, 51] described the concept of remote surgery, as shown in Figure 2.5a. There is a master device that collects the motion data of the doctor and provide image scene to the doctor. And on the patient side, there is a slave that is controlled by the master, normally the slave is a kind of robotics that performs the operation. The communication between the master and slave is accomplished by satellite or optic fiber network with low latency time. The quantified self related part in remote surgery is on the doctor side, because the master needs to collect and analyze the motion of doctor precisely. Connolly *et al.* [14] developed an IMU-based electronic goniometric glove that detects the articulation movement of the hand, the system can measure how the hand and fingers move and then control the robotics move in the same way. While Tanaka *et al.* [15] integrated flex sensor and IMU into the glove to measure the motion of doctor's hand, in which the flex sensor is able to obtain how the finger is bent. The research [52] utilized EMG sensors to capture how the doctor's arm is moving and to control the robotic arm.

Another kind of application that uses robotic to assist people's life is generally

(a) Remote surgery [51].

(b) Rehabilitation or strength enhancing [53].

Figure 2.5: Quantified self application related to robotics.

in the field of haptic control, which means that the robotics and human have interaction with each other. Here, the robotics is not only an actuator that moves according to people, but also a sensor system that measures the motion of people. A common use of this kind of robotics is wearable exoskeleton, as shown in Figure 2.5b [53]. People can wear the device on the body to help them finish some tasks, such as rehabilitation exercises, labor work. The survey [54] reported the state-of-the-art researches of knee exoskeleton for gait rehabilitation. The researches showed that in order to assist human to conduct lower limb activities, the exoskeleton needs to know the exact activity and gait cycle of the wearer. This is also the same with the arm exoskeleton [55, 56]. Thus, the motion data of the body part is significantly important for this kind of application. Most of the time the sensors used for obtaining the data are IMUs and EMGs. Moreover, exoskeleton is also helpful for the people who need to conduct extensive labor work. The authors in [57] developed a lower limb exoskeleton to enhance people's strength of carrying

heavy cargo, the motion data is collected by EMGs placed on the shank and thigh.

2.2.3 Localization and navigation

At the beginning, when the devices and methods of localization and navigation were firstly proposed, they were designed for the aircraft and some military use. At that time, the used devices like GPS receiver and extremely expensive IMU are not very prevalent in the civilians. With the development of the technology in recent years, these sensors become cheaper and prevalent in commercial products, new technologies for localization and navigation also have been investigated. A most used application of localization and navigation is the digital map embedded in our smartphone. With the help of GPS and digital map, people are able to find their location and where are their places of interest, then the navigation provides the route for getting there through walking or driving car. Besides, in a smart watch, the combination of GPS and barometer obtains an accurate 3D position. This functionality enables the smart watch to evaluate the performance when people is climbing mountain and trail running, which is particularly very useful when in the competition. Also, the record of position from GPS guarantees people can trace back their route when they get lost.

Even though GPS is the most prevalent sensor in localization, it still has some disadvantages. For example, the satellite signals are very vulnerable to obstacles. Therefore, GPS is not suitable for indoor environment or the applications that need high accuracy in positioning. Other types of sensors-based methods were proposed for human localization. IMU sensor has been well investigated, since it measures the kinematics information of people. The researches [58–61] proposed the double integration-based pedestrian navigation system with foot-mounted IMU sensor. The idea of this kind of navigation system is integrating the measured acceleration twice to obtain the displacement of the sensor, with the help of some other methods of coordinate transferring and error elimination, the localization can be obtained from the displacement. On contrary, the authors in [62–64] used IMU sensors and proposed a kind of pedestrian dead reckoning (PDR) system that also provides the localization. PDR estimates the people's orientation and walking stride

length, then the displacement is obtained by accumulating the stride lengths.

Wireless signal is also possible for localization. Torteeka *et al.* [33] proposed a receive WIFI signal strength indication (RSSI)-based indoor positioning system, their method showed a reasonable positioning accuracy. Knauth *et al.* [65] proposed a large area positioning method that uses a smartphone, a data fusion method that fuses inertial signal and RSSI was applied into the system. It is able to identify the floor where people are standing, and also is applicable in large building. While the researchers in [34,35] proposed RSS-based methods that are applicable outdoor for a larger area, the systems utilizes mobile phone signal base station as the sources of RSS. In the work [11], the authors proposed a WIFI/GPS fusion method to reduce the localization error. The wireless signal-based localization method is very useful in the guidance and navigation application in museum and shopping mall.

As discussed above, the three types of sensors (GPS, IMU, wireless beacon) are utilized for localization under different scenarios. GPS is cheap and widely used, but it is only applicable in open area where the satellites signals are available. By contrast, IMU sensor and wireless beacon have no such restrictions, they are both applicable for indoor and outdoor environments. However, the error of IMU-based method increases with the time, and the RSS-based method has large error in outdoor environment. Therefore, many applications utilize a combination of different sensors to ensure the accuracy of localization, such as GPS/IMU [10] for pedestrian navigation, WIFI/GPS [11] to improve the outdoor localization accuracy in outdoor, WIFI/IMU [33] for indoor localization.

2.2.4 Health monitoring and caring

Since quantified self captures information generated by human body, a main field of applications is related to health, *i.e.* health monitoring and caring. Indeed, in Section 2.2.2, the exoskeleton used for rehabilitation is a kind of heath caring device. Apart from the exoskeleton, some other researches also investigated how to help injured people. Zhu *et al.* [66] used a single wrist-worn tri-axial accelerometer for rehabilitation exercise recognition, this method provides an evaluation of how the people is performing the exercises: good, average or bad. Chiang *et al.* [67]

proposed a method that monitors the knee range of motion with the help of two IMU sensors placed on the thigh and shank, this method mainly offers a continuous monitoring and objective assessment of knee range of motion recovery for total knee arthroplasty surgery. Khoury *et al.* [21] utilized pressure sensors placed in the shoes to detect gait cycles and to distinguish Parkinson's disease subjects from healthy subjects, or even to distinguish from the neurodegenerative diseases such as Amyotrophic Lateral Sclerosis and Huntington's disease. The author in [68–70] proposed EMG-based systems designed for upper limb rehabilitation, these systems can assist the injured people to fulfill daily activities and evaluate whether patients are performing the movement correctly.

Monitoring the health status of people is another field of quantified self applications. Unlike the rehabilitation use, health monitoring can be used for both healthy and unhealthy people. Zhou *et al.* [71] proposed a health care system for monitoring the status of the elderly who is living alone via a finger worn IMU sensor and a smartphone. The system can automatically collect and analyze the data of daily activities, then upload the evaluation result to the supporter (the organization who runs the system), finally the supporter is able to provide life support immediately. Fall detection helps people can get access to medical support when they suddenly fall down and lose the ability to move. Lee *et al.* [72] used inertial sensing-based to detect the fall and near-fall scenarios, whereas Agrawal *et al.* [73] utilized a video surveillance system to detect falls. The researches in [37, 38] used IoT sensor network or smart home to monitor the health status. Djelouat *et al.* [13] developed an ECG-based real-time heart health monitoring system using compressive sensing on a heterogeneous multicore edge-device, this system can be joined as a node in the network of IoT and provide heart related information.

2.2.5 Choice of the sensor

Now we have discussed about the sensors types and possible applications in quantified self. In this thesis, our research target is recognizing the lower limb locomotion activities. Then what kind of sensor type should be adopted is a question before starting the research, because selecting a proper sensor in the research can ensure

the research be proceeded well and obtain acceptable results. Figure 2.6 displays the relationships among the sensor types and possible applications, please remind that not all the sensors and possible applications are listed in the figure, but a dominated trends for most of the researches. As we can see, for each application many types of sensors can be utilized. Particularly for the activity recognition and analysis, there are five sensors that can be utilized: EMG, IMU, pressure sensor, wireless beacon and camera.

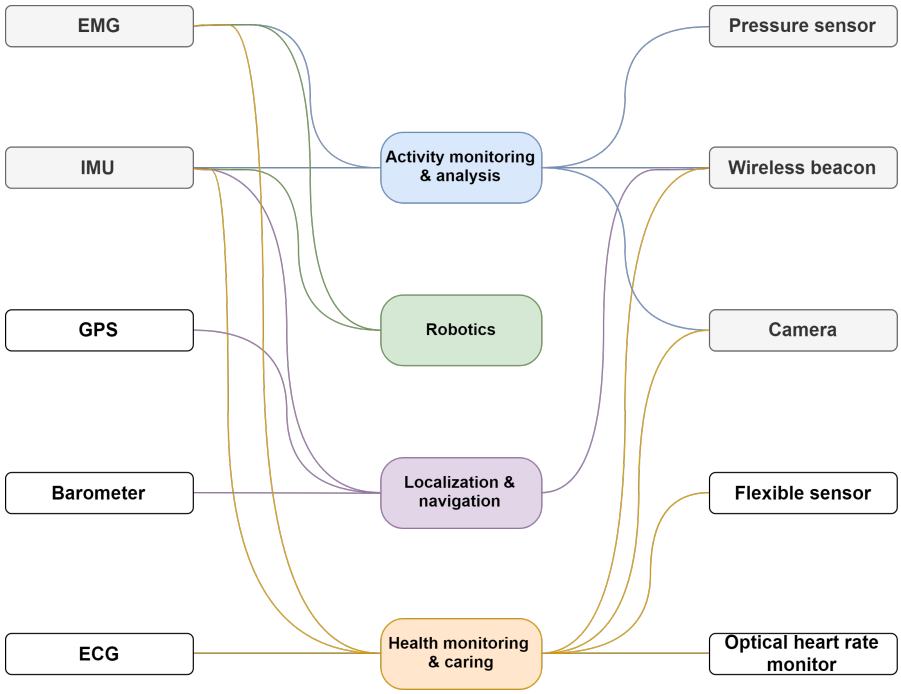


Figure 2.6: Appropriate sensors for possible quantified self applications.

Among the five sensors for activity recognition and analysis, EMG, IMU and pressure sensor are wearable sensor, on the other hand, wireless beacon and camera are non-wearable sensors. As described in Section 2.1, the major distinction between wearable and non-wearable sensors is that the former one can be worn on the body and be taken to any place where the wearer is going. Since the aim of our research is recognizing the locomotion activity, then it is impractical to confine people in a certain area. On contrary, when people perform the considered activities, it may appear in the buildings, on the road or somewhere else. Therefore, it is impossible to use non-wearable sensors to recognize the activity in such a large area. As for

the three wearable sensors, EMG needs to place electrodes on the muscles tightly to measure the electric signal. So it is not very convenient to use EMG sensors to recognize activities that take place in everyday. The pressure sensor may be a good way to recognize the locomotion activity, but it has some drawbacks when using it. Normally pressure sensor is made like insole and placed in the shoe, so it may cause inconvenience when people need to change the shoes to wear. Another problem is that pressure sensor is very expensive. By contrast, IMU sensor is easy to wear and is relatively cheap compared to the pressure sensor, that is why IMU sensor is widely used in the research of activity recognition. Therefore, in this thesis, we choose IMU sensor to collect motion data for the recognition of lower limb locomotion activity.

2.3 Algorithms in recent years

In this section, we focus on introducing the prevalent IMU-based algorithms that recognize activities. Recently, numerous researches have been investigated in the field of human activity recognition (HAR). The algorithms can generally classified into two dominant categories: (i), traditional classifier; (ii) Markov classifiers; (iii), deep learning methods.

In the first category, plenty of classifiers have been proposed and investigated. Parri *et al.* [74] proposed a fuzzy-logical classifier to identify lower limb locomotion mode, with the assistance of gait phases. The authors developed a lower limb wearable robot system that can help impaired people to perform locomotion activity. Martinez-Hernandez *et al.* [6] developed an adaptive Bayesian inference system using three sensors placed on leg to recognize walking on different road conditions, *i.e.* level-ground, ramp ascent and descent. They introduced gait phases and attempted to recognize activities and gait phases simultaneously. The high accuracy (99.87%) indicates that gait phases can significantly improve the accuracy for walking activity. Chen *et al.* [75] proposed a robust activity recognition algorithm based on principal component analysis (PCA) and on-line support vector machine (OSVM), the algorithm obtained a robust recognition accuracy over a smartphone dataset

collected in six different orientations. Safi *et al.* [76] proposed a system of three IMU sensors placed on the chest, right thigh and left ankle to detect twelve static and dynamic activities. They studied seven classifiers to recognize the activities, including k-Nearest Neighbour (kNN), support vector machine (SVM), random forest (RF), multi-layer perceptron (MLP), classification and regression tree (CART), naive Bayes (NB), and Gaussian mixture model (GMM). The results showed that kNN obtained the best performance with 96.26% recognition rate. In the work [77], the authors compared the performances among the classifiers of SVM, NB, kNN and kStar. Results showed that kNN and kStar obtained the highest accuracy while Naive Bayes obtained the lowest. The authors in [78] also used several features and classifiers to test their impacts on recognition accuracy, with the help of acceleration of 9 sensors on the body. They proved that detecting particular locomotion activities accurately needs specific features and classifiers. Their results showed that mean and standard deviation features provided the best accuracy out of all features evaluated by both kNN and ensemble methods, while spectral entropy produced the worst performance. They also concluded that data pre-processing has nearly no impact on recognition accuracy. Finally, Wen *et al.* [79] proposed an AdaBoost-based algorithm to adapt and refine the model at run-time, by automatically selecting the most discriminating features. Their results were tested on several smartphone data-sets and showed significant improvement in recognition performance.

In the second category, Markov models are widely adopted in the field of activity recognition. San-Segundo *et al.* [80] proposed a human activity recognition and segmentation system based on HMMs for recognizing and segmenting six activities: walking, sitting, standing, lying, stair ascent and descent. Six HMMs are accordingly trained in the system for recognizing and segmenting six activities Viterbi algorithm is then applied to determine which model fits the data the best, the recognized activity is finally the output of the Viterbi algorithm. The experimental results showed that the their proposed algorithm obtained an activity recognition error rate and a segmentation error rate lower than 10% and 0.5% respectively. Zhao *et al.* [81] proposed a 2-layer model to detect six gait phases of walking, the

algorithm used Neural Network (NN) to provide a pre-decision of gait phases to Hidden Markov Model (HMM), the final decision of gait phase from HMM obtained an accuracy of 98.11%. The limitation of this study is that only the activity of walking was considered, and the authors only tested their algorithm on straight forward walking, not free walking. In [82], hidden semi-Markov model (HSMM) and semi-Markov conditional random field (SMCRF) were applied to recognise human activity in smart home. The results showed that HSMM consistently outperformed HMM, while SMCRF obtained a similar result to CRF. However, because daily activities at home do not have stationary property, it is not practical to use a stationary transition matrix to represent the activity switches. Moreover, the authors only used Gaussian density to represent the class conditional observation density, which is quite limited for a complex scenario. The authors in [83] used HMM equipped with multiple regression (MHMMR) to automatically recognize activities and activity transitions in an unsupervised way, their method obtained higher accuracy than the other unsupervised methods, such as GMM and HMM. But in order to obtain the highest recognition accuracy, it is required to use the all the sensors placed on chest, thigh and ankle, which may probably reduce the flexibility of method.

In the third category, the deep learning methods are also very prevalent. Generally, this family of methods is more inclined for image processing, so it needs to convert sensor data to image description to support extraction of discriminative features. For example, Alsheikh *et al.* [84] proposed to use spectrogram representation to convert a three dimensional acceleration signal into a image like function of frequency and time. As reported in the survey [85], convolutional neural network (CNN) is an important category of discriminative deep learning model for HAR. The work [86] proposed convolutional recurrent neural network to recognize daily activity; their algorithm gained an improvement of 6% compared to the state-of-the-art works. While Ignatov [87] used CNN to recognize six human activities, including three lower limb activities, in real-time from accelerometers. This method obtained an accuracy higher than 97%, and achieved an overall accuracy of 82.76% on a cross-dataset experiment. Hassan *et al.* [88] proposed a smartphone-based

method for activity training and recognition via deep belief network (DBN), the results showed that DBN outperformed SVM and artificial neural network with an accuracy of 95.85% over twelve static and dynamic activities. Recently, as reported in [89], transfer learning and semantic approach have raised great research interest. Bao [90] and Rokni [91] used transfer learning to automatically construct model for newly added wearable sensors; they obtained an accuracy enhancement between 9.3%-10%. Ding *et al.* [92] tested three transfer learning methods —CNN, maximum mean discrepancy (MMD) and domain-adversarial neural network (DANN)— to dispose unlabeled data, and they found out that MMD work best on two public datasets. However, the recognition accuracy highly depends on the performance of labeling from source devices, thus it still requires a reliable method to obtain the activity label.

Some other methods can also be applied to the dedicated applications and obtain good results. Schneider *et al.* [93] proposed an automatic extraction and selection method of highly relevant features, the method was tested on eight datasets and obtained a general accuracy over 90%. Rezaie *et al.* [94] proposed a feedback controller framework to adapt sampling rate for better efficiency and higher accuracy. Dao *et al.* [95] introduced a man-in-loop decision architecture and data sharing among users, and gradually obtained a high accuracy. The works in [96] and [97] both proposed data fusion methods between motion data and location information for indoor scenario, the area of the house is divided and labeled into several parts, the motion data and location label together helped boosting the accuracy.

There are such a plenty of researches that have been investigated in the field of activity recognition. Due to the limit of the space, it is impossible to list all the methods in this thesis. But based on the introduction of the methods above, we are able to have a general view on these methods. As described in Chapter 1, the proposed algorithms in this thesis are based on the TMC, which belongs to the second algorithm category and is a kind of machine learning algorithm. It needs a carefully established model and is solved with mathematical methods, such as Bayes' theorem, EM algorithm, *etc.*

2.4 Public datasets

A lot of IMU-based public datasets of various activities have been published on-line by the researchers all over the world.

According to the published researches and the related IMU-based datasets, we have made a summary of the main features of these datasets, as shown in Table 2.1. In the Table, the listed datasets are frequently used in many researches. In a general view, datasets D1-D10 collected the 'normal activities' that we perform in everyday, including static activities (sitting, lying, standing...), locomotion activities (walking, running, climbing stairs...), sport exercises (cycling, plying soccer, rope jumping...), daily activities (watching TV, house cleaning, drinking, eating...). The sensors are mostly placed on the arms, legs or trunk of the subjects. Particularly, D5-D9 used IMU module embedded in smartphone(s) to collect the motion data, D10 not only used three IMUs, but also used a heart rate monitor, this enables the dataset available for estimating the intensity of the activity. The dataset D11 used nineteen IMUs placed on two arms and collected ten factory activities, such as open (close) hood, open (close) trunk, check steering wheel... Unlike the first eleven datasets in the Table, both the datasets D12 and D13 are for gait analysis. D12 is the largest IMU-based gait dataset in the world; it involved 744 subjects that perform walking on three road conditions: level ground, slope up and slope down. It used three IMUs placed on the center, left and right of the waist to collect the data. The large quantity of involved subjects has an age ranging from 2-78 years, and has a balanced gender ratio. D12 is targeted for the group of healthy people, whereas D13 collected the gait from ten subjects that had Parkinson's disease. In this dataset, three IMUs were placed on the hip and two legs, and the collected data were labeled in two categories: with or without the freezing of gait (FoG) during walking.

In this thesis, we have built our own dataset that use one IMU sensor placed on the right shoe, as shown in Figure 3.3. Ten subjects were enrolled in this dataset, and were asked to perform four lower limb locomotion activity, *i.e.* walking, running, stair ascent and stair descent. This dataset is specifically designed for on-line

recognition. So it is not only available for the algorithms that are proceeded in batch mode, but also is available for those on-line algorithm; details are given in Chapter 4.4.1. In addition, the dataset D1 in Table 2.1 is also utilized in this thesis, for validating the batch mode recognition performance of SemiTMC-GMM, details can be read in Chapter 5.4.1.

2.5 Conclusion

In this Chapter, the background of quantified self and our research target are described carefully. In Section 2.1, the types of sensors available for quantified self research are presented. The sensors are classified into two basic categories, *i.e.* wearable and non-wearable sensors, the different types of sensors in each category are introduced carefully, including what kind of signals they measure, how they are used... In Section 2.2, the possible applications of quantified self are classified into four categories: (i), activity recognition and analysis; (ii), robotics; (iii), localization and navigation; (iv), health monitoring and caring. In each kind of applications, researches that based on different sensors are discussed there. As in consequence, according to the sensor types and related applications that are discussed, we described how to choose sensor for a specific research. Particularly, the reason of why choosing IMU to collect data for this thesis is carefully discussed. The recent IMU-based algorithms for activity recognition are described in Section 2.3. In the last Section, some public IMU-based datasets are presented.

To sum up, in this thesis we concentrate on signals issued from IMU to investigate the lower limb locomotion activities. In the next Chapter, we will start with the presentation of a precise algorithm to detect gait cycle of the considered activities.

Table 2.1: Some public IMU-based datasets for the research of activity recognition and analysis.

ID	Name	Sensor(s) placement	Subjects number	Activity number and type	Description
D1 [98]	SDA	5 IMUs: torso, right arm, left arm, right leg, left leg	8	**19 daily activities and sport exercises**: sitting, standing, lying on back and on right side, ascending and descending stairs, standing in an elevator still, moving around in an elevator, walking in a parking lot, walking on a treadmill, running on a treadmill, exercising on a stepper, exercising on a cross trainer, cycling on an exercise bike in horizontal and vertical positions, rowing, jumping, playing basketball	Each activity was performed for 5 minutes, sampling rate is 25HZ.
D2 [99]	OPPORTUNITY	19 sensors: 7 IMUs and 12 3D acceleration sensors	4	**more than 18 daily living activities**: lying on the deckchair, get up, open and close door, open and close fridge, open and close dishwasher, open and close drawer, clean table, drink from cup, toggle switch, groom, prepare coffee, drink coffee, prepare sandwich, eat sandwich, clean up, break...	Sampling rate is 32Hz. The activities of the user in the scenario are annotated on different levels: "modes of locomotion" classes; low-level actions relating 13 actions to 23 objects; 17 mid-level gesture classes; and 5 high-level activity classes
D3 [100]	USC-HAD	one IMU placed on front right rip	14	**12 daily living activities**: walking forward, walking left, walking right, walking upstairs, walking downstairs, running forward, jumping, sitting, standing, sleeping, elevator up, elevator down	Sampling rate is 100Hz. each subject was asked to perform 5 trials for each activity on different days at various indoor and outdoor locations. On average, it took 6 hours for each subject to complete the whole data collection procedure.
D4 [101]	SHO	5 IMUs: right jeans pocket, left jeans pocket, belt position towards the right leg, right upper arm, right wrist	10	**7 daily activities**: walking, running, sitting, standing, jogging, biking, walking upstairs and walking downstairs	Sampling rate is 50Hz. Each activity was performed for 3-4 minutes by the subjects.
D5 [102]	UCI Smartphone	one IMU in a smartphone on the waist	30	**6 daily activities**: walking, stair ascent, stair descent, sitting, standing, laying	Sampling rate is 50Hz. The obtained dataset has been randomly partitioned into two sets, where 70% of the volunteers was selected for generating the training data and 30% the test data.
D6 [103]	ActiveMiles	5 smartphones, placement not fixed	10	**7 daily activities**: running, walking, cycling, casual movement, public transportation, idle, standing	Sampling rate is 50-200Hz. The dataset contains unconstrained real world human activity, there are no limitations on where the smartphone is located, i.e. pocket, bag, or held in the hand.

Table 2.1: Some public IMU-based datasets for the research of activity recognition and analysis.

ID	Name	Sensor(s) placement	Subjects number	Activity number and type	Description
D7 [104]	WISDM v1.1	one smartphone placed in front trouser pocket	29	**6 daily activities**: walking, jogging, upstairs, downstairs, sitting, standing	Sampling rate is 20Hz. This dataset contains data collected through controlled, laboratory conditions.
D8 [105]	WISDM v2.0	one smartphone placed in front trouser pocket	59	**6 daily activities**: walking, jogging, stairs, sitting, standing, lying down	Sampling rate is 20Hz. This dataset contains real world data.
D9 [106]	SHL	4 smartphones placed on torso, in backpack, in hand, in front trouser pocket	3	**8 static and locomotion activities**: still, walking, running, cycling, driving car, on the bus, on the train, on the subway	Sampling rate is 100Hz. It contains 750 hours of labelled locomotion data. The data was collected in a real life scenario.
D10 [107]	PAMAP2	3 IMUs: wrist, chest, ankle. And a heart rate monitor.	9	**18 physical activities**: lying, sitting, standing, walking, running, cycling, nordic walking, watching TV, computer work, car driving, stair ascent, stair descent, vacuum cleaning, ironing, folding laundry, house cleaning, playing soccer, rope jumping	IMU sampling rate is 100Hz, heart monitor sampling rate is 9Hz. Can be used for activity recognition and intensity estimation.
D11 [108]	Skoda	19 sensors: 10 nodes on the right arm and 9 on the left arm	1	**10 factory activities**: write on notepad, open hood, close hood, check gap door, open door, check steering wheel, open and close trunk, close both doors, close doors, check trunks	Sampling rate is 96Hz. Each activity was performed 19 times.
D12 [109]	world's largest IMU-based gait database	3 IMUs, at the left, center and right of waist	744	**3 walking activities**: level walking, walking slope up, walking slope down	Sampling rate is 100Hz. It is mainly collected for the use of gait analysis.
D13 [110]	Daphnet Gait	3 IMUs: hip and two legs	10	**2 activities**: with freeze event and without freeze event	Sampling rate is 64Hz. It aims at the research of Parkinson's disease patients that experience freezing of gait (FoG) during walking tasks.

Chapter 3

Gait cycle detection using HMM

Genrally, the two legs in many lower limb locomotion activities repeat a specific cycle. For example, the most general activity that people perform everyday—walking—has the foot from attaching to the ground to swinging in the air and then attaching to the ground again. This periodic pattern is called gait cycle and generally there are two common ways for segmenting each gait cycle:

(1) One simplest way is that one gait cycle is divided into two gait phases [111], namely stance phase and swing phase.

(2) Based on the two gait phases in the first approach, we can divide the gait cycle into more detailed phases. As introduced in [112], we utilize four gait phases in one gait cycle, *i.e.* stance, push-up, swing and step down.

The scheme of the second walking gait cycle characterization is shown in Figure 3.1. If taking the three phases of push-up, swing and step down as one single phase, it will then form the swing phase of the first gait cycle segmentation. These kinds of segmentation work for many other lower limb activities, such as running, stair ascent, stair descent, *etc.* The only difference relying on the duration and on the motion intensity of each gait phase.

In the last decade, many researches and applications have been investigated that involve gait analysis. In [60], we proposed a pedestrian navigation algorithm using a foot-mounted IMU sensor. The algorithm utilized a HMM model to detect zero velocity period (stance phases) for an extended Kalman filter to reduce the

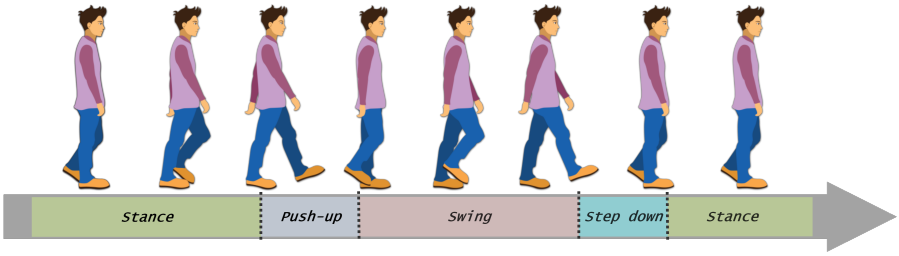


Figure 3.1: Right foot gait phases of walking cycle: push-up → swing → step down → stance. Similar gait cycle can be deduced for other activities, such as running, stair climbing.

navigation error. Ertop *et al.* [113] developed a robotic gait trainer based on real-time gait detection, and the precise gait assistance and resistance helped injured people doing walking in a fluid environment. Gait can also be used for clinical evaluation in identifying and understanding gait problems. Mueske *et al.* [114] firstly studied how to use gait analysis on pathology identification and surgical recommendations for children with spina bifida. Zhao *et al.* [115] proposed an IMU sensor-based gait analysis system for the rehabilitation assessment of patients with gait disorders, the system can provide the differences of gait phases between the two feet, then make it possible to investigate the gait disorder.

Since gait analysis is so important for such a large number of applications, an HMM-based gait phase detection algorithm will be described in the remaining.

3.1 Left-to-Right HMC model

In an HMM, there are two stochastic processes: observation $\boldsymbol{Y}$ and hidden state $\boldsymbol{U}$, particularly when the hidden state is discrete, the model is also called as hidden Markov chain (HMC). Let us assume an HMC model with real-valued observations $\boldsymbol{Y} = \{\boldsymbol{Y}_1, \cdots \boldsymbol{Y}_N\}$, each $\boldsymbol{Y}_n \in \mathbb{R}^w$, where $1 \leq n \leq N$, and w is the observation dimension. Let the hidden state be modeled with $\boldsymbol{U} = \{U_1, \cdots, U_N\}$ where $U_n \in \Gamma = \{1, \cdots, \tau\}$, here τ equals to 4 if we consider the four gait phases.

Let a realization of U_n, $\boldsymbol{Y}_n$ be denoted by u_n and $\boldsymbol{y}_n$ respectively. For simplification, we denote the probability $p(U_n = u_n | \boldsymbol{Y}_1 = \boldsymbol{y}_1, \ldots, \boldsymbol{Y}_N = \boldsymbol{y}_N)$ by $p(u_n | \boldsymbol{y}_1^N)$ for example. Given an observed sequence $\boldsymbol{y}_1^N$, we can derive the probability with

the specific hidden state u_1^N:

$$p\left(u_1^N, \boldsymbol{y}_1^N\right) = p\left(u_1\right) p\left(\boldsymbol{y}_n | u_1\right) \prod_{n=1}^{N-1} p\left(u_{n+1} | u_n\right) p\left(\boldsymbol{y}_{n+1} | u_{n+1}\right). \tag{3.1}$$

The probability $p\left(u_{n+1}|u_n\right)$ is the hidden state transition probability, and $p\left(\boldsymbol{y}_n|u_n\right)$ is the observation probability density conditioned on u_n.

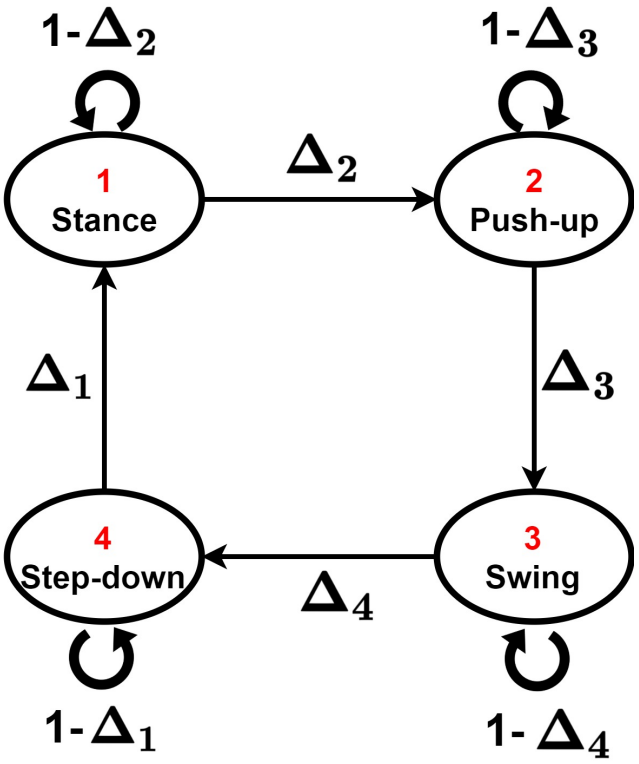


Figure 3.2: Transition order of gait phases, Δ_k are the transition probability that from previous state to the current state k.

As shown in Figure 3.1, the four phases switch one after the other and repeat periodically, thus we can introduce a Left-to-Right style state transition for $\boldsymbol{U}$. This kind of model is referred as LR-HMC. In Figure 3.2, the numbers $1, \ldots, 4$ represent stance, push-up, swing and step down respectively. Therefore, according to the transition graph, we get the following transition matrix.

$$\boldsymbol{A} = \begin{bmatrix} 1-\Delta_2 & \Delta_2 & 0 & 0 \\ 0 & 1-\Delta_3 & \Delta_3 & 0 \\ 0 & 0 & 1-\Delta_4 & \Delta_4 \\ \Delta_1 & 0 & 0 & 1-\Delta_1 \end{bmatrix} \tag{3.2}$$

In this chapter, the observation distribution conditioned on a specific hidden state k forms:

$$p\left(\boldsymbol{y}_n | u_n = k\right) \sim \mathcal{N}\left(\boldsymbol{\mu}_k, \Sigma_k\right), \tag{3.3}$$

where $k \in \Gamma$, $\mathcal{N}$ stands for the conventional Gaussian distribution with mean value $\boldsymbol{\mu}_k$ and covariance Σ_k. Thus, the probability of sequence $(u_1^N, \boldsymbol{y}_1^N)$ can be obtained through Equations (3.1) to (3.3).

3.2 Parameter estimation

Before starting the explanation, we need to introduce the parameters set first. As described above, the parameter set can be defined as $\theta = \{\zeta_k, \Delta_k, \boldsymbol{\mu}_k, \Sigma_k\}$, in which ζ_k is the initial probability of hidden state, $\boldsymbol{\mu}_k$ and Σ_k are the mean value and covariance of observation given $u_n = k$. Also, let denote a_{lk} is the l-th row and k-th column element in $\boldsymbol{A}$.

From previous section, it is now clear how the hidden state transfers and how to compute the observation probability. In this section, we will focus on how to obtain the filtering and smoothing probabilities, and applying parameter estimation based on the well-known Baum-Welch algorithm [7], also called as "forward-backward procedure". This algorithm is based on the Expectation-Maximization (EM) principle. The parameters are obtained through recursively performing the expectation and maximization steps, *i.e.* using the updated $\theta^{(i)}$ that obtained in the i-th EM iteration to calculate the two steps again to obtain $\theta^{(i+1)}$, until it reaches a maximum iteration number. The initial θ does not need the ground truth of hidden state at first, therefore Baum-Welch algorithm is an unsupervised learning method.

3.2.1 Expectation step

Firstly, the forward probabilities $\alpha_n(u_n)$ and backward probabilities $\beta_n(u_n)$ are defined as follows

$$\begin{aligned} \alpha_n(u_n) &= p\left(u_n, \boldsymbol{y}_n | \boldsymbol{y}_1^{n-1}\right), \\ \beta_n(u_n) &= p\left(\boldsymbol{y}_{n+1}^{N} | u_n, \boldsymbol{y}_n\right). \end{aligned} \tag{3.4}$$

Then, it needs to recursively calculate the forward probability $\alpha_n(u_n)$ and backward probability $\beta_n(u_n)$ for all the $n \in \{1, \ldots, N\}$, by Equations (3.5) and (3.6)

$$\begin{aligned} \alpha_1(u_n) &= p\left(u_1, y_1\right) = p\left(u_1\right) p\left(y_1 | u_1\right), \\ \alpha_{n+1}(u_n) &= p\left(u_{n+1}, \boldsymbol{y}_{n+1} | \boldsymbol{y}_1^{n}\right) \\ &= \sum_{u_n \in \Gamma} \left\{ p\left(u_n, \boldsymbol{y}_n | \boldsymbol{y}_1^{n-1}\right) p\left(u_{n+1}, \boldsymbol{y}_{n+1} | u_n, \boldsymbol{y}_n, \boldsymbol{y}_1^{n-1}\right) \right\} \\ &= \sum_{u_n \in \Gamma} \left\{ \alpha_n(u_n) p\left(u_{n+1}, \boldsymbol{y}_{n+1} | u_n, \boldsymbol{y}_n\right) \right\}, \end{aligned} \tag{3.5}$$

$$\begin{aligned} \beta_N(u_n) &= 1, \\ \beta_n(u_n) &= p\left(\boldsymbol{y}_{n+1}^{N} | u_n, \boldsymbol{y}_n\right) \\ &= \sum_{u_{n+1} \in \Gamma} \left\{ p\left(u_{n+1}, \boldsymbol{y}_{n+1} | u_n, \boldsymbol{y}_n\right) p\left(y_{n+2}^{N} | u_{n+1}, \boldsymbol{y}_{n+1}, u_n, \boldsymbol{y}_n\right) \right\} \\ &= \sum_{u_{n+1} \in \Gamma} \left\{ p\left(u_{n+1}, \boldsymbol{y}_{n+1} | u_n, \boldsymbol{y}_n\right) p\left(y_{n+2}^{N} | u_{n+1}, \boldsymbol{y}_{n+1}\right) \right\} \\ &= \sum_{u_{n+1} \in \Gamma} \left\{ p\left(u_{n+1}, \boldsymbol{y}_{n+1} | u_n, \boldsymbol{y}_n\right) \beta_{n+1}(u_{n+1}) \right\}. \end{aligned} \tag{3.6}$$

Because $\boldsymbol{y}_n$ and $\boldsymbol{y}_{n+1}$ are independent if u_n is given, then

$$p\left(u_{n+1}, \boldsymbol{y}_{n+1} | u_n, \boldsymbol{y}_n\right) = p\left(u_{n+1} | u_n\right) p\left(\boldsymbol{y}_{n+1} | u_{n+1}\right). \tag{3.7}$$

From forward and backward probabilities, the so-called filtering probabilities $p(u_n|\boldsymbol{y}_1^n)$, smoothing probabilities $p(u_n|\boldsymbol{y}_1^N)$ and the joint probabilities $p(u_n, u_{n+1}|\boldsymbol{y}_1^N)$ can be obtained by

$$p\left(u_n | \boldsymbol{y}_1^n\right) = \frac{p\left(u_n, \boldsymbol{y}_n | \boldsymbol{y}_1^{n-1}\right)}{\sum_{u_n \in \Gamma} p\left(u_n, \boldsymbol{y}_n | \boldsymbol{y}_1^{n-1}\right)} = \frac{\alpha_n(u_n)}{\sum_{u_n \in \Gamma} \alpha_n(u_n)}, \tag{3.8}$$

$$
\begin{aligned}
p\left(u_n|\boldsymbol{y}_1^N\right) &= p\left(u_n|\boldsymbol{y}_n, \boldsymbol{y}_1^{n-1}, \boldsymbol{y}_{n+1}^N\right) \\
&= \frac{p\left(u_n, \boldsymbol{y}_n, \boldsymbol{y}_{n+1}^N|\boldsymbol{y}_1^{n-1}\right)}{\sum_{u_n\in\Gamma} p\left(u_n, \boldsymbol{y}_n, \boldsymbol{y}_{n+1}^N|\boldsymbol{y}_1^{n-1}\right)} \\
&= \frac{p\left(u_n, \boldsymbol{y}_n|\boldsymbol{y}_1^{n-1}\right) p\left(\boldsymbol{y}_{n+1}^N|u_n, \boldsymbol{y}_n, \cancel{\boldsymbol{y}_1^{n-1}}\right)}{\sum_{u_n\in\Gamma} p\left(u_n, \boldsymbol{y}_n|\boldsymbol{y}_1^{n-1}\right) p\left(\boldsymbol{y}_{n+1}^N|u_n, \boldsymbol{y}_n, \cancel{\boldsymbol{y}_1^{n-1}}\right)} \\
&= \frac{\alpha_n(u_n)\beta_n(u_n)}{\sum_{u_n\in\Gamma} \alpha_n(u_n)\beta_n(u_n)},
\end{aligned}
\tag{3.9}
$$

$$
\begin{aligned}
p(u_n, u_{n+1}|\boldsymbol{y}_1^N) &= \frac{p\left(u_n|\boldsymbol{y}_1^{n-1}\right)\cdot p\left(u_{n+1}, \boldsymbol{y}_{n+1}|u_n, \boldsymbol{y}_n\right)\cdot p\left(\boldsymbol{y}_{n+1}^N|u_{n+1}, \boldsymbol{y}_{n+1}\right)}{\sum_{u_n,u_{n+1}\in\Gamma}\left\{p\left(u_n|\boldsymbol{y}_1^{n-1}\right)\cdot p\left(u_{n+1}, \boldsymbol{y}_{n+1}|u_n, \boldsymbol{y}_n\right)\cdot p\left(\boldsymbol{y}_{n+1}^N|u_{n+1}, \boldsymbol{y}_{n+1}\right)\right\}} \\
&= \frac{\alpha_n(u_n)\cdot p\left(u_{n+1}, \boldsymbol{y}_{n+1}|u_n, \boldsymbol{y}_n\right)\cdot\beta_{n+1}(u_{n+1})}{\sum_{u_n,u_{n+1}\in\Gamma}\left\{\alpha_n(u_n)\cdot p\left(u_{n+1}, \boldsymbol{y}_{n+1}|u_n, \boldsymbol{y}_{n+1}\right)\cdot\beta_{n+1}(u_{n+1})\right\}}.
\end{aligned}
\tag{3.10}
$$

We next note that $\gamma_n(u_n) = p\left(u_n|\boldsymbol{y}_1^N\right)$ and $\xi_n(u_n, u_{n+1}) = p(u_n, u_{n+1}|\boldsymbol{y}_1^N)$. Estimated hidden states can then be recovered through the Bayesian MPM (Maximum Posterior Mode) criterion:

$$
\hat{u}_n = \arg\max_{u_n\in\Gamma}\gamma_n(u_n). \tag{3.11}
$$

3.2.2 Maximization step

From the expectation results of $\gamma_n(u_n)$ and $\xi_n(u_n,)$, the parameters can be updated by maximizing the likelihood function of $(\boldsymbol{U}_1^N, \boldsymbol{Y}_1^N)$, the proof is given in the Appendix A, here we briefly list re-estimation formula of parameters in θ:

$$
\zeta_k = \frac{\gamma_1(u_1 = k)}{\sum_{k=1}^{\tau}\gamma_1(u_1 = k)}, \tag{3.12}
$$

$$
\boldsymbol{\mu}_k = \frac{\sum_{n=1}^{N}\gamma_n(u_n = k)\boldsymbol{y}_n}{\sum_{n=1}^{N}\gamma_n(u_n = k)}, \tag{3.13}
$$

$$\Sigma_k = \frac{\sum_{n=1}^{N} \gamma_n(u_n = k)\,(\boldsymbol{y}_n - \boldsymbol{\mu}_k)\,(\boldsymbol{y}_n - \boldsymbol{\mu}_k)^{\mathsf{T}}}{\sum_{n-1}^{N} \gamma_n(u_n = k)}, \tag{3.14}$$

$$a_{lk} = \frac{\sum_{n=1}^{N-1} \xi_n(u_n = l, u_{n+1} = k)}{\sum_{n=1}^{N-1} \gamma_n(u_n = l)}. \tag{3.15}$$

Particularly, for the transition matrix of LR-HMC described in Equation (3.2), Δ_1 to Δ_4 can be obtained by

$$\Delta_1 = \frac{\sum_{n-1}^{N-1} \xi_n(4,1)}{\sum_{n=1}^{N-1} [\xi_n(4,4) + \xi_n(4,1)]}, \tag{3.16}$$

$$\Delta_k = \frac{\sum_{n-1}^{N-1} \xi_n(k-1,k)}{\sum_{n=1}^{N-1} [\xi_n(k-1,k-1) + \xi_n(k-1,k)]} \quad , k = 2,3,4. \tag{3.17}$$

The EM algorithm can be accomplished through recursively performing the Expectation step and Maximization step, the procedures are shown in the pseudo-code below:

Table 3.1: Procedures of EM algorithm.

1:	$\boldsymbol{\theta}^{(0)} \longleftarrow$ initial parameters;
	$\boldsymbol{y}_1^N \longleftarrow$ observations;
	iterMax $\longleftarrow$ max EM iteration;
2:	**For** $i = 1$ to *iterMax*
2.1:	$\alpha_n(u_n)$ =**Forward**$\big(\boldsymbol{\theta}^{(i-1)}, \boldsymbol{y}_1^N\big)$;
2.2:	$\beta_n(u_n)$ =**Backward**$\big(\boldsymbol{\theta}^{(i-1)}, \boldsymbol{y}_1^N\big)$;
2.3:	$[\gamma_n(u_n),\xi_n(u_n, u_{n+1})]$=**Expectation**$\big(\alpha_n(u_n), \beta_n(u_n), \boldsymbol{\theta}^{(i-1)}\big)$;
2.4:	$\boldsymbol{\theta}^{(i)}$=**Maximization**$\big(\gamma_n(u_n), \xi_n(u_n, u_{n+1}), \boldsymbol{y}_1^N\big)$;
	End For
3:	$\boldsymbol{\theta} = \boldsymbol{\theta}^{(i)}$;
4:	Output $\boldsymbol{\theta}$.

3.3 Gait phase detection using LR-HMC

This section describes the results of LR-HMC detecting gait phases for each activity.

3.3.1 Experiment setups

We utilize a foot-mounted IMU sensor[1] to obtain the foot acceleration and angular rate. As shown in Figure 3.3, the sensor is small enough, it can transfer the data through wireless connection or store the data in a SD card for the analysis after experiment. The coordinate of the sensor is also shown in the Figure, with $\mathcal{X}$-axis of the sensor pointing to right, $\mathcal{Y}$-axis pointing ahead and $\mathcal{Z}$-axis up. Thus, the acceleration in the 3 axes are referred as $a^{\mathcal{X}}$, $a^{\mathcal{Y}}$ and $a^{\mathcal{Z}}$, the angular rate are referred as $\omega^{\mathcal{X}}$, $\omega^{\mathcal{Y}}$ and $\omega^{\mathcal{Z}}$. Given the orientation of the sensor fixed on the shoe and the normal motion of the ankle, $a^{\mathcal{Y}}$ and $a^{\mathcal{Z}}$ seem to be the most informative in acceleration, $\omega^{\mathcal{X}}$ seems to be the most informative in angular rate. Throughout this thesis, the Shimmer3 sensor sampling rate is set to $100Hz$, the scale of accelerometer is set to $\pm 8G$, the scale of gyroscope is set to $\pm 1000deg/s$.

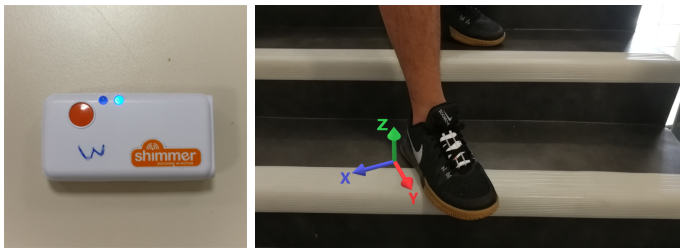
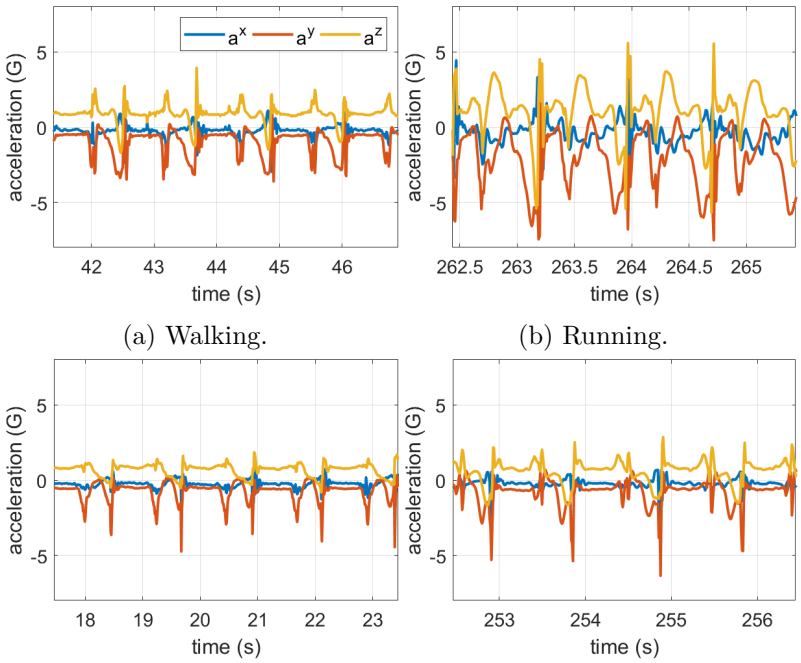


Figure 3.3: Left: Shimmer3 IMU sensor. Right: the placement of the sensor on right shoe.

Four lower limb activities —walking, running, stair ascent and stair descent— are considered here for the gait phase detection. One subject was asked to perform these four activities, each activity kept a constant speed, *i.e.* speed did not change during the activity. The Figure 3.4 shows the sensor acceleration acquired from the four activities, and Figure 3.5 shows the angular rate. Through the figures, we can see that the signals vary differently among the four activities, and they all have

[1]Shimmer3 GSR+, more details at the manufacturer's site http://www.shimmersensing.com/images/uploads/docs/ConsensysPRO_Spec_Sheet_v1.1.0.pdf

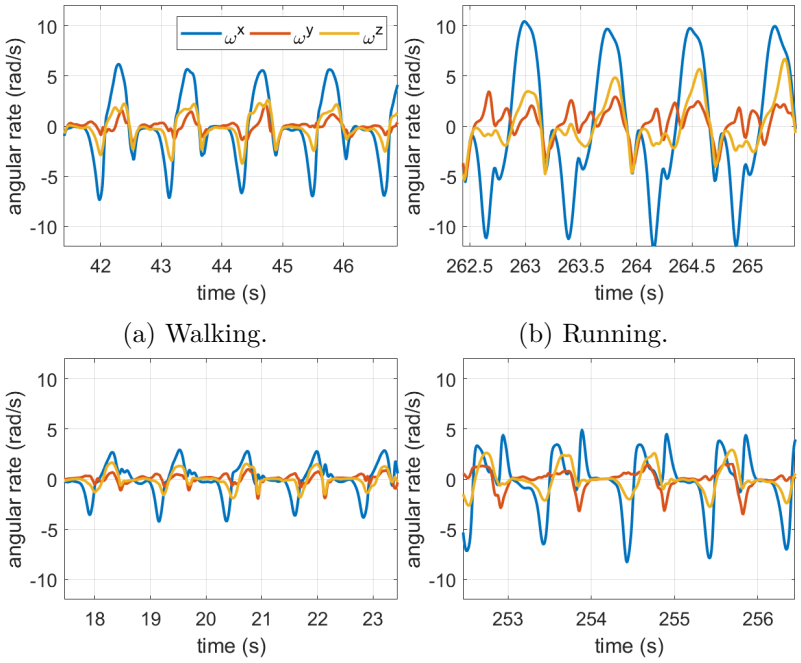
periodic cycles, which indicate the gait cycle.



(a) Walking. (b) Running.

(c) Stair ascent. (d) Stair descent.

Figure 3.4: Sensor acceleration of four activities for each axis, *i.e.* $a^{\mathcal{X}}$, $a^{\mathcal{Y}}$, $a^{\mathcal{Z}}$.



(a) Walking. (b) Running.

(c) Stair ascent. (d) Stair descent.

Figure 3.5: Sensor angular rates of four activities for each axis, *i.e.* $\omega^{\mathcal{X}}$, $\omega^{\mathcal{Y}}$, $\omega^{\mathcal{Z}}$.

3.3.2 Model initialization

Before using Baum-Welch algorithm to learn parameters, the parameter initialization $\theta^{(0)}$ should firstly be obtained. So we need a roughly segmented gait phases that can be used for parameter learning. An initialization of gait phases of one specific activity is developed for the LR-HMC, with the following procedure:

(1) Set filtering cut-off frequency f and stance threshold h for the specific activity. These two values are obtained through experience.

(2) Use a Butterworth low-pass filter to filter the norm of angular rate according to f, then segment the filtered angular rate by h. All the periods below h will be regarded as stance gait phase, all the periods above will be regarded as non-stance phases.

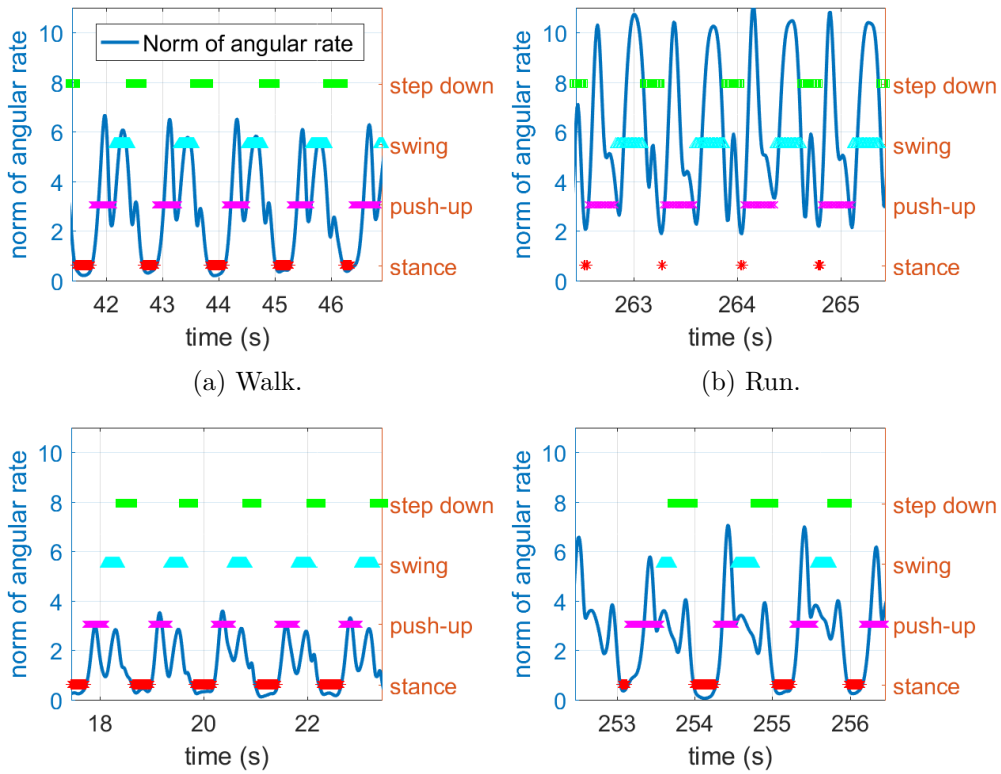
(3) In each period of non-stance phase, the three peaks represent the three non-stance gait phases, *i.e.* push-up, swing, and step-down. Thus, the three non-stance gait phases can be initialized according to the peaks. Here, we simply use four indices: the start and end indices of the non-stance phase, the middle index between the first and second peaks, the middle index between the second and third peaks, to obtain the three non-stance gait phases.

(4) Once gait phases have been segmented for all the activities, $p(u_n, u_{n+1})$ and $p(\boldsymbol{y}_n|u_n)$ can be easily obtained by using trivial empirical estimators, ζ and $\boldsymbol{A}$ can be derived from $p(u_n, u_{n+1})$.

An excerpt of the initialization procedure is given in Fig. 3.6. The f and h of each activity are set according to Table 3.2. The initialization is only a rough segmentation, it does not work for every gait cycle because bias may sometime appear. Its usage is to initialize θ as $\theta^{(0)}$, then use this $\theta^{(0)}$ to learn the model, so it does not need to segment all the gait cycles.

3.3.3 Feature extraction

Feature extraction provides LR-HMC the needed observations. In [78], the authors proved that detecting particular locomotion activities accurately needs specific fea-



(a) Walk. (b) Run.

(c) Stair ascent. (d) Stair descent.

Figure 3.6: The initial hidden state obtained by the initialization process. In each sub-figures, blue line is the filtered angular rate norm. The red, purple, cyan and green represent stance, push-up, swing and step-down, respectively.

Table 3.2: Filter cut-off frequency f and stance threshold h used for each activity.

	Walk	**Run**	**Stair ascent**	**Stair descent**
$f(Hz)$	5.00	9.00	4.50	6.00
$h(rad/s)$	0.52	1.92	0.52	0.52

tures and classifiers, and the results showed that mean and standard deviation features on sliding window provided the best accuracy out of all features evaluated by both KNN and ensemble methods, while spectral entropy produced the worst performance. As a matter of fact, many kinds of features in time domain or frequency domain can be extracted from the original signals, such as mean, standard deviation, maximum, root mean square, peak count, magnitude, spectral energy or entropy, mean/median/peak frequency. The combination of mean and standard

deviation may depict the main characters in time domain, *i.e.* the direction (positive or negative), amplitude, variation. Thus, using these two features is enough to represent the characters in time domain. As for the frequency domain, it needs a large quantity data to obtain accurate features to depict frequency, which is not appropriate in our case. Then, based on their conclusion and our own tests, two kinds of features are selected for observations: the sliding mean μ_n and sliding standard deviation σ_n of the acceleration and angular rate along the 3 sensor axes. The calculations *w.r.t.* acceleration is done according to

$$\begin{aligned} m_n^i &= \frac{1}{W} \sum_{j=n-W+1}^{n} a_j^i, \\ \sigma_n^i &= \sqrt{\frac{1}{W} \sum_{j=n-W+1}^{n} \left(a_j^i - m_n^i\right)^2}, \end{aligned} \tag{3.18}$$

where $i \in \{\mathcal{X}, \mathcal{Y}, \mathcal{Z}\}$ represents the axis of sensor, W is the length of the sliding window, it is set to 15 according to our experience. Same features can be extracted from angular rate. Hence, the observation dimension is $w = 12$.

3.3.4 Gait detection

After feature extraction and parameter initialization, we are able to learn model parameters through the Baum-Welch algorithm, by using Equations (3.4) to (3.16). At the mean time, the detected gait phases can be obtained by Equation (3.11).

Table 3.3 shows the learned transition matrix of each activity. We can see that transition matrices are different from each other activity, due to the duration of each gait phase of the activities are different. Since the gait phases last a period of time, which means that hidden states keep the same in most of the time, so Δ_1 to Δ_4 are very small. Another reason is because the experiment of each activity started from standing still on the ground. Thus, the learned initial probability ζ_k of all the activities are the same, *i.e.* $\zeta_1 = 1$ and $\zeta_2 = \zeta_3 = \zeta_4 = 0$. Table 3.4 shows the estimated mean value of Gaussian $p(\boldsymbol{y}_n|u_n)$, we can see that $\boldsymbol{\mu}$ varies differently in one activity.

Table 3.3: Gait phase transition matrices $\boldsymbol{A}$ for the four activities.

		Stance	Push-up	Swing	Step down
Walk	**Stance**	0.9712	0.0288	0	0
	Push-up	0	0.9400	0.0600	0
	Swing	0	0	0.9614	0.0386
	Step down	0.0307	0	0	0.9693
Run	**Stance**	0.9267	0.0733	0	0
	Push-up	0	0.9600	0.0400	0
	Swing	0	0	0.9518	0.0482
	Step down	0.0620	0	0	0.9380
Stair ascent	**Stance**	0.9800	0.0200	0	0
	Push-up	0	0.9577	0.0423	0
	Swing	0	0	0.9649	0.0351
	Step down	0.0404	0	0	0.9596
Stair descent	**Stance**	0.9785	0.0215	0	0
	Push-up	0	0.9570	0.0430	0
	Swing	0	0	0.9574	0.0426
	Step down	0.0376	0	0	0.9624

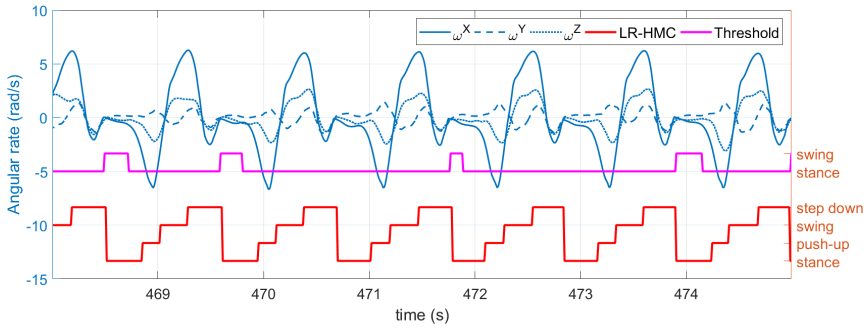
In order to evaluate the segmentation performance, the threshold method proposed in [23] is compared to LR-HMC; the used parameters for the threshold method are tuned for getting the best result. As shown in Figure 3.7, the threshold method can only detect stance and swing gait phases, it also fails several times to detect stance phase for walking, running and stair ascent. By contrast, LR-HMC can detect more complex gait phases and obtains a very regular gait pattern. Besides, the parameters in LR-HMC can be automatically learned by the unsupervised Baum-Welch algorithm, while the threshold method need to manually set the parameter for each activity.

3.4 Conclusion

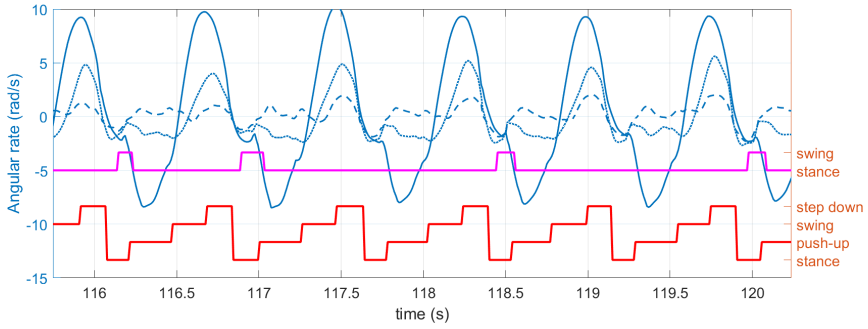
This chapter aims to use HMM-based algorithm to detect gait phases related to lower limb locomotion activity. We developed a LR-HMC model that can detect the gait phases from a single activity, with the help of a foot-mounted IMU sensor.

Table 3.4: The estimated observation mean value $\boldsymbol{\mu}$.

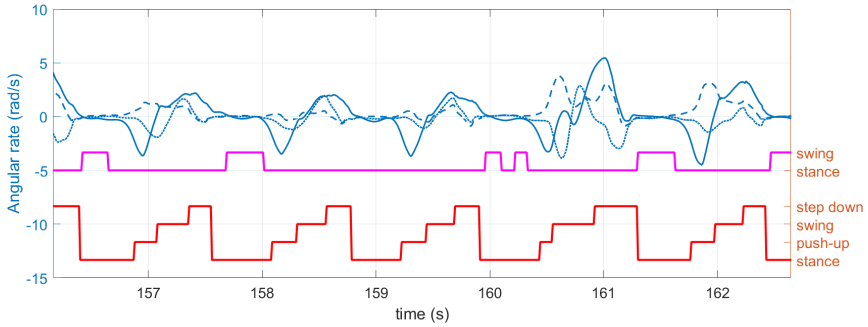
		$m^{a^{\mathcal{X}}}$	$m^{a^{\mathcal{Y}}}$	$m^{a^{\mathcal{Z}}}$	$m^{\omega^{\mathcal{X}}}$	$m^{\omega^{\mathcal{Y}}}$	$m^{\omega^{\mathcal{Z}}}$	$\sigma^{a^{\mathcal{X}}}$	$\sigma^{a^{\mathcal{Y}}}$	$\sigma^{a^{\mathcal{Z}}}$	$\sigma^{\omega^{\mathcal{X}}}$	$\sigma^{\omega^{\mathcal{Y}}}$	$\sigma^{\omega^{\mathcal{Z}}}$
	Unit		G			rad/s			G			rad/s	
Walk	**Stance**	-0.2217	-0.5625	0.8105	-0.7777	0.0964	-0.2505	0.0439	0.0275	0.0347	0.3924	0.0753	0.1232
	Push-up	-0.5226	-1.1675	1.2658	-5.0785	0.2940	-1.4701	0.2445	0.6647	0.3167	2.3172	0.4982	0.9151
	Swing	-0.4363	-0.8731	1.0483	3.1140	-0.2940	0.9646	0.2401	0.6895	0.3011	1.7990	0.6599	0.7377
	Step down	-0.1164	-1.8999	0.3175	0.9786	-0.1393	0.1929	0.5616	0.9297	0.9664	2.0745	1.0511	1.1955
Run	**Stance**	0.1734	-1.9422	0.6153	-1.7112	-1.1276	-1.6002	1.4181	1.7405	1.8330	3.3251	2.0547	2.3372
	Push-up	-0.5445	-1.7143	1.5241	-6.2619	0.4481	-1.2722	0.3567	1.0388	0.8827	3.2423	1.4501	0.7973
	Swing	-0.8796	-1.0768	1.5824	4.5127	0.4177	0.4637	0.3004	0.8747	0.5624	2.9095	1.4829	1.5516
	Step down	0.0898	-4.1186	-0.4314	5.4869	0.5797	2.6378	0.8274	1.4476	1.6998	3.3420	1.3882	3.0614
Stair ascent	**Stance**	-0.2367	-0.5191	0.8257	-0.2141	0.0450	-0.0565	0.0269	0.0264	0.0314	0.2101	0.0661	0.0644
	Push-up	-0.4302	-1.1738	0.8988	-2.2263	0.9128	-0.7827	0.2367	0.5724	0.1691	1.3848	0.7734	0.4858
	Swing	-0.1100	-0.3766	0.5295	1.4974	0.9026	0.1996	0.1918	0.4254	0.1753	0.7241	0.4366	0.6715
	Step down	-0.0341	-1.1921	0.4192	1.5317	0.1134	-0.3883	0.4411	0.8798	0.4859	0.9411	0.9132	0.9819
Stair descent	**Stance**	-0.2142	-0.5731	0.8038	-0.0783	-0.0717	-0.1418	0.0614	0.0495	0.0749	0.4554	0.1106	0.1612
	Push-up	-0.2784	-0.4992	0.9533	-3.2698	-0.7351	-0.9506	0.1912	0.3266	0.3910	2.8384	0.7372	0.7354
	Swing	-0.5498	-0.8657	0.2220	1.8539	-0.4892	1.6907	0.2068	0.4115	0.2593	1.0418	0.5241	0.8322
	Step down	-0.1525	-1.3664	0.1087	0.9208	-1.0753	0.5447	0.7902	0.7918	0.7358	1.8564	1.8043	0.9878



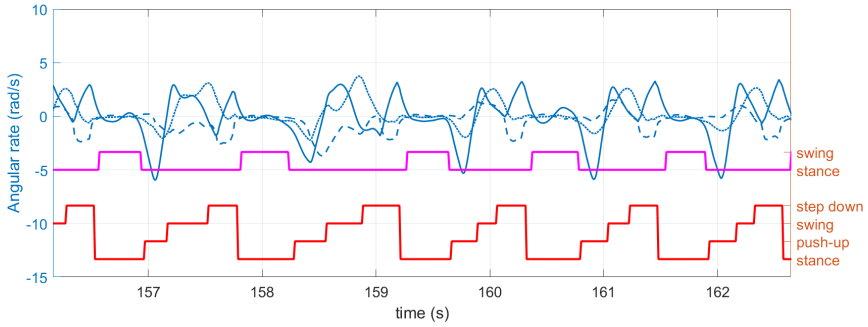
(a) Walk.



(b) Run.



(c) Stair ascent.



(d) Stair descent.

Figure 3.7: The detected gait cycle by LR-HMC and threshold method.

In the LR-HMC model, the gait phase acts as the hidden state, a specific left-right state transition graph is applied. The learned model and restored hidden state show that the proposed model is applicable for gait detection.

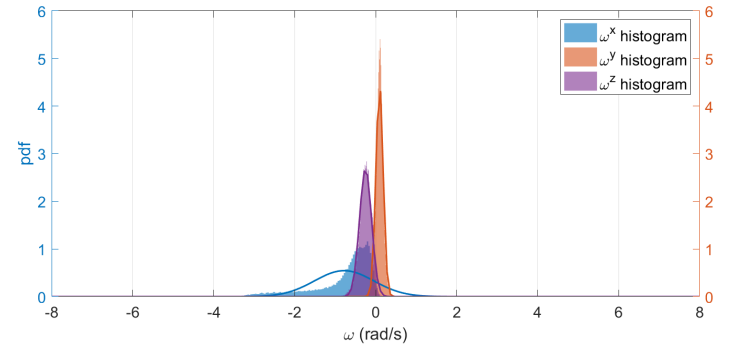
However, there are some limitations. First of all, the LR-HMC is a stationary model, the learned parameters can only work for one pattern of a single activity. That means the model is not able to handle more complex cases, such as various activity speeds, suitable for many people, change of activity pattern caused by different road conditions, *etc.* Therefore, it is very limited to use this model in a realistic application. Secondly, the model only suits for one activity, *i.e.* it can not automatically identify the activity. In the next two chapters, we will focus on dealing with these issues.

Chapter 4

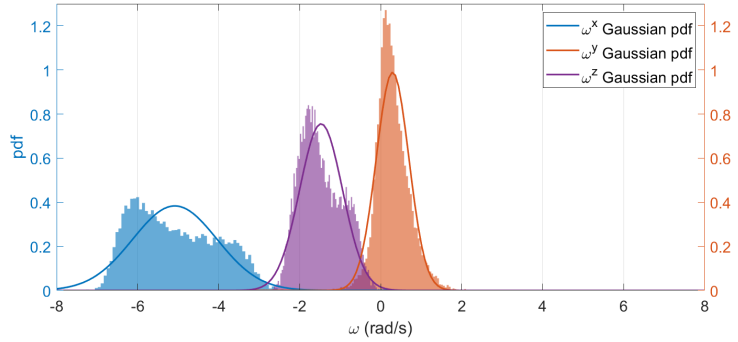
Adaptive on-line recognition using non-parametric TMC

This chapter details a non-parametric triplet Markov chain (TMC) model designed for adaptive on-line recognition of lower limb locomotion activity. The proposed algorithm is able to recognize different activities and detect gait phases simultaneously. Besides, it should consider a wide range of activity pattern from different healthy people, and be robust to different speeds as well.

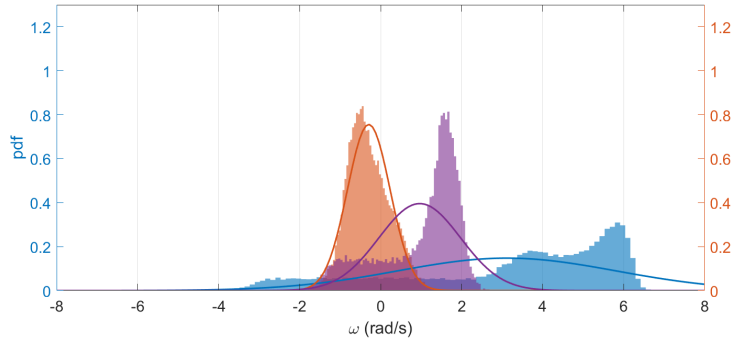
In the Chapter 3, the LR-HMC is utilized for detecting the gait phases of different lower limb locomotion activities. Figure 4.1 shows the angular rate histogram of each gait phase and the estimated Gaussian probability density, *w.r.t.* the activity of walking. Seeing the histograms, we can observe that their shapes are not strictly Gaussian, especially for the histograms of $\omega^{\mathcal{X}}$, which is prevalent for angular rate. Therefore, the estimated Gaussian density cannot match the real class-conditional observation density perfectly. On the other hand, for the swing and step down gait phases, apparently their histograms are extremely different from each other, but the estimated Gaussian densities look like similarly. It is because that the mean value $\boldsymbol{\mu}$ are very close, even though their densities are totally different. Please note that this is only one activity performed by one person at a constant speed. However our goal is to recognize activity in a much more complex situations, then such a simple density approximation is not realistic. A straight forward idea is using histograms themselves rather than Gaussian density to represent the class-conditional observation density. So, following this idea, we propose a non-parametric TMC, referred as TMC-HIST, to fulfill the recognition task. Based on TMC-HIST, an adaptive on-line recognition algorithm is then established.



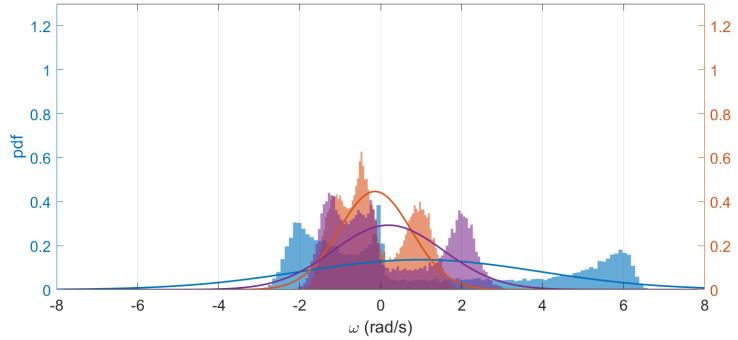
(a) Stance.



(b) Push-up.



(c) Swing.



(d) Step down.

Figure 4.1: The histogram of the sliding mean of angular rate, and the estimated marginal Gaussian probability density, *w.r.t.* the activity of walking.

4.1 TMC-HIST

In previous LR-HMC, only the hidden state $\boldsymbol{U}$ is considered because we just need to detect the gait phase. However, since now we need to recover both the activity and gait phase at the same time, a TMC is introduced here to model the motion infrastructure. Consider three stochastic processes. The first one is $\boldsymbol{X} = (X_1, \cdots, X_N)$, where $X_n \in \Lambda = \{1, \cdots, r\}$, $n \in \{1, \cdots, N\}$. X_n represents the desired activity, r is the number of activities that are considered. The second one is $\boldsymbol{U} \in \Gamma = \{1, \cdots, \tau\}$, which represents the gait phase, it is the same as the one in LR-HMC that described in Section 3.1. The real-valued process $\boldsymbol{Y}_n \in \mathbb{R}^w$ represents the observation. Therefore, total number of possible combinations of the couple (X_n, U_n) is $r \times \tau$.

Denote $\boldsymbol{V} = (\boldsymbol{X}, \boldsymbol{U})$ and $\boldsymbol{T} = (\boldsymbol{V}, \boldsymbol{Y})$. Then $\boldsymbol{T}$ is said to be a TMC if it is Markovian [116]. TMCs are strictly more general than pairwise Markov chains (PMCs) [117], which are themselves more general than HMMs, see for example [118] for detailed explanations. In a general TMC, none of the processes $\boldsymbol{X}, \boldsymbol{Y}, (\boldsymbol{X}, \boldsymbol{U}), (\boldsymbol{X}, \boldsymbol{Y}), (\boldsymbol{U}, \boldsymbol{Y})$ are necessarily Markovian [119], but the conventional algorithms of the parameter learning still work for TMCs. The Baum–Welch algorithm (but not necessarily the Viterbi algorithm) applies in triplet Markov models, so Bayesian MPM criterion can be used to recover both $\boldsymbol{X}$ and $\boldsymbol{U}$, *i.e.* activities and gait phases, from $\boldsymbol{Y}$ only, once parameters of the model are known.

Let the realizations of X_n, U_n and $\boldsymbol{Y}_n$ be denoted by x_n, u_n and $\boldsymbol{y}_n$ respectively, so $\boldsymbol{v}_n = (x_n, u_n)$, $\boldsymbol{t}_n = (\boldsymbol{v}_n, \boldsymbol{y}_n)$. Also, for simplification, we will denote the probability $p(X_n = x_n, U_n = u_n | \boldsymbol{Y}_1 = \boldsymbol{y}_1, \ldots, \boldsymbol{Y}_N = \boldsymbol{y}_N)$ by $p(x_n, u_n | \boldsymbol{y}_1^N)$ for example. The dependency graph of the specific TMC suited for our application is shown in Figure 4.2. The transition probability of $\boldsymbol{T}$, $p(\boldsymbol{t}_{n+1} | \boldsymbol{t}_n) = p(\boldsymbol{v}_{n+1}, \boldsymbol{y}_{n+1} | \boldsymbol{v}_n, \boldsymbol{y}_n)$, is simplified to

$$p(\boldsymbol{t}_{n+1} | \boldsymbol{t}_n) = p(x_{n+1}, u_{n+1} | x_n, u_n)\, p(\boldsymbol{y}_{n+1} | x_{n+1}, u_{n+1}), \tag{4.1}$$

which provides the process $\boldsymbol{T} = (\boldsymbol{V}, \boldsymbol{Y})$, with $\boldsymbol{V} = (\boldsymbol{X}, \boldsymbol{U})$, the structure of a classical HMC. The first term $p(x_{n+1}, u_{n+1} | x_n, u_n)$ is the state transition probability, the

dimension of the transition matrix being $(r \times \tau) \times (r \times \tau)$. For activity recognition, the state can only transfer from one gait phase to the next gait phase within the same activity, or, from stance phase of one activity to push-up phase of another activity, as shown in Figure 4.3. Because the number of activities is $r = 4$, so the dimension of the state joint probability matrix $p(\boldsymbol{v}_n, \boldsymbol{v}_{n+1})$ is 16. Therefore, there are only 44 non-zero entries in the total 256. Process $\boldsymbol{V}$ has the shape of a cyclic left-right Markov chain similar as the one in LR-HMC.

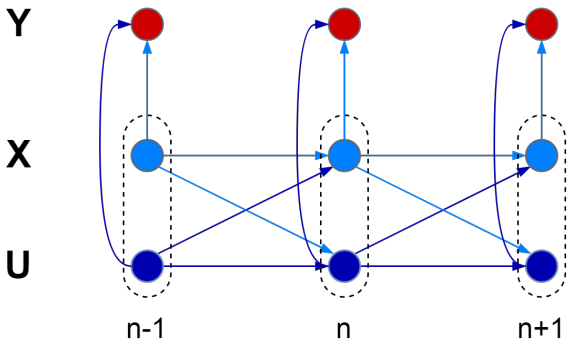


Figure 4.2: TMC dependency graph for activity recognition.

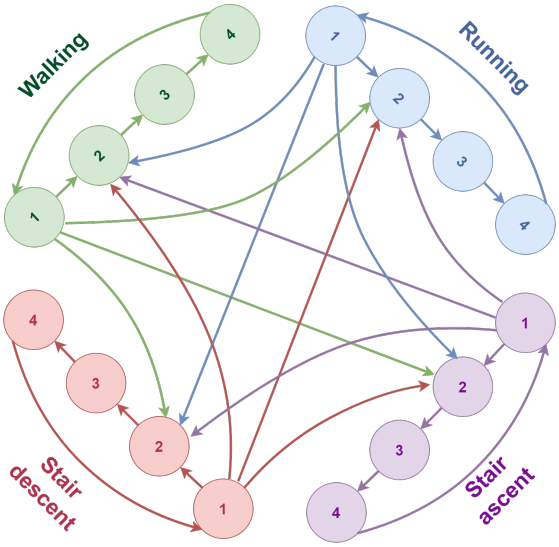


Figure 4.3: State transition graph of the TMC-based activity recognition algorithm. The values (1,2,3,4) represent the stance, push-up, swing and step down respectively, for the four gait phases.

The second term in Equation (4.1) is the probability of observing $\boldsymbol{y}_n$ conditionally to each state. Most of the time, this kind of density is modeled by using Gaussian densities, whereas in this work we propose a non-parametric modeling method by using histograms to represent $p(\boldsymbol{y}_n|\boldsymbol{v}_n)$ for its best ability to fit users' steps, whatever their size. Because $\boldsymbol{y}_n$ is a w-dimensional vector, then w histograms can be derived from $\boldsymbol{y}_n$ for each $\boldsymbol{v}_n$, thus the total number of histograms in TMC-HIST is $16 \times w$. In LR-HMC, $p(\boldsymbol{y}_n|u_n)$ is obtained by the Gaussian density function given the $\boldsymbol{\mu}_k$ and Σ_k, while in TMC-HIST, $p(\boldsymbol{y}_n|\boldsymbol{v}_n)$ is provided by the histograms *w.r.t.* specific $\boldsymbol{v}_n$. Since there are w histograms for each $\boldsymbol{v}_n$, then $p(\boldsymbol{y}_n|\boldsymbol{v}_n)$ equals to the product of all the w histogram densities.

It is sure that the three axes of the sensor are correlated in the movement, while the applied histograms represent the marginal probability density of each axis. Using a high dimensional histogram would probably suit the motion more closely, but it may require too much data to form a reliable density and drastically increase the computation time.

4.2 Parameter estimation

The parameter set to be estimated is $\theta = \{\zeta_k, a_{lk}, B_k\}$,$l, k \in \Lambda \times \Gamma$, where a_{lk} is the l-th row and k-th column element in the transition matrix A, B is the histograms. The calculation of forward-backward procedure for TMC-HIST is very similar with the one of LR-HMC, as described in Section 3.2. Replace u_n with $\boldsymbol{v}_n$ in the Equations (3.4) to (3.6), the forward probability $\alpha_n(\boldsymbol{v}_n)$ and backward probability $\beta_n(\boldsymbol{v}_n)$ are then obtained. Denote the filtering probability $p(\boldsymbol{v}_n|\boldsymbol{y}_1^n)$, smoothing probabilities $\gamma_n(\boldsymbol{v}_n) = p(\boldsymbol{v}_n|\boldsymbol{y}_1^N)$ and $\xi_n(\boldsymbol{v}_n, \boldsymbol{v}_{n+1}) = p(\boldsymbol{v}_n, \boldsymbol{v}_{n+1}|\boldsymbol{y}_1^N)$, they can be similarly calculated by

$$p\left(\boldsymbol{v}_n|\boldsymbol{y}_1^n\right) = \frac{\alpha_n(\boldsymbol{v}_n)}{\sum\limits_{\boldsymbol{v}_n \in \Lambda \times \Gamma} \alpha_n(\boldsymbol{v}_n)}, \tag{4.2}$$

$$\gamma_n(\boldsymbol{v}_n) = \frac{\alpha_n(\boldsymbol{v}_n)\beta_n(\boldsymbol{v}_n)}{\sum\limits_{\boldsymbol{v}_n \in \Lambda \times \Gamma} \alpha_n(\boldsymbol{v}_n)\beta_n(\boldsymbol{v}_n)}, \tag{4.3}$$

$$\xi_n(\boldsymbol{v}_n, \boldsymbol{v}_{n+1}) = \frac{\alpha_n(\boldsymbol{v}_n)\; p\left(\boldsymbol{v}_{n+1}, \boldsymbol{y}_{n+1} | \boldsymbol{v}_n, \boldsymbol{y}_n\right)\; \beta_{n+1}(\boldsymbol{v}_{n+1})}{\sum\limits_{\boldsymbol{v}_n, \boldsymbol{v}_{n+1} \in \Lambda \times \Gamma} \alpha_n(\boldsymbol{v}_n)\; p\left(\boldsymbol{v}_{n+1}, \boldsymbol{y}_{n+1} | \boldsymbol{v}_n, \boldsymbol{y}_n\right)\; \beta_{n+1}(\boldsymbol{v}_{n+1})}. \tag{4.4}$$

The estimated hidden state is then recovered by

$$\hat{\boldsymbol{v}}_n = \arg \max_{\boldsymbol{v}_n \in \Lambda \times \Gamma} p\left(\boldsymbol{v}_n | \boldsymbol{y}_1^N\right). \tag{4.5}$$

These calculations can be performed once the parameter set θ is known. The well-known Expectation-Maximization (EM) principle is generally applied for learning parameters because it provides the exact re-estimation formula for parameters under Gaussian distribution assumptions. As we deal with non-parametric histograms, the expectation step can be acomplished by Equations (4.3) and (4.4). But unfortunately it is not possible to retrieve θ through the maximization step that described in Section 3.2.2, which utilizes partial differential to seek the maximum point. Hence, we make use of another unsupervised learning method called "Iterative Conditional Estimation" (ICE) [117, 120], which is applicable in a wide range of situations. Here, we simply recall the ICE procedure:

(1) Based on the known activity x_n, initialize hidden state u_n by the same method described in Section 3.3.2, then the initial ζ_k, a_{lk} and B_k can be very easily acquired;

(2) Compute the forward-backward algorithm using current parameters, and compute the state transition probability conditioned on observations:

$$p\left(\boldsymbol{v}_{n+1} | \boldsymbol{v}_n, \boldsymbol{y}_1^N\right) = \frac{\xi_n(\boldsymbol{v}_n, \boldsymbol{v}_{n+1})}{\gamma_n(\boldsymbol{v}_n)}; \tag{4.6}$$

(3) Simulate a realization of state sequence $\tilde{\boldsymbol{v}} = (\tilde{\boldsymbol{v}}_1, \ldots, \tilde{\boldsymbol{v}}_N)$ by using $p(v_1 | \boldsymbol{y}_1^N)$ and $p(\boldsymbol{v}_{n+1} | \boldsymbol{v}_n, \boldsymbol{y}_1^N)$ given in Equations (4.3) and (4.6);

(4) Update ζ_k, A and B_k according to $\tilde{\boldsymbol{v}}$ and $\boldsymbol{y}_1^N$;

(5) Stop when the number of ICE iterations reaches a predefined maximum value, 100 for example, else go back to step (2).

The ICE parameter estimation algorithm is an unsupervised learning method, even though we use the ground truth of activity in the first step. In fact, the initialization is a kind of semi-unsupervised method, but ICE applied here is exactly unsupervised. Then, using the learned model we can recover activity $\hat{x}_n$ and gait phase $\hat{u}_n$ in the batch mode (or off-line), by adopting the smoothing described in Equation (4.3), then obtain

$$\begin{aligned} \hat{x}_n &= \arg \max_{x_n \in \Lambda} \left\{ \sum_{u_n \in \Gamma} p\left(x_n, u_n | \boldsymbol{y}_1^N\right) \right\}, \\ \hat{u}_n &= \arg \max_{u_n \in \Gamma} \left\{ \sum_{x_n \in \Lambda} p\left(x_n, u_n | \boldsymbol{y}_1^N\right) \right\}. \end{aligned} \tag{4.7}$$

4.3 Adaptive on-line recognition

In the previous section, the model learning is relied on the assumption that TMC-HIST is stationary, *i.e.* once the model is learned, the estimated parameters are not able to change. This limits the application of recognition and may probably reduce the recognition accuracy. Because in our daily life, the pattern of our activity may change according to the activity speed and road condition. Besides, the pattern is of course differed among the people, according to their gender, height, stride length, *etc.* Therefore, it is unrealistic to generate a universal model that can fits for the situations and people as much as possible. While, an alternative way to cope with this problem is making the model can adapt itself according to the situation at run time. So, this section will explain how to make TMC-HIST work adaptively on-line.

The diagram of the adaptive on-line algorithm based on TMC-HIST for lower limb locomotion activities recognition is displayed in Figure 4.4. The entire algorithm is composed of four stages: (i) model training, (ii) data acquisition and preprocessing, (iii) complete gait detection, (iv) final decision and posterior update. The (i) stage will be described firstly, (ii) and (iii) stages will then be presented together, (iv) stage will finally be described.

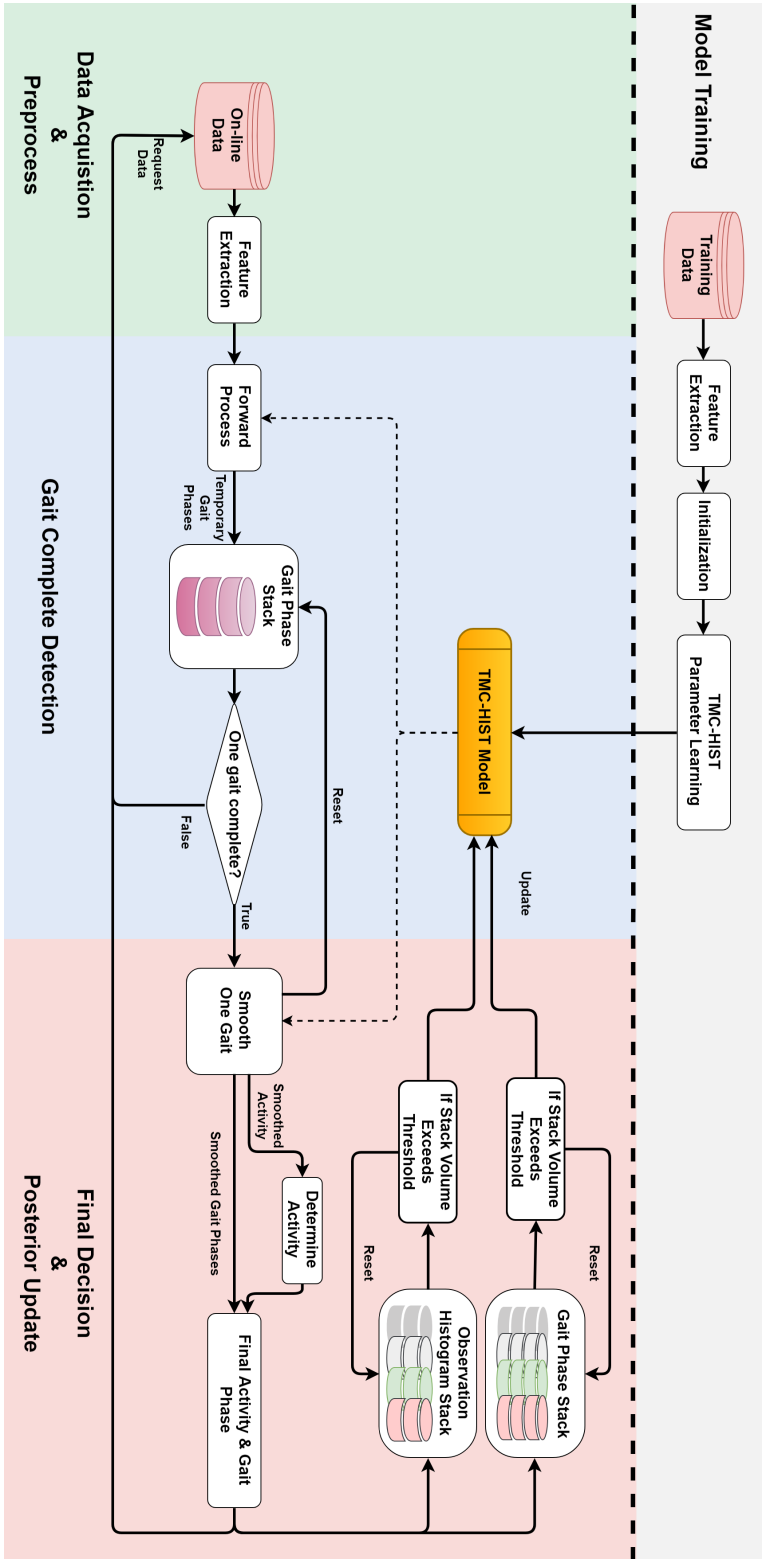


Figure 4.4: Diagram of the adaptive on-line recognition algorithm.

4.3.1 Model Training

The TMC-HIST model needs to be trained before being used on-line, this trained model will act as an initial model for the on-line recognition. It guarantees that the algorithm is initialized in a status that will allow a roughly correct classification. The trained model needs to detect all kinds of activities and activity switches, thus the training data should contain the four activities and all possible activity switches. While it should be noticed that on-line data does not have this restriction.

As similar in the Section 3.3.3, the extracted features are sliding mean $\boldsymbol{m}$ and standard deviation σ of acceleration and angular rate. Let the extracted features be the observation and use ICE algorithm depicted in Section 4.2 to obtain the learned model.

4.3.2 On-line Data Acquisition and Complete Gait Detection

The data is fed to the on-line algorithm once it arrives. Then feature extraction is performed based on the new data and the stored ones.. The data is accumulated during the required time to obtain a complete gait with the sequence: push-up $\rightarrow$ swing $\rightarrow$ step down $\rightarrow$ stance. In order to decide whether a gait cycle is complete, a one-step forward process of TMC-HIST model is firstly conducted to estimate current gait phase, which derives from the marginal probability of $p(x_n, u_n|\boldsymbol{y}_1^n)$ over x_n, using MPM method

$$\tilde{u}_n = \arg \max_{u_n \in \Gamma} \left\{ \sum_{x_n \in \Lambda} p\left(x_n, u_n|\boldsymbol{y}_1^n\right) \right\}. \tag{4.8}$$

Then, the current gait phase and the features are stored in a stack, which contains all the gait phases and corresponding features of the current gait cycle. Afterward, a decision is made to verify whether an entire gait is complete or not. If the stack is shorter than a threshold t_{min} or does not contain a complete sequence of gait phases, the algorithm will return and wait for another new data to come in. The current forward result $p(x_n, u_n|\boldsymbol{y}_1^n)$, obtained from Equation (4.2), will be stored for the use of the forward process of the next sampling time. The value of t_{min} is set to

0.5s based on the fastest speed of gait cycle among the four activities. Otherwise, it means that an entire gait is completed and all the features stored will be sent to the last stage.

4.3.3 Final Decision and Posterior Update

The activity and gait phases in one detected gait cycle are denoted by x'_n and u'_n, and N' is the number of samples for the corresponding gait cycle. This stage proceeds only when one gait is completed and contains two steps: 1) final decision for activity and gait phases, 2) posterior update of the joint probability $p(\boldsymbol{v}'_n, \boldsymbol{v}'_{n+1}|\boldsymbol{y}_1^{N'})$ and the histograms that represent $p(\boldsymbol{y}_n|x_n, u_n)$, if necessary. First, the features from the previous stage are smoothed by the forward-backward process to get $p(\boldsymbol{v}'_n|\boldsymbol{y}_1^{N'})$, then the smoothed activities and gait phases are obtained using MPM criterion again:

$$\begin{aligned}\widehat{x'_n} &= \arg\max_{x'_n\in\Lambda}\left\{\sum_{u'_n\in\Gamma} p\left(x'_n, u'_n|\boldsymbol{y}_1^{N'}\right)\right\},\\ \widehat{u'_n} &= \arg\max_{u'_n\in\Gamma}\left\{\sum_{x'_n\in\Lambda} p\left(x'_n, u'_n|\boldsymbol{y}_1^{N'}\right)\right\}.\end{aligned} \tag{4.9}$$

The final decision of gait phases is $\widehat{u'_n}$, for each sampling time. However, there is only one possible activity for each gait cycle, so the final decision of activity $\widehat{x'}$ depends on the most frequent estimated activity among $\widehat{x'_n}$:

$$\widehat{x'} = \arg\max_{k\in\Lambda}\left\{\sum_{n=1}^{N'} \mathbb{1}_{\widehat{x'_n}=k}\right\}, \tag{4.10}$$

where $\mathbb{1}$ is a boolean-valued function which takes value 1 when it satisfies the condition, or takes 0 if not. Then, all the $\widehat{x'_n}$ within one gait cycle are set to the final decision $\widehat{x'}$.

Based on the final decision of activity and gait phases, we can accumulate the features into the stacks for posterior update. Since the activity number is r and gait phase number is $\tau = 4$, so in the gait phase stacks, there are r different 4×4 matrices for each activity and each matrix is a counter relative to the $p(u'_n, u'_{n+1}|\boldsymbol{y}_1^{N'})$, which is a marginal probability of $p(\boldsymbol{v}'_n, \boldsymbol{v}'_{n+1}|\boldsymbol{y}_1^{N'})$ over all x'_n and x'_{n+1}. The sequence

of gait phases from final decision are divided into $N' - 1$ pairs of (u'_n, u'_{n+1}), the number of each (u'_n, u'_{n+1}) case is then accumulated into the matrix according to $\widehat{x'}$. As the name suggests, the observation histograms stacks contain $m \times q$ different stacks and accumulate the observations, *i.e.* features, according to $\widehat{x'}$ and $\widehat{u'_n}$.

Two thresholds η_{gait} and $\eta_{observation}$ are applied here to decide whether these two kinds of stacks are large enough to update the TMC-HIST model. When one matrix in gait phase stacks has accumulated more than η_{gait} gaits, the joint probability of the relative activity will be calculated, and the corresponding joint probability in TMC-HIST model will be updated. After that, all the entries in this matrix are reset to zero. Likewise, the histograms in TMC-HIST model are replaced if the related stack's volume exceeds $\eta_{observation}$; reset is also conducted after the update. It should be noticed is that all the matrices and histogram stacks are accumulated and updated independently, because the duration of each gait phase is different from each other. Thanks to the posterior update, the on-line algorithm can adjust the parameters in TMC-HIST model according to the users' activity patterns and the road conditions, such as the speed of foot strike, ascent or descent slope...

4.4 Experiment and results

4.4.1 Experiment setup

Ten healthy subjects were invited for the experiment: three females and seven males, age from 25 to 47 years old, weight from 47 to 83 kg, height from 160 to 184 cm. The sensor sampling rate was set to $100Hz$, and the range of gyroscope to $\pm 1000deg/s$. The window size W for feature extraction was set to 15 samples based on our experience, which corresponds to $0.15s$. It was determined by the stance duration when running, which is the shortest gait phase duration of all the activities. This is reasonable since a window size bigger than the duration may reduce the detection accuracy of the shortest gait phase, whereas a too small W may not be sufficient for calculating the mean value and standard deviation. The ranges of the histograms in TMC-HIST were set from $-15rad/s$ to $15rad/s$ for mean value, and from $0rad/s$ to $15rad/s$ for standard deviation, and the bins number of

the histograms was all set to 300.

The proposed algorithm is evaluated on a designed experiment, which is conducted by the subjects. We utilize a 2-fold cross-validation method, by equally separating the subjects into two groups. This way guarantees that training data and testing data come from different sources. The experiment was conducted around a building with four floors on the campus of École Centrale de Lyon (France). The selected path consists of walking and running around the building (with ramp road conditions), as well as climbing and descending stairs. The exact sequence of activities is

(1) 600 meters of walking,

(2) 600 meters of running,

(3) four round trips of climbing stairs from ground floor to the 4th floor and back to the ground floor,

(4) repeat Steps (1) to (3) a second time.

Hence, one experiment consists in repeating two times the same sequence of activities, called here "a section". The ground truth of the experiment can be seen in Figure 4.7. It consists of $1200m$ of walking and running, 32 floors of stair ascent and descent. So, in the experiment there are four lower limb locomotion activities considered, which are most frequently performed in our daily life. The average time to complete one experiment is about 30 minutes. All the subjects perform the activities at their preferred speed, but are asked to keep the same speed within one experiment as much as possible. This ensures that the activity patterns and speeds vary among the subjects, but almost keep constant within one experiment. The speed ranges of the four activities among the subjects are $1.09 - 1.68m/s$, $2.14 - 3.82m/s$, $89.79 - 123.68$ stairs/min, $96.07 - 140.97$ stairs/min. The details of the activities of each subject is displayed in Table 4.1.

The implementation of the algorithms is done in Matlab, the code is running on a $3.2GHz$ CPU computer with 64-bit Win7 operating system. The average experiment time of all the subjects is 29.80 minutes, while the average calculation

Table 4.1: Details of each subject and activity.

	Gender	Age	Height (cm)	Weight (kg)	Data duration (min)	Walking spped (m/s)	Running speed (m/s)	Stair ascent speed (stair/min)	Stair descent speed (stair/min)
Subject 0	Male	28	184	72	28.77	1.3364	3.8193	118.3929	122.4231
Subject 1	Female	27	160	53	34.39	1.2687	2.4013	89.7953	111.3364
Subject 2	Male	30	177	71	34.82	1.1674	2.6869	94.0617	106.4287
Subject 3	Male	47	178	83	28.77	1.6781	3.0394	94.489	105.5059
Subject 4	Female	28	162	53	33.11	1.2671	2.1373	123.6822	133.6888
Subject 5	Male	25	170	52	33.60	1.1423	3.1555	103.928	106.0547
Subject 6	Male	31	176	62	33.60	1.4444	2.8957	113.062	140.9741
Subject 7	Male	27	177	73	33.52	1.2738	2.8623	93.0552	96.0725
Subject 8	Male	29	170	70	32.29	1.1539	3.2994	123.4343	107.6871
Subject 9	Female	28	160	47	35.20	1.0919	2.6799	117.0202	101.4551

time by code is 2.60 minutes. Since calculation time are more than 11 times faster than that experimental one, thus the processing can be theoretically made on-line on a processor of frequency higher than $3.2GHz/11 = 291MHz$.

4.4.2 Experimental results

As shown in Figure 3.3, the sensor is fixed on the shoe of each of subject. In Chapter 3, both acceleration and angular rate are involved in the feature extraction. However, because the subjects wear different types of shoes, especially for the front slope where the sensor is placed, thus the acceleration readings vary differently in each sensor axis. Therefore, in order to verify whether the variation of acceleration has an impact on the recognition performance, we utilize two kinds of features to test our TMC-HIST model. The first one only takes the angular rate and extract the sliding mean and standard deviation in the three sensor axes, whereas the second one take both acceleration and angular rate to extract the sliding mean and standard deviation. Thus, the observation dimension for the two kinds of features are 6 and 12 respectively.

Hereinafter, the experimental results will be displayed separately according to the two observations. In each of them, a global analysis over the ten subjects is given at first, to show a general performance of the batch model recognition and of the adaptive on-line recognition algorithm. Then, one subject's data will be used for precisely analysing the parameters updating.

4.4.2.1 Observation of angular rate

The confusion matrix of batch mode recognition using TMC-HIST based on 6-dimensional (6D) observation is shown in Table 4.2. The overall accuracy is 83.17% and the Matthews correlation coefficient (MCC)[1] [121]) is 0.7823. Table 4.3 shows the sensitivity, specificity, F1 score and MCC of each individual activity by the batch mode recognition.

For investigating the influences of η_{gait} and $\eta_{observation}$ –the size of the stacks

[1]MCC is a measure for multi-category classification, it can balance the influence that produced by the different proportion of each category, a value close to 1 means a perfect classification.

Table 4.2: Confusion matrix of batch mode recognition, using 6D observation.

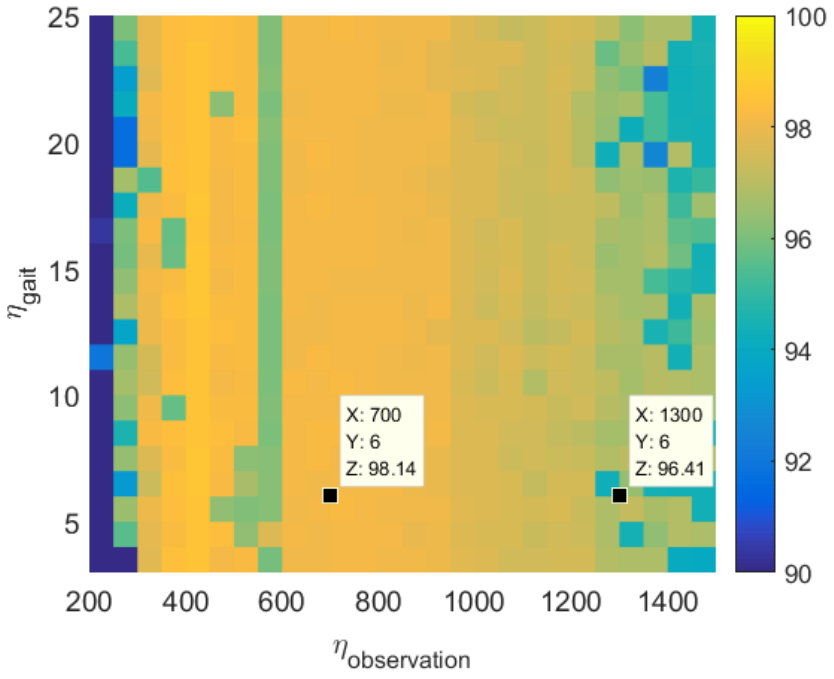
		Predicted activity			
		Walk	**Run**	**Stair ascent**	**Stair descent**
True activity	**Walk**	**74.34**%	16.71%	5.45%	3.50%
	Run	3.79%	**95.38**%	0.54%	0.29%
	Stair ascent	2.45%	4.00%	**85.70**%	7.85%
	Stair descent	1.82%	2.32%	3.70%	**92.15**%

Table 4.3: The sensitivity, specificity, F1 score, MCC value of the batch mode recognition by TMC-HIST, using 6D observation.

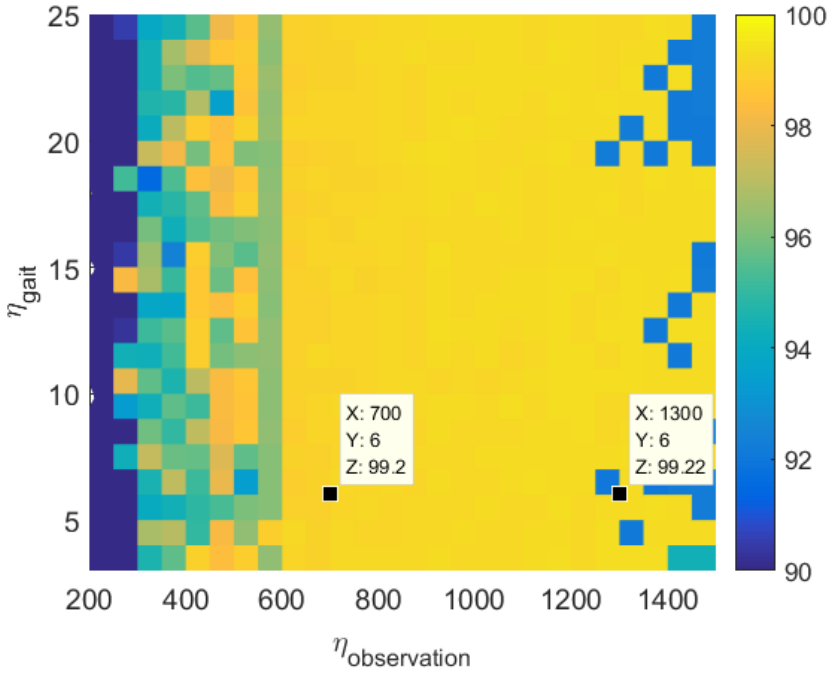
	Activity				
	Walking	**Running**	**Stair ascent**	**Stair descent**	**Total**
Sensitivity	0.7434	0.9538	0.8570	0.9215	0.8689
Specificity	0.9715	0.8834	0.9610	0.9656	0.9454
F1 Score	0.8385	0.8048	0.8250	0.8654	0.8334
MCC	0.7368	0.7554	0.7935	0.8437	0.7823

before updating TMC-HIST– have on the activity recognition performance. Figure 4.5 shows the influence on average accuracy over the ten subjects *w.r.t.* different values of the thresholds η_{gait} and $\eta_{observation}$. From the results, $\eta_{observation}$ affects the accuracy more than η_{gait} does, which means that the update of histograms is more critical than the update of joint probability matrix. The accuracy obtained in the second section is higher than in the first section, which indicates our proposed on-line algorithm can adjust the parameters in TMC-HIST model properly and improve the accuracy gradually. We can see that, as the value of $\eta_{observation}$ increases, the average accuracy reaches above 99% when $\eta_{observation} \in [600, 750]$. According to this experiment, we select $\eta_{observation} = 700$ and $\eta_{gait} = 6$ to analyse the performance of the proposed on-line algorithm.

Confusion matrices of the first and second sections of the experiment are shown in Table 4.4. We can see that the recognition performance of each activity is improved from the first to the second section. The overall accuracy increases from



(a)



(b)

Figure 4.5: Averaged activity recognition accuracy (in %) according to $\eta_{observation}$ and η_{gait} (size of stacks), for the first section of experiment (a) and for the second section (b), using 6D observation.

Table 4.4: Confusion matrix of the first section (up) and second section (down) of experiments, using 6D observation.

		Predicted activity			
		Walk	**Run**	**Stair ascent**	**Stair descent**
True activity	**Walk**	**98.69**%	0.58%	0.37%	0.36%
	Run	0.48%	**99.44**%	0.00%	0.08%
	Stair ascent	0.13%	1.18%	**97.65**%	1.05%
	Stair descent	0.29%	0.05%	4.96%	**94.70**%

		Predicted activity			
		Walk	**Run**	**Stair ascent**	**Stair descent**
True activity	**Walk**	**99.39**%	0.21%	0.23%	0.16%
	Run	0.16%	**99.83**%	0.01%	0.00%
	Stair ascent	0.86%	0.00%	**98.62**%	0.52%
	Stair descent	0.09%	0.00%	1.71%	**98.20**%

Table 4.5: The sensitivity, specificity, F1 score, MCC value of the on-line mode recognition by TMC-HIST, using 6D observation. Up: first section, down: second section.

	Activity				
	Walking	**Running**	**Stair ascent**	**Stair descent**	**Total**
Sensitivity	0.9869	0.9944	0.9765	0.9470	0.9762
Specificity	0.9968	0.9940	0.9897	0.9959	0.9941
F1 Score	0.9918	0.9864	0.9590	0.9606	0.9744
MCC	0.9838	0.9827	0.9520	0.9544	0.9682

	Activity				
	Walking	**Running**	**Stair ascent**	**Stair descent**	**Total**
Sensitivity	0.9939	0.9983	0.9862	0.9820	0.9901
Specificity	0.9965	0.9987	0.9958	0.9982	0.9973
F1 Score	0.9951	0.9968	0.9815	0.9855	0.9897
MCC	0.9905	0.9959	0.9782	0.9831	0.9869

98.14% in the first section to 99.20% in the second section, while MCC increases from 0.9682 to 0.9869. Table 4.5 shows the sensitivity, specificity, F1 score and MCC of each individual activity by the on-line recognition. In a general view, the coefficients are improved in the second section. The better performances in the second section show that the parameters adjustment in the proposed on-line algorithm is beneficial, which gradually leads to an improvement of recognition performances.

In order to intuitively understand the adaptation of the proposed on-line algorithm, Figure 4.6 shows the accuracy of each activity in the most recent 1000 samples, *i.e.*, the accuracy that calculated from the latest 10 seconds with respect to each activity. The most recent accuracy can be used for investigating the converging rate. As shown in the figure, running is the fastest to converge at an accuracy over 99%, walking converges at about $40s$, stair ascent and descent reach a relatively high accuracy after $100s$. The drops in accuracy after convergence is due to the activity switch, which may affect the recognition behavior during a short time, but will perform well again after the activity switch. It is not difficult to find that walking and running are easier to be classified than climbing stairs, showing that the model approximation with respect to climbing stairs is slower than that of walking and running. Nevertheless, all the activities reach a high accuracy level in the experiments finally.

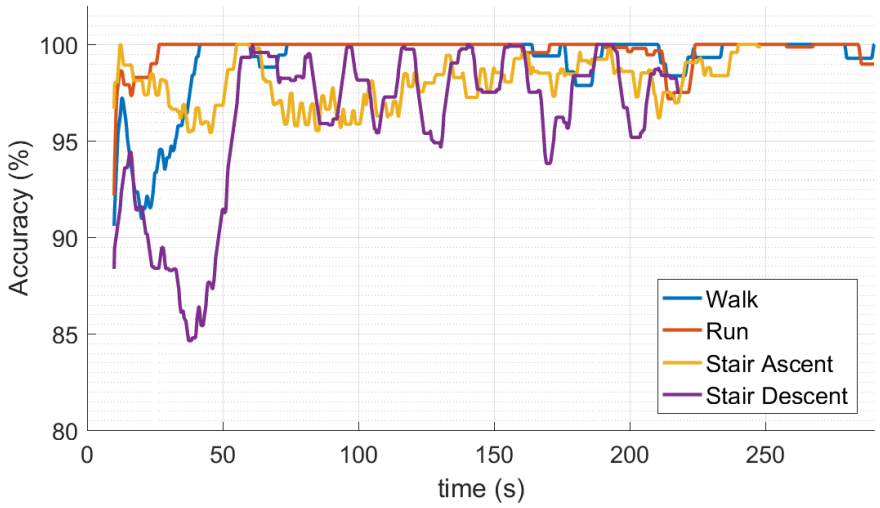


Figure 4.6: Accuracy in the most recent 1000 samples *w.r.t.* each activities.

For better understanding parameters updating, a typical subject's data is selected here (subject 5, male, 25 years old, 52kg weight and 170cm tall). One reason for selecting this experimental data is because the accuracy and MCC in the second section, 98.99% and 0.9800 respectively, are lower than the averaged values of all the subjects. So, Figure 4.7 shows the ground truth and classified activities of the subject 5. It can be seen that, in the second section, some samples of walking are wrongly classified as stair descent. And stair ascent and stair descent are misclassified at the beginning. But as expected, the performance in the second section is much better than in the previous one. Regarding the gait phases, there is no precise evaluation method because of the lacking of the proper device to collect the ground truth of gait cycles. However, as shown in Figure 4.8, it is obvious that the gait cycles detection becomes more regular when the model's parameters have converged to fit the subject gait rhythm. Indeed, gait cycles are introduced in the proposed algorithm to make the on-line algorithm possible and help to obtain a better activity recognising result, therefore, a well-detected gait cycle assists to obtain a good recognition of activity.

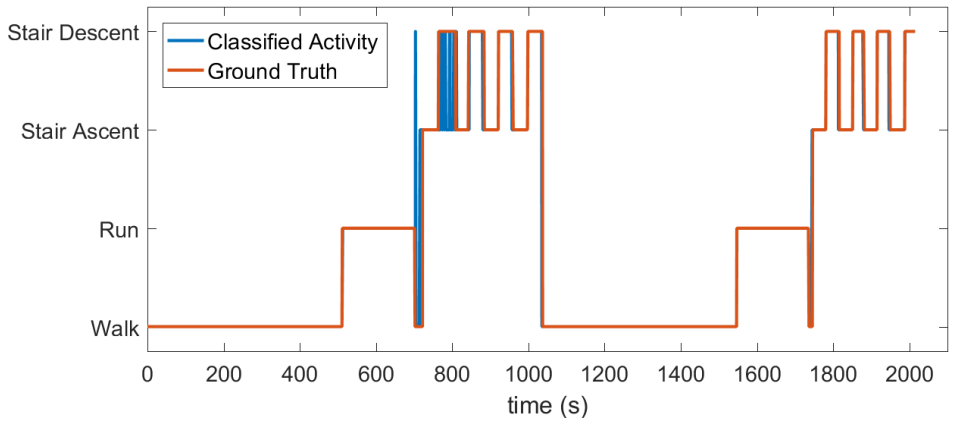


Figure 4.7: Recognized activities from TMC-HIST for subject 5, using 6D observation. For the first section, accuracy and MCC are 94.49% and 0.9055 respectively, while, for the second section, the values are 98.99% and 0.9800 respectively.

(a) Walking at beginning.

(b) Walking when converged.

(c) Running at beginning.

(d) Running when converged.

(e) Stair ascent at beginning.

(f) Stair ascent when converged.

(g) Stair descent at beginning.

(h) Stair descent when converged.

Figure 4.8: The detected gait cycles at the beginning of each activity (left column) compared to the ones when the estimation of the model's parameter has converged (right column).

4.4.2.2 Observation of acceleration and angular rate

The confusion matrix of batch mode recognition using TMC-HIST based on 12-dimensional (12D) observation is shown in Table 4.6, the accordingly coefficients are displayed in Table 4.7. The overall accuracy is 78.66% and the Matthews correlation coefficient is 0.7319.

Figure 4.9 shows the influence on average accuracy over the ten subjects *w.r.t.* different values of the thresholds η_{gait} and $\eta_{observation}$. We can see that the accuracy at $\eta_{observation} = 700$, $\eta_{gait} = 6$ is the same as the one in Figure 4.5b. But the accuracy at $\eta_{observation} = 1300$, $\eta_{gait} = 6$ is much lower than the one in Figure 4.5a.

Table 4.6: Confusion matrix of batch mode recognition, using 12D observation.

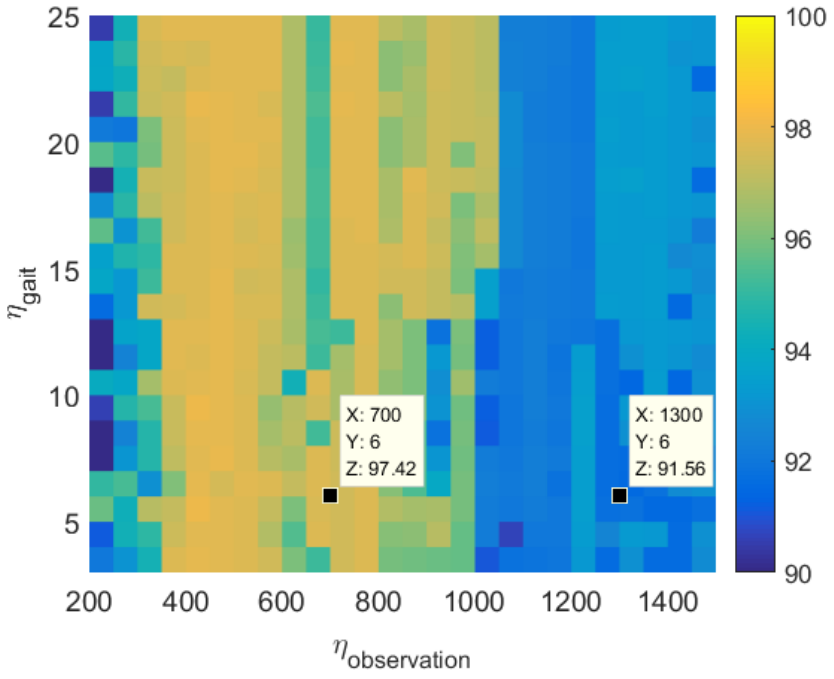
		Predicted activity			
		Walk	**Run**	**Stair ascent**	**Stair descent**
True activity	**Walk**	**64.30**%	15.51%	12.44%	7.75%
	Run	1.24%	**95.83**%	0.61%	2.33%
	Stair ascent	1.54%	2.70%	**87.39**%	8.37%
	Stair descent	0.66%	0.72%	6.11%	**92.51**%

Table 4.7: The sensitivity, specificity, F1 score, MCC value of the batch mode recognition by TMC-HIST, using 12D observation.

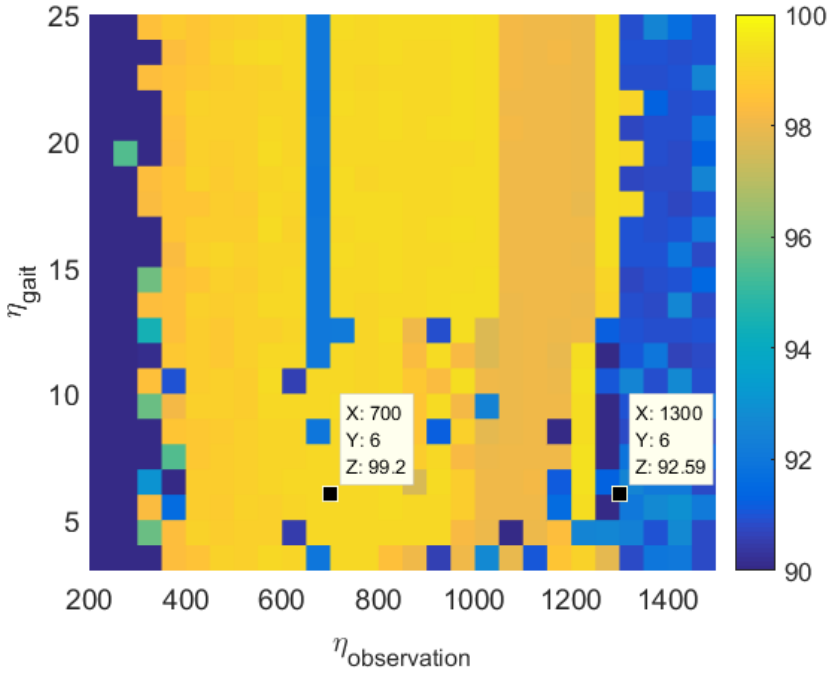
	Activity				
	Walking	**Running**	**Stair ascent**	**Stair descent**	**Total**
Sensitivity	0.6430	0.9583	0.8739	0.9251	0.8501
Specificity	0.9884	0.8964	0.9166	0.9352	0.9341
F1 Score	0.7770	0.8231	0.7449	0.7986	0.7859
MCC	0.6762	0.7781	0.7031	0.7701	0.7319

Particularly when $\eta_{observation} = 1300$, $\eta_{gait} = 6$, the confusion matrices of the two sections are displayed in Table 4.8. The sensitivity, specificity, F1 score, MCC of the two sections are shown in Table 4.9.

As the same as the Section 4.4.2.1, we display the results from subject 5 to show the performance of on-line updating, where the thresholds $\eta_{observation} = 1300$,



(a)



(b)

Figure 4.9: Averaged activity recognition accuracy (in %) according to $\eta_{observation}$ and η_{gait} (size of stacks), for the first section of experiment (a) and for the second section (b), using 12D observation.

Table 4.8: Confusion matrix of the first section (up) and second section (down) of experiments, using 12D observation.

		Predicted activity			
		Walk	**Run**	**Stair ascent**	**Stair descent**
True activity	**Walk**	**98.48**%	0.39%	0.78%	0.35%
	Run	0.27%	**99.56**%	0.08%	0.09%
	Stair ascent	2.47%	0.00%	**95.42**%	2.11%
	Stair descent	0.98%	0.00%	6.52%	**92.50**%

		Predicted activity			
		Walk	**Run**	**Stair ascent**	**Stair descent**
True activity	**Walk**	**99.56**%	0.06%	0.29%	0.09%
	Run	0.13%	**99.86**%	0.01%	0.00%
	Stair ascent	0.79%	0.00%	**98.5**%	0.66%
	Stair descent	0.28%	0.00%	2.07%	**97.65**%

Table 4.9: The sensitivity, specificity, F1 score, MCC value of the on-line mode recognition by TMC-HIST, using 12D observation. Up: first section, down: second section.

	Activity				
	Walking	**Running**	**Stair ascent**	**Stair descent**	**Total**
Sensitivity	0.9848	0.9956	0.9542	0.9250	0.9649
Specificity	0.9889	0.9975	0.9845	0.9942	0.9913
F1 Score	0.9867	0.9933	0.9337	0.9436	0.9643
MCC	0.9737	0.9915	0.9223	0.9348	0.9556

	Activity				
	Walking	**Running**	**Stair ascent**	**Stair descent**	**Total**
Sensitivity	0.9956	0.9986	0.9855	0.9765	0.9891
Specificity	0.9963	0.9996	0.9949	0.9983	0.9973
F1 Score	0.9958	0.9987	0.9787	0.9830	0.9890
MCC	0.9919	0.9983	0.9748	0.9802	0.9863

$\eta_{gait} = 6$. Figure 4.10 is the accuracy in recent 10 seconds of each activity. Figure 4.11 is the recognized activities of subject 5. From the Figures, we can see that the performance of using 12D feature is similar with the one of 6D feature (Figure 4.7 and 4.6). All the activities reach a high accuracy level before 100 seconds. While the recognition accuracy is improved from 93.25% in the first section to 98.81%, MCC is improved from 0.8869 to 0.9766. The detected gait cycles are shown in Figure 4.12.

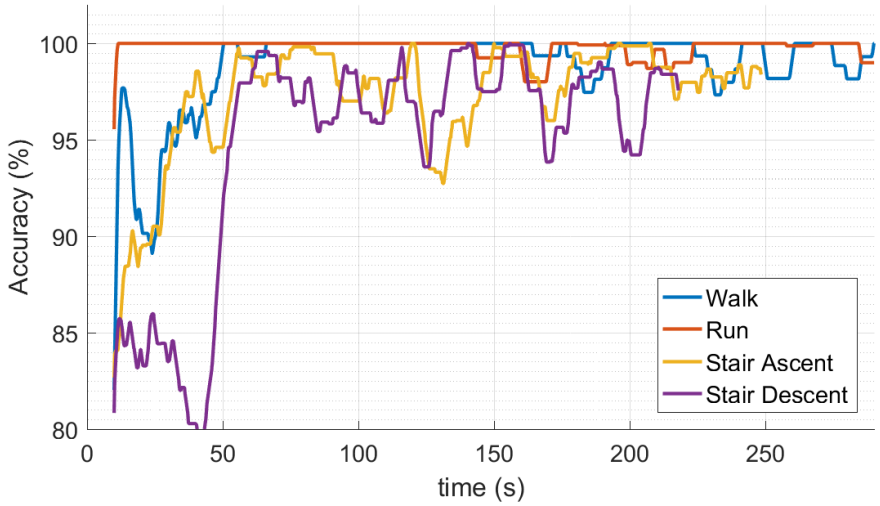


Figure 4.10: Accuracy in the most recent 1000 samples *w.r.t.* each activities.

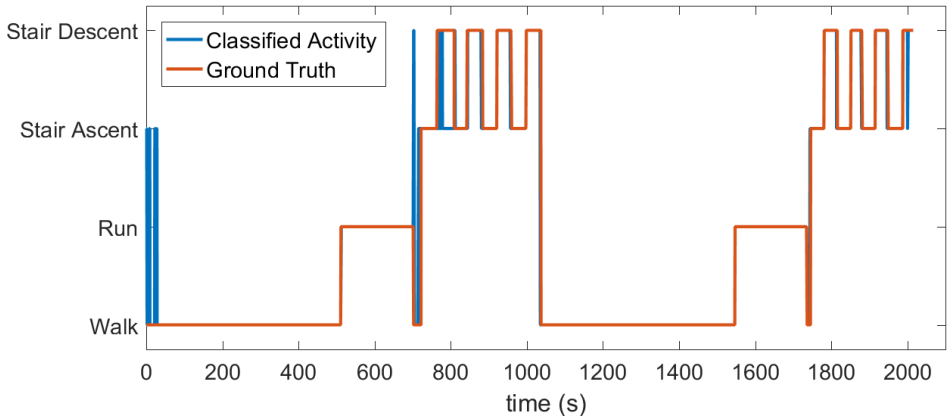
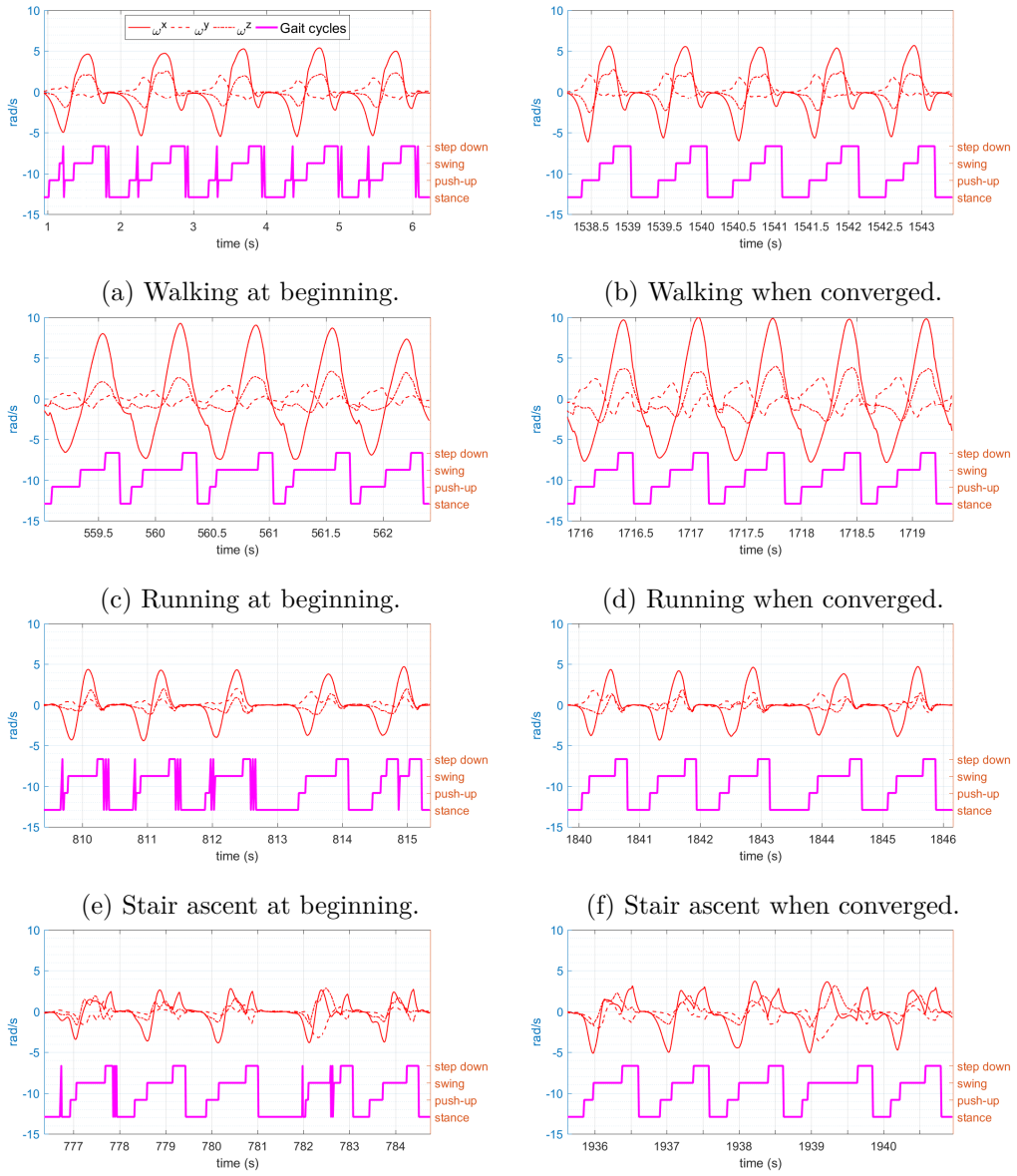


Figure 4.11: Recognized activities from TMC-HIST for subject 5, using 12D observation. For the first section, accuracy and MCC are 93.25% and 0.8869 respectively, while, for the second section, the values are 98.81% and 0.9766 respectively.



(a) Walking at beginning.

(b) Walking when converged.

(c) Running at beginning.

(d) Running when converged.

(e) Stair ascent at beginning.

(f) Stair ascent when converged.

(g) Stair descent at beginning.

(h) Stair descent when converged.

Figure 4.12: The detected gait cycles at the beginning of each activity (left column) compared to the ones when the estimation of the model's parameter has converged (right column).

4.5 Discussion

This section proposes an in-depth discussion on the experimental results for key points of the proposed algorithm, such as the impact of observation, classification

accuracy, convergence rate, gait phase detection, parameter update...

4.5.1 Impact of acceleration

The two Sections 4.4.2.1 and 4.4.2.2 show the performances obtained by two kinds of observations, the first section only uses angular rates as observation (6D), while the second section uses both angular rates and accelerations (12D). In a general view, only using angular rates seems to be more appropriate than using both angular rates and accelerations in our experiments. The former one obtains better performances in both batch mode and on-line mode recognition.

In batch mode recognition, 6D observation obtains an accuracy of 83.17%, while the one of 12D observation is 78.66%. The MCC values of the two are 0.7823 and 0.7319 respectively. In on-line mode recognition, both of the two kinds of observations can reach a high accuracy in the second experiment section. However, comparing the Figure 4.5b and 4.9b, we can see that 6D observation is more robust and suitable *w.r.t.* different η_{gait} and $\eta_{observation}$, especially when $\eta_{observation}$ is very large. The distinctions in batch mode recognition is because of the different shoes that used by different subjects, since the front slopes of the shoes may cause distinctions in $a^{\mathcal{X}}$ and $a^{\mathcal{Y}}$ (Figure 3.3). Consequently, the batch mode trained model acts as the initial model for the on-line recognition. It is sure that a better initial model can obtain a more robust on-line recognition result.

Specifically, at the point $\eta_{observation} = 700$ and $\eta_{gait} = 6$ where both the two kinds of observations reach an accuracy of 99.20%, Figure 4.6 and 4.10 show that the parameter update behaves similarly between the two. Likewise, the detected gait cycles, Figure 4.8 and 4.12, also behave similarly. This is because that if the initial model can properly provide estimated state at the beginning, then the model will be updated to fit the pattern of specific subject. It means that all the histograms related to angular rates and accelerations reflect the specific subject after correct parameter update. Therefore, the impact of acceleration is reduced.

Because using 6D observation is more robust than the other one, the discussions in following will be based on the results of 6D observation.

4.5.2 Recognition Performance

A comparison of the performances of the proposed adaptive on-line algorithm compared to some state-of-the-art algorithms is displayed in Table 4.10. These algorithms are evaluated according to several aspects including the number of sensors, the average accuracy, the MCC...

Among all the works that use one single sensor reported in Table 4.10, the proposed algorithm obtains the highest average accuracy with 99.20%. While the Phase Variable obtained an accuracy of 98.30% for only 3 activities: walking, stair ascent and descent. Descriptor-based methods obtained 97.12% on 6 lower limb activities, whereas energy expenditure prediction obtained 95.05% on 5 lower limb activities. Adaboost Stump got an accuracy range from 95%-98% on 5 lower limb activities and PLP+HMM obtained $97.5 \pm 1.6\%$ on 6 activities, with 3 lower limb activities identical to the work of Phase Variable. In the group of multiple sensors, Log-Sum distance used 5 sensors to classify 6 activities (standing still, sitting still, laying down, walking forward, stair ascent and descent). The accuracy of Log-sum distance can reach 99%. Empirical mode decomposition used 17 sensors to classify 9 activities and obtained an overall accuracy at 97.78%. Among the 9 activities, the 7 lower limb activities had an accuracy at about 96%. Adaptive BasIS used 3 sensors and obtained 99.87% on three activities: walking on level ground, ramp ascent and descent. NWFE+PCA+LS-SVM used 2 sensors and obtained an overall accuracy at 99.65% for 10 daily activities. Among the 10 activities, the 5 lower limb activities had an accuracy at about 96.46%. If looking at all the algorithms that give an accuracy higher than 99%, the proposed algorithm is the only one that uses one single sensor, besides, the proposed algorithm obtained the highest MCC value. The use of multiple sensors is very interesting and can give very high accuracy. Nevertheless, recognizing activities with only one sensor is still relevant because it is more realistic than multiple sensors for quantified-self applications. It should be noticed that the state-of-the-art works and our algorithm are tested on different datasets, consisting of different activities and different amounts of samples. But from a general view of the results, we can still state that our proposed algorithm

Table 4.10: Comparison of the performance *w.r.t.* state-of-the-art algorithms for recognising lower limb locomotion activities.

Method	**Sensors**	**Position**	**No. of activity**	**No. of subject**	**Accuracy**	**MCC**	**On-line**
Adaptive BasIS (Martinez-Hernandez, 2018 [6])	3	foot, shank thigh	3	8	99.87%	—	NO
Phase Variable (Harrison & Michael, 2018, [122])	1	thigh	3	7	98.30%	—	NO
NWFE+PCA+LS-SVM (Yu-Liang Hsu, 2018, [40])	2	ankle, wrist	10	23	99.65% (daily activity)	0.9804	NO
Descriptor-based (Ankita Jain, 2018, [123])	1	waist	6	30	97.12%	0.966	NO
Log-Sum Distance (Faisal Sikder, 2017, [124])	5	waist, wrists ankles	6	3	99.00%	0.9734	NO
Energy expenditure prediction (Lin, 2017, [125])	1	foot	5	10	95.05%	0.9336	NO
Adaboost Stump (Li, 2017, [126])	1	trouser pocket	5	12	$95\% - 98\%$	—	YES
Empirical mode decomposition (Ayachi, 2016, [41])	17	whole body	9	4	97.78%	0.823	NO
PLP+HMM (San-Segundo, 2016, [80])	1	waist	6	30	$97.5 \pm 1.6\%$	—	NO
The proposed algorithm (TMC-HIST)	1	foot	4	10	99.20%	0.9869	YES

obtain performances comparable to or even better than the best ones, whereas allowing on-line processing.

4.5.3 Convergence Rate

It can be deduced from Figure 4.5 and 4.6 that the convergence rate is affected by two aspects: the thresholds ($\eta_{observation}$, η_{gait}) and the activity category. These two aspects will be discussed in the following.

The first aspect is the thresholds that control the parameters updating, *cf.* Figure 4.5. In the first experimental section, the accuracy at $\eta_{observation} = 1300$ is relatively lower than that of $\eta_{observation} = 700$ (96.41% compared to 98.14%), which means that using a small value of $\eta_{observation}$ gives a faster convergence rate than using a large value. While in the second section, the accuracies where $\eta_{observation} = 700$ and $\eta_{observation} = 1300$ are improved to 99.2% and 99.22% respectively. The close performance in the second section indicates that high accuracy can be reached through parameter updating with sufficient data. It should be noticed that in Figure 4.5b, there are some isolated values when $\eta_{observation}$ is larger than 1200. This is because that in the experiment of one special subject, a large quantity of walking data are classified as running at the beginning, and the too late updating makes all the consequent walking data be classified as running. On the contrary, we can see that a too small $\eta_{observation}$ leads to a worse result in the second section compared to the first section. It is due to the fact that the accumulated data number in each stack is not enough for representing the distribution of $p(\boldsymbol{y}_n|\boldsymbol{v}_n)$. A large number of bins in histogram requires more data to form the proper density, if $\eta_{observation}$ is too small, the updated histograms can not properly represent $p(\boldsymbol{y}_n|\boldsymbol{v}_n)$. In this case, the classification performance will be not too bad since the initial model is obtained from training data. But, after several updates by the inadequate observation stacks, TMC-HIST model will be far from the actual one, which results in a reduction of accuracy. Therefore, based on the accuracy shown in Figure 4.5b, we recommend $\eta_{observation}$ to be equal to $2 - 2.5$ times the bins number of histograms.

It seems like the variation in h_{gait} has little impact on the accuracy. The main reason is that the transition probability $p(\boldsymbol{t}_{n+1}|\boldsymbol{t}_n)$ where $\boldsymbol{v}_{n+1} \neq \boldsymbol{v}_n$ is very small

for all kinds of $\boldsymbol{v}_n$ and all subjects. Indeed, each state will keep the same for a period of time and then transfer to the next state. Thus, a slight change in $p(\boldsymbol{v}_n, \boldsymbol{v}_{n+1})$ will not lead to a significant change in the classification performance.

Another aspect is the activity category. Figure 4.6 shows that walking and running are faster than the other two to reach an accuracy higher than 99%. This means that the parameters in TMC-HIST relative to walking and running converge faster than those of stair ascent and stair descent. This phenomenon is certainly to the amount of effort required to climb 32 floors. It is then more difficult to keep the same pace within one experiment. As shown in Figure 4.8, we can find that the feature patterns of each gait cycle corresponding to walking and running are similar in both two experimental sections, while the distinctions of stairs ascent and descent are larger than the former two activities. As a result, convergence rates corresponding to stair ascent and descent are slowed down by the distinction, and that is also why the accuracy fluctuates extensively after reaching a high value. Regarding the fastest convergence rate (running), the signal values are much higher than the others, which indicates that the running signal is unique from the others and easy to distinguish.

4.5.4 Adaptation and On-line

As a matter of fact, the gait complete detection stage of the algorithm makes the on-line possible, while the posterior update stage enables the TMC-HIST model to be adaptive. Introducing the gait cycle to assist locomotion is practical, because each gait phase gives a prior condition to confine the number of possible states to be estimated in TMC-HIST. Indeed, the Adaptive BasIS algorithm [6] also introduced gait cycles to help classifying three activities of level ground walking, ramp ascent and descent walking. The Phase Variable algorithm tried to segment each gait cycle to classify the activities of level ground walking, stair ascent and descent activities. Both of these two algorithms obtained good results, as illustrated in section 4.5.2, but they did not utilize the actual structure of the gait phases in one gait cycle, *i.e.* the transition sequence of gait phases. While in our proposed algorithm, a Markov chain is established to mimic this structure. Generally, segmentation of gait cycles

can only be accomplished in an off-line scenario, it is difficult to make a decision if only one sample is known. The authors of Phase Variable algorithm [122] stated that their method can be applied to the on-line scenario if using a small time window. However, thanks to the state transitions among the activities and gait phases, gait phase can be estimated at each sampling time with the help of previous estimation. Activity is then detected after the current gait cycle is completed, and does not need to wait for the entire data or to set a time window for segmentation. It should be noticed that the gait detection is based on the forward procedure of TMC-HIST model, which is not as reliable as the backward procedure. A precise evaluation of the algorithm behavior with respect to gait phases was not conducted because of the lacking of proper device to collect ground truth.

Many adaptive methods were proposed for motion analysis. A machine learning method using adaptive local motion descriptor in [127] was proposed for recognizing human motion in videos. A fast and adaptive sparse representation method in [128] reached an accuracy up to 94% for the recognition of human activities using wearable sensors. Zhang [129] proposed an adaptive time window method for human activity classification and reached an accuracy up to 99.2%. Hameed [130] proposed adaptive zero crossing technique to detect muscle activity based on electrocardiography signals. Li [126] and Wen [79] used Adaboost for human activity classification with inertial sensors, and obtained accuracy up to 98%, particularly, Li's method was applied in on-line scenarios. In our proposed algorithm, the adaptive functionality is conducted by the posterior update stage, by updating the parameters in TMC-HIST to approximate the user's activity patterns. Combining with the gait cycle detection, our results show that a roughly estimated gait cycle makes posterior update works appropriately, then activity classification is improved because of parameters adaptation. See Figure 4.8, for each activity, the detected gait cycles are more regular than the ones detected at the beginning. In return, the improved activity classification results can refine the gait cycle detection till to the gait cycle detection behaves well.

4.6 Conclusion

In this Chapter, we were looking for an algorithm which can recognize lower limb locomotion activities and gait phases simultaneously. The LR-HMC model presented in Chpater 3 is then extended into a more sophisticated structure, *i.e.* TMC-HIST that recognizes lower limb locomotion activities and gait phases simultaneously. In TMC-HIST, an auxiliary hidden process $\boldsymbol{X}$ is introduced into the model, then a hidden state transition graph (Figure 4.3) is proposed to mimic the real transition order of the activities and gait phases. Histograms are utilized in TMC-HIST to represent the class conditioned observation density, for better representing the density than the Gaussian. Because of the non-parametric densities, ICE principle is applied to learn the parameters of the model. Then, an adaptive on-line recognition activity algorithm based on TMC-HIST is developed. This algorithm is designed for recognising activities at run-time and adaptively updating the model to fit the motion pattern of subject. Finally, the algorithm is performed over the experiment data. Experimental results show that TMC-HIST is capable of estimating gait phases and activities accurately, and the adaptive on-line algorithm can adjust the model parameter correctly to reach a high recognition accuracy.

However, there are some weaknesses of TMC-HIST. The activity estimation is only available when an entire gait cycle is completed, and also the estimated activity is obtained by taking the dominated one within one gait cycle. So, on the next Chapter, a parametric TMC model is proposed to tackle these issues.

Chapter 5

Adaptive on-line recognition using Parametric Semi TMC

We described in the previous Chapter of using TMC-HIST to recognize lower limb locomotion activities. However, the TMC-HIST algorithm presents some weaknesses.

From the Figures 4.7 and 4.8, we can see that the activity (hidden state $\boldsymbol{X}$) and gait phase (hidden state $\boldsymbol{U}$) share transition similarities: they both keep stable most of the time, the transitions from one state to another one are quite rare. In fact, this is the reality when people perform displacement: they do not switch activity every second, while the gait phase also keeps the same over many samplings, if the sensor sampling rate is not set very low. In TMC-HIST, we defined the hidden state transition graph (see Figure 4.3) to restrict the possible directions among the hidden states, *i.e.* the hidden state transition matrix $\boldsymbol{A}$ is a specific sparse matrix with many zero probabilities. For the non-zero entries in $\boldsymbol{A}$, the diagonal elements (the states keep same) is nearly close to 1, and the others are close to 0, which means that the probability of transferring to a different state is very low. Please note that it is still possible for the state to transfer to another state at every sampling time, even though the probability is very low. As shown in Equation (4.3), the estimated state depends on the forward probability $\alpha_n(\boldsymbol{v}_n)$ and backward probability $\beta_n(\boldsymbol{v}_n)$, while $\alpha_n(\boldsymbol{v}_n)$ and $\beta_n(\boldsymbol{v}_n)$ depend on state transition probability and class-conditional observation density respectively, see Equations (3.5) and (3.6). Thus, the smoothed probability $p(\boldsymbol{v}_n|\boldsymbol{y}_1^N)$ is affected by both transition probability and class-conditional observation density. That is why the estimated gait phases in the left column of Figure 4.8 have unusual transition or-

der, since the non-converged class-conditional observation densities may give biased probability to calculate $p(\boldsymbol{v}_n|\boldsymbol{y}_1^N)$. To overcome the limit of the state transition in TMC-HIST, semi-Markov structure is introduced in this Chapter. Semi-Markov structure allows the hidden state to keep the same for a while by introducing a sojourn hidden state process into the model, which naturally coincides the activity and gait transition during the motion. Therefore, using semi-Markov may help to detect a more regular gait cycle than TMC-HIST, which in return may probably enhance the activity recognition accuracy.

Another problem is the non-parametric class-conditional observation density in TMC-HIST, in which each histogram represents the marginal density *w.r.t.* one sensor axis and does not consider the correlations among the three axes of the sensor. It is sure that the three axes of the sensor are correlated during the motion, and the extracted features are correlated with each other. So using a w-dimensional histogram should be more reasonable than the 1-dimensional histogram, *i.e.* the marginal density. However, too much data should be needed to build the w-dimensional histogram. As an example, Figure 4.5 shows that the 1-dimensional histogram needs at least 600 data to form the density. Because there are 16 possible hidden states, then the minimum needed data volume is about $600 \times 16 = 9600$. By contrast, the minimum data volume for w-dimensional histogram may probably be $600^w \times 16$, which is impractical in the application. Hence, Gaussian mixture model (GMM) density is introduced to approximate the non-linear class-conditional observation density in this Chapter, by adding another hidden state process into the model. GMM involves the correlation through the $w \times w$ covariance matrices, so it conveys more information than that of 1-dimensional histogram. Since the parametric density is introduced in the model, on-line EM algorithm can be adopted to make the model adaptive and work at run time.

Because of the weaknesses of TMC-HIST and their corresponding solutions, a new model is going to be developed in this Chapter. Based on the conventional TMC structure described in Section 4.1, semi-Markov structure and GMM density are added to enrich the model, this new model is referred as SemiTMC-GMM. Thus, apart from the three already existing processes ($\boldsymbol{X}$,$\boldsymbol{U}$ and $\boldsymbol{Y}$), other two hidden

state processes are added to support the semi-Markov structure and GMM density respectively. Regarding to the sophisticated infrastructure of SemiTMC-GMM, TMC is gradually equipped with the other two processes in following sections, *i.e.* applying GMM to TMC to obtain the TMC-GMM model and then applying semi-Markov to TMC-GMM to obtain SemiTMC-GMM. Then, how to learn the parameter of SemiTMC-GMM in batch mode is given, an on-line parameter learning algorithm based on EM is given as well.

5.1 TMC-GMM

Assume that the Markovian $\boldsymbol{T} = (\boldsymbol{X}, \boldsymbol{U}, \boldsymbol{Y})$ forms the conventional TMC model as described in Chapter 4, where $X_n \in \Lambda = \{1, \cdots, r\}$ and $U_n \in \Gamma = \{1, \cdots, \tau\}$ are the hidden state processes representing activity and gait phase respectively, $\boldsymbol{Y}_n \in \mathbb{R}^w$ is a real-valued process representing the observation. Recalling that $\boldsymbol{V} = (\boldsymbol{X}, \boldsymbol{U})$ and $\boldsymbol{T} = (\boldsymbol{V}, \boldsymbol{Y})$, then $\boldsymbol{V}_n \in \Lambda \times \Gamma$. The transition of $\boldsymbol{T}$ follows the Equation (4.1). Unlike the non-parametric class-conditional observation density in TMC-HIST, let $p(\boldsymbol{y}_n|\boldsymbol{v}_n)$ be the conventional Gaussian density:

$$p\left(\boldsymbol{y}_n|\boldsymbol{v}_n = i\right) \sim \mathcal{N}\left(\boldsymbol{\mu}_i, \Sigma_i\right), i \in \Lambda \times \Gamma. \tag{5.1}$$

For obtaining the probability of individual x_n conditioned on all observations, we only need to compute the marginal probability of $p(x_n, u_n|\boldsymbol{y}_1^n)$ and $p(x_n, u_n|\boldsymbol{y}_1^N)$; the $p(x_n|\boldsymbol{y}_1^n)$, $p(x_n|\boldsymbol{y}_1^N)$ are simply given by:

$$\begin{aligned} p(x_n|\boldsymbol{y}_1^n) &= \sum_{u_n} p(x_n, u_n|\boldsymbol{y}_1^n), \\ p(x_n|\boldsymbol{y}_1^N) &= \sum_{u_n} p(x_n, u_n|\boldsymbol{y}_1^N). \end{aligned} \tag{5.2}$$

Likewise, $p(u_n|\boldsymbol{y}_1^n)$ and $p(u_n|\boldsymbol{y}_1^N)$ can be obtained in a similar way. The estimated

hidden state $\hat{x}_n$ and $\hat{u}_n$ are obtained by the MPM criteria:

$$\begin{aligned} \hat{x}_n &= \arg \max_{x_n \in \Lambda} p\left(x_n | \boldsymbol{y}_1^N\right), \\ \hat{u}_n &= \arg \max_{u_n \in \Gamma} p\left(u_n | \boldsymbol{y}_1^N\right). \end{aligned} \tag{5.3}$$

When extending TMC to TMC-GMM, it needs to introduce Gaussian mixture density into the class-conditional observation probability. In fact, embedding GMM in TMC can be regarded as introducing a new statistic process $\boldsymbol{H} = (H_1, \cdots, H_N)$ into TMC, where H_n takes its value h_n in a finite set $K = \{1, \cdots, \kappa\}$ and κ is the number of Gaussian components in the mixture. Please remind that $\boldsymbol{H}$ has no realistic meaning, it is just a latent variable in the model to introduce the mixtures. Let c_{ij} be the weight of jth Gaussian mixture component when $\boldsymbol{v}_n = i$, with the constraint $\sum_{j=1}^{\kappa} c_{ij} = 1$. $\boldsymbol{\mu}_{ij}$ and Σ_{ij} are the mean value and covariance of the Gaussian mixture component. Let us set $\boldsymbol{Z} = (\boldsymbol{T}, \boldsymbol{H})$, and consider that $\boldsymbol{Z}$ is Markovian with the following transitions:

$$p(\boldsymbol{z}_{n+1}|\boldsymbol{z}_n) = p(\boldsymbol{v}_{n+1}|\boldsymbol{v}_n) p(h_{n+1}|\boldsymbol{v}_{n+1}) p(\boldsymbol{y}_{n+1}|\boldsymbol{v}_{n+1}, h_{n+1}), \tag{5.4}$$

where $p(\boldsymbol{y}_n|\boldsymbol{v}_n)$ is

$$\begin{aligned} p(\boldsymbol{y}_n|\boldsymbol{v}_n) &= \sum_{j=1}^{\kappa} c_{ij} \cdot p(\boldsymbol{y}_n|\boldsymbol{v}_n = i, h_n = j), \\ p(\boldsymbol{y}_n|\boldsymbol{v}_n = i, h_n = j) &\sim \mathcal{N}\left(\boldsymbol{\mu}_{ij}, \Sigma_{ij}\right), \quad i \in \Lambda \times \Gamma, j \in K, \end{aligned} \tag{5.5}$$

with $p(h_n = j|\boldsymbol{v}_n = i) = c_{ij}$. We can see that Equations (5.4) and (5.5) are the extensions from Equations (4.1) and (5.1), by introducing a new process H. The dependency graph of TMC-GMM is shown in Figure 5.1a.

Please notice that the only difference between TMC and TMC-GMM is the Gaussian densities in TMC are replaced with Gaussian mixtures, all the other calculations remain the same. Then estimating the individual x_n and u_n in TMC-GMM follows the same as in TMC, by using Equation (5.2) and (5.3).

5.2 Semi TMC-GMM

In the Markov model considered in our previous work [7], the remaining time of the sojourn of the hidden state $\boldsymbol{v}_n$ is of geometric distribution. While considering $\boldsymbol{V}$ as semi-Markovian seems to better suited to our problematic, as in general $\boldsymbol{v}_n$ has no geometric remaining sojourn time. For example, the gait phase has a minimum duration, while in geometric distribution the maximal probability is for null duration. In real applications of classic hidden semi-Markov model (HSMM) [131, 132], there is a fixed maximum sojourn time for each possible value of $\boldsymbol{v}_n$. When $\boldsymbol{v}_n$ switches to a new value, the maximal possible random sojourn time is shorter than a fixed value M. Once the sojourn time has elapsed at time n, the hidden state must change to a different value, *i.e.* $p(\boldsymbol{v}_{n+1} = \boldsymbol{v}_n) = 0$. This implies that the maximum sojourn time should be large enough to cover the largest possible sojourn time, which appears as a drawback in our application. In another semi-Markov approach described in [133] that we adopt here, the random sojourn time is not the exact remaining duration of the state, but the minimum remaining sojourn time. This means that once the sojourn time elapsed, the next hidden state is possible to stay the same. This character allows make the maximum value M, which is the maximum of the minimum sojourn time, significantly smaller than the one in classic HSMM. This accelerates the entire method since the dimension of transition matrix is reduced.

To be more precise, consider a new stochastic process $\boldsymbol{D} = (D_1, \cdots, D_N)$ that represents the minimum remaining sojourn time in current hidden state $\boldsymbol{v}_n$, and the possible realization of each D_n (denoted by d_n) takes its value in $L = \{0, 1, \cdots, \ell\}$. Thus for $\boldsymbol{V}_n = \boldsymbol{v}_n$ and $D_n = d_n$, we have $\boldsymbol{v}_n = \boldsymbol{v}_{n+1} = \cdots = \boldsymbol{v}_{n+d_n}$. And $\boldsymbol{v}_{n+d_n}$ is obtained *w.r.t.* $p(\boldsymbol{v}_{n+d_n+1}|\boldsymbol{v}_{n+d_n})$, which is a transition similar to the ones in the TMC and TMC-GMM. Thus, $\boldsymbol{v}_{n+d_n+1}$ is possible to be the same as $\boldsymbol{v}_{n+d_n}$. Once $\boldsymbol{v}_{n+d_n+1}$ is set, a new minimum sojourn time d_{n+1} is obtained in $L = \{0, 1, \cdots, \ell\}$. Please notice that for $D_n = d_n \neq 0$, there is $D_{n+1} = d_{n+1} = d_n - 1$, which is specified in Equation 5.8.

Finally, SemiTMC-GMM is extended from TMC-GMM $\boldsymbol{Z}$ via the couple $(\boldsymbol{Z}, \boldsymbol{D})$,

and follows the transition probabilities:

$$p(\boldsymbol{z}_{n+1}, d_{n+1}|\boldsymbol{z}_n, d_n) = p(\boldsymbol{v}_{n+1}|\boldsymbol{z}_n, d_n)p(h_{n+1}|\boldsymbol{v}_{n+1})p(d_{n+1}|\boldsymbol{v}_{n+1}, d_n)p(\boldsymbol{y}_{n+1}|\boldsymbol{v}_{n+1}, h_{n+1}), \tag{5.6}$$

$$p(\boldsymbol{v}_{n+1}|\boldsymbol{z}_n, d_n) = \begin{cases} \delta_{\boldsymbol{v}_n}(\boldsymbol{v}_{n+1}), & d_n > 0 \\ p^*(\boldsymbol{v}_{n+1}|\boldsymbol{v}_n), & d_n = 0 \end{cases}, \tag{5.7}$$

$$p(d_{n+1}|\boldsymbol{v}_{n+1}, d_n) = \begin{cases} \delta_{d_n - 1}(d_{n+1}), & d_n > 0 \\ p(d_{n+1}|\boldsymbol{v}_{n+1}), & d_n = 0 \end{cases}, \tag{5.8}$$

where δ is the Kronecker function ($\delta_a(b) = 1$ for $a = b$ and $\delta_a(b) = 0$ for $a \neq b$).

The properties of the four terms on the right side of Equation (5.6) are clarified in following:

1. $p(\boldsymbol{v}_{n+1}|\boldsymbol{z}_n, d_n)$ is the transition probability of $\boldsymbol{v}_{n+1}$ conditioned on $(\boldsymbol{z}_n, d_n)$. In Equation (5.7), p^* is introduced for representing the transition probability when $d_n = 0$. We can see that $\boldsymbol{v}_{n+1}$ is only probably be different from $\boldsymbol{v}_n$ when $d_n = 0$, otherwise $\boldsymbol{v}_{n+1}$ will be exactly the same as $\boldsymbol{v}_n$. When $d_n = 0$, the transition $p^*(\boldsymbol{v}_{n+1}|\boldsymbol{v}_n)$ behaves the same as the state transition of TMC and TMC-GMM, which means that $\boldsymbol{v}_{n+1}$ can be different from or same as $\boldsymbol{v}_n$, depending on the distribution of $p^*(\boldsymbol{v}_{n+1}|\boldsymbol{v}_n)$.

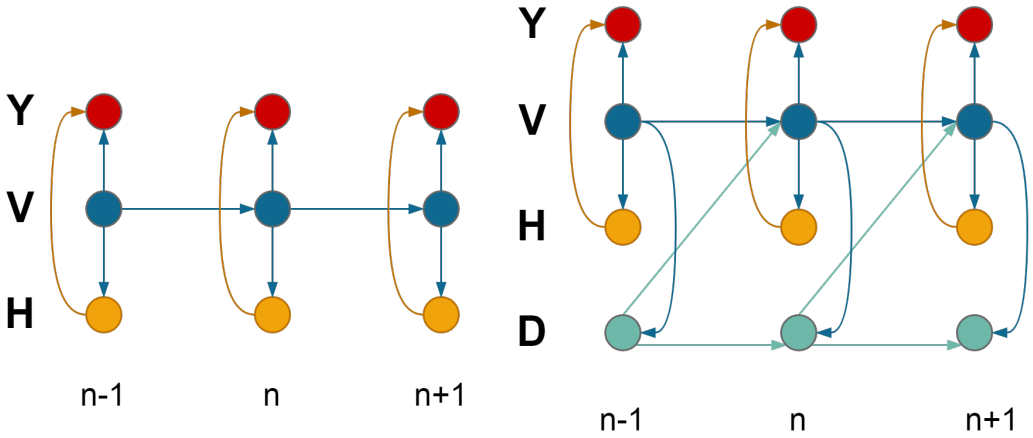
2. $p(d_{n+1}|\boldsymbol{v}_{n+1}, d_n)$ is the probability of the minimal remaining sojourn time of $\boldsymbol{v}_{n+1}$, conditioned on $\boldsymbol{v}_{n+1}$ and d_n.

3. $p(h_{n+1}|\boldsymbol{v}_{n+1})$ and $p(\boldsymbol{y}_{n+1}|\boldsymbol{v}_{n+1}, h_{n+1})$ are same as the ones in TMC-GMM, shown in Equation (5.5).

Now, Equations (5.7) and (5.8) together describe how the hidden states, V_n and D_n, transfer in SemiTMC-GMM.

By comparing the two models, *i.e.* TMC-GMM and SemiTMC-GMM, the dependency graphs are shown in Figure 5.1. The couple $\boldsymbol{V} = (\boldsymbol{X}, \boldsymbol{U})$ is regarded as one hidden state for reducing the complexity of the graphs, also remind that the total number of processes in the three models are 3, 4 and 5 respectively. For

the dependency of TMC in Figure 4.2, the observation $\boldsymbol{Y}$ (red node) is dependent on $\boldsymbol{V}$ (blue node), whereas $\boldsymbol{Y}$ is dependent on both $\boldsymbol{V}$ and $\boldsymbol{H}$ (yellow node) in TMC-GMM and SemiTMC-GMM. Besides, sojourn state $\boldsymbol{D}$ (green node) has no link with $\boldsymbol{Y}$ in SemiTMC-GMM.



(a) TMC-GMM. (b) SemiTMC-GMM.

Figure 5.1: Dependency graph.

Estimating the individual x_n and u_n will be different from the former models, for the sense of introducing the sojourn state D_n. The probabilities of x_n can be obtained by

$$\begin{aligned} p(x_n|\boldsymbol{y}_1^n) &= \sum_{u_n}\sum_{d_n} p(x_n, u_n, d_n|\boldsymbol{y}_1^n), \\ p(x_n|\boldsymbol{y}_1^N) &= \sum_{u_n}\sum_{d_n} p(x_n, u_n, d_n|\boldsymbol{y}_1^N). \end{aligned} \tag{5.9}$$

The probabilities $p(u_n|\boldsymbol{y}_1^n)$ and $p(u_n|\boldsymbol{y}_1^N)$ are obtained in a similar way. Finally, the estimated hidden state $\hat{x}_n$ and $\hat{u}_n$ can be obtained by Equation (5.3).

To summarize, the proposed SemiTMC-GMM is a model contains five stochastic processes $\boldsymbol{X}$, $\boldsymbol{U}$, $\boldsymbol{D}$, $\boldsymbol{H}$, $\boldsymbol{Y}$, with Markov distribution of $\boldsymbol{Z}^* = (\boldsymbol{X}, \boldsymbol{U}, \boldsymbol{D}, \boldsymbol{H}, \boldsymbol{Y})$. The process $\boldsymbol{X}$ models the activities we are looking for, $\boldsymbol{Y}$ models the observation, $\boldsymbol{U}$ models the introduced gait or leg phase, $\boldsymbol{D}$ models the semi-Markovianity of $\boldsymbol{V} = (\boldsymbol{X}, \boldsymbol{U})$, and $\boldsymbol{H}$ models the presence of Gaussian mixtures. Thus, $\boldsymbol{Z}^* = (\boldsymbol{V}, \boldsymbol{Y}, \boldsymbol{W})$ can be regarded as a classic TMC with hidden state $\boldsymbol{V}$, observed $\boldsymbol{Y}$, and an additional latent $\boldsymbol{W} = (\boldsymbol{D}, \boldsymbol{H})$.

5.3 Parameter estimation

From the previous section, it is now clear how the additional hidden states $\boldsymbol{H}$ and $\boldsymbol{D}$ work in SemiTMC-GMM and how to compute the observation probability. In this section, we focus on how to obtain the filtering and smoothing probabilities, and to apply parameter updating based on the on-line EM algorithm.

Before starting the explanation, we need to introduce the parameter set first. As described in the previous section, the parameter set can be defined as $\theta = \{\zeta_k, a_{lk}, c_{ij}, \boldsymbol{\mu}_{ij}, \Sigma_{ij}\}$, in which ζ_k is the initial probability of hidden state, and a_{lk} is the l-th row and k-th column element in the transition matrix A. Because GMM density only depends on $\boldsymbol{v}_n$, then $i \in \Lambda \times \Gamma$, $j \in K$. While in SemiTMC-GMM, the entire hidden state is $(\boldsymbol{V}, \boldsymbol{D})$, then $l, k \in \Lambda \times \Gamma \times L$, and l, k equal to the couple of (i, d_n). Therefore, the initial probability becomes $\zeta_k = p((v_1, d_1) = k)$, and $a_{lk} = p((\boldsymbol{v}_{n+1}, d_{n+1}) = k | (\boldsymbol{v}_n, d_n) = l)$. For simplification, the indices i, j, l, k will keep the same meaning and will no longer be specified in the remaining of this Chapter.

5.3.1 Batch mode EM algorithm

Baum-Whelch algorithm is used again for obtaining parameters. Here we simply describe how to extend it to SemiTMC-GMM model. The forward probability $\alpha_n(k)$ and backward probability $\beta_n(k)$ can be obtained similarly as TMC, according to Section 4.2. Then, expectation step is done by defining the following probabilities:

$$\gamma_n(k) = p((\boldsymbol{v}_n, d_n) = k | \boldsymbol{y}_1^N) = \frac{\alpha_n(k)\beta_n(k)}{\sum\limits_{k' \in \Lambda \times \Gamma \times L} \alpha_n(k')\beta_n(k')}, \tag{5.10}$$

$$\tilde{\gamma}_n(i) = \sum_{d_n} \gamma_n((i, d_n)) = \sum_{d_n} p(\boldsymbol{v}_n = i, d_n | \boldsymbol{y}_1^N), \tag{5.11}$$

$$\tilde{\gamma}_n(i, j) = \tilde{\gamma}(i) \cdot \frac{c_{ij} p(\boldsymbol{y}_n | \boldsymbol{v}_n = i, h_n = j)}{\sum\limits_{j' \in K} c_{ij'} p(\boldsymbol{y}_n | \boldsymbol{v}_n = i, h_n = j')}, \tag{5.12}$$

$$\xi_n(l,k) = \frac{\alpha_n(l) \cdot p\left(\boldsymbol{y}_{n+1}, h_{n+1}, (\boldsymbol{v}_{n+1}, d_{n+1}) = k \mid \boldsymbol{y}_n, h_n, (\boldsymbol{v}_n, d_n) = l\right) \cdot \beta_{n+1}(k)}{\sum\limits_{l',k' \in \Lambda \times \Gamma \times L} \left\{ \alpha_n(l') \cdot p\left(\boldsymbol{y}_{n+1}, h_{n+1}, (\boldsymbol{v}_{n+1}, d_{n+1}) = k' \mid \boldsymbol{y}_n, h_n, (\boldsymbol{v}_n, d_n) = l'\right) \cdot \beta_{n+1}(k') \right\}}. \tag{5.13}$$

$\gamma_n(k)$ is the probability of $(\boldsymbol{v}_n, d_n)$ conditioned on all observed data $\boldsymbol{y}_1^N$. $\tilde{\gamma}_n(k)$ is the marginal probability of $\gamma_n(k)$ over d_n, this probability is the one that we are looking for to estimate the concerning hidden state $\boldsymbol{v}_n$. $\tilde{\gamma}_n(i,j)$ is the probability of each Gaussian component *w.r.t.* $\tilde{\gamma}_n(k)$; this probability helps to compute the parameters related to Gaussian mixture, *i.e.* c_{kj}, $\boldsymbol{\mu}_{kj}$, Σ_{kj}. $\xi_n(l,k)$ is the joint probability of $(\boldsymbol{v}_n, d_n) = l$ and $(\boldsymbol{v}_{n+1}, d_{n+1}) = k$ conditioned on $\boldsymbol{y}_1^N$. Here we give the formula of parameter update by using Equations (5.10)-(5.13):

$$\zeta_k = \gamma_1(k), \tag{5.14}$$

$$a_{lk} = \frac{\sum\limits_{n=1}^{N-1} \xi_n(l,k)}{\sum\limits_{n=1}^{N-1} \gamma_n(l)}, \tag{5.15}$$

$$c_{ij} = \frac{\sum\limits_{n=1}^{N} \tilde{\gamma}_n(i,j)}{\sum\limits_{n=1}^{N} \tilde{\gamma}_n(i)}, \tag{5.16}$$

$$\boldsymbol{\mu}_{ij} = \frac{\sum\limits_{n=1}^{N} \tilde{\gamma}_n(i,j) \boldsymbol{y}_n}{\sum\limits_{n=1}^{N} \tilde{\gamma}_n(i,j)}, \tag{5.17}$$

$$\Sigma_{ij} = \frac{\sum\limits_{n=1}^{N} \tilde{\gamma}_n(i,j) (\boldsymbol{y}_n - \boldsymbol{\mu}_{ij})^{\mathsf{T}} (\boldsymbol{y}_n - \boldsymbol{\mu}_{ij})}{\sum\limits_{n=1}^{N} \tilde{\gamma}_n(i,j)}. \tag{5.18}$$

In fact, Equations (5.10)-(5.13) are the expectation step in one iteration of EM algorithm, while Equations (5.14)-(5.18) are the maximization step. Then, the parameter can be learned by recursively performing the two steps.

5.3.2 On-line estimation

Now, let us consider the statistics of SemiTMC-GMM below:

$$s^{(1)}_{n',lk} = \mathbb{1}\{(\boldsymbol{v}_{n'}, d_{n'}) = l, (\boldsymbol{v}_{n'+1}, d_{n'+1}) = k\}, \tag{5.19}$$

$$s^{(2)}_{n',k} = \mathbb{1}\{(\boldsymbol{v}_{n'}, d_{n'}) = k\}, \tag{5.20}$$

$$s^{(3)}_{n',ij} = \mathbb{1}\{\boldsymbol{v}_{n'} = i, h_{n'} = j\}, \tag{5.21}$$

$$s^{(4)}_{n',ij} = \mathbb{1}\{\boldsymbol{v}_{n'} = i, h_{n'} = j\}\boldsymbol{y}_{n'}, \tag{5.22}$$

$$s^{(5)}_{n',ij} = \mathbb{1}\{\boldsymbol{v}_{n'} = i, h_{n'} = j\}\boldsymbol{y}^{\mathsf{T}}_{n'}\boldsymbol{y}_{n'}, \tag{5.23}$$

where $\mathbb{1}\{\cdot\}$ is the indicator function, $n' = 1, \ldots, N$. Then, the statistics vector at time n' is of the form $s_{n'} = \left\{s^{(1)}_{n',lk}, s^{(2)}_{n',k}, s^{(3)}_{n',ij}, s^{(4)}_{n',ij}, s^{(5)}_{n',ij}\right\}$. Then, denote that $S_n = \left\{S^{(1)}_{n,lk}, S^{(2)}_{n,k}, S^{(3)}_{n,ij}, S^{(4)}_{n,ij}, S^{(5)}_{n,ij}\right\}$, which follow the calculations:

$$S_n = \frac{1}{n}\mathbf{E}_\theta\left(\sum_{n'=1}^{n} s_{n'}\right)\Bigg|\boldsymbol{y}_1^n. \tag{5.24}$$

$$\tilde{S}^{(2)}_{n,i} = \sum_{d_n} S^{(2)}_{n,(i,d_n)}, \tag{5.25}$$

$$\zeta_k = S^{(2)}_{1,k}, \tag{5.26}$$

$$a_{n,lk} = S^{(1)}_{n,lk}\Big/S^{(2)}_{n,k}, \tag{5.27}$$

$$c_{n,ij} = S^{(3)}_{n,ij}\Big/\tilde{S}^{(2)}_{n,i}, \tag{5.28}$$

$$\boldsymbol{\mu}_{n,ij} = S^{(4)}_{n,ij}\Big/S^{(3)}_{n,ij}, \tag{5.29}$$

$$\Sigma_{n,ij} = S^{(5)}_{n,ij}\Big/S^{(3)}_{n,ij} - \boldsymbol{\mu}^{\mathsf{T}}_{n,ij}\boldsymbol{\mu}_{n,ij}, \tag{5.30}$$

In order to apply the on-line estimation, a common way [134] is to update the sufficient statistics when a new observed data come in

$$S_{n+1} = (1 - \rho_{n+1}) \cdot S_n + \rho_{n+1} \cdot \mathbf{E}_{\theta_n}\left(s_{n+1}\Big|\boldsymbol{y}_{n+1}\right), \tag{5.31}$$

where ρ_n is the stepsize sequence that satisfies $\sum_{n=1}^{\infty} \rho_n = \infty$, $\sum_{n=1}^{\infty} \rho_n^2 < \infty$. Normally it is set to $\rho_n = 1/n$. Then, the new parameter θ_{n+1} is available by Equations (5.25)-(5.30). The estimation of x_{n+1}, u_{n+1} can be obtained by Equation (5.9) and (5.3).

Unlike [134], we do not update θ at every sampling time. Instead, we set a window length W_l and accumulate the latest W_l observed data first. Then use Equations (5.10)-(5.13) to get the smoothed result, compute the sequenced statistics $s_n|_1^{W_l}$ for all the W_l data by Equations (5.19)-(5.23). Afterward, update the sequenced sufficient statistics $S_n|_1^{W_l}$ and $\theta_n|_1^{W_l}$ by Equation (5.31) and Equations (5.25)-(5.30), respectively. It should be noticed that in on-line mode, the initial probability ζ_k is not necessary.

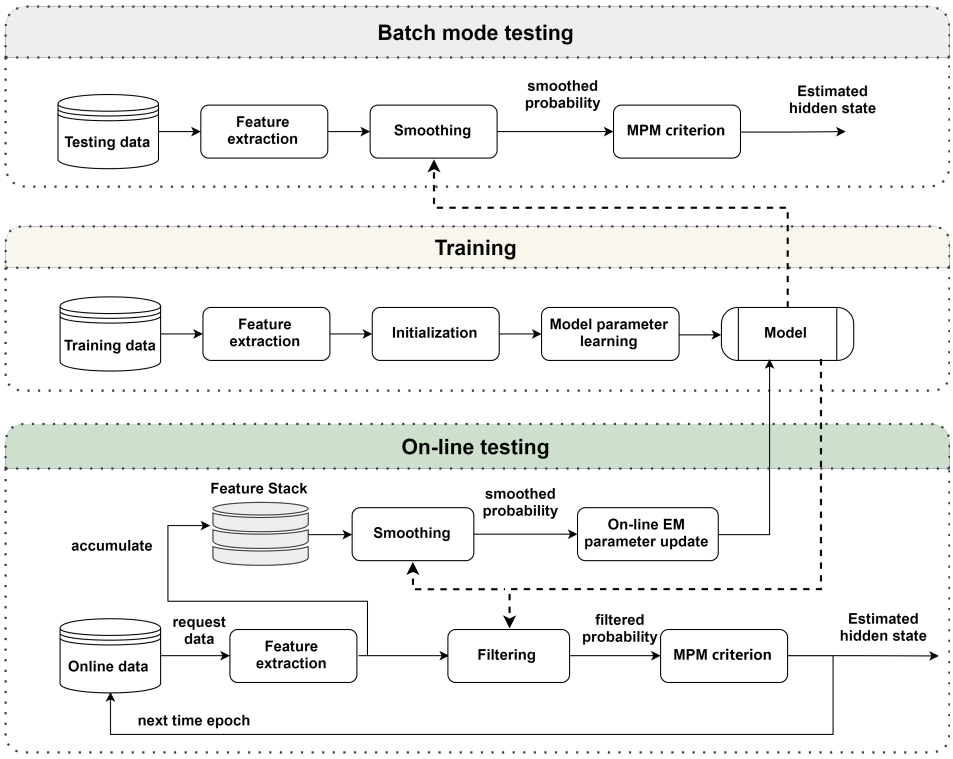


Figure 5.2: SemiTMC-GMM diagram of the traning stage, and the testing stage for both batch mode and on-line testing.

After describing the batch mode and on-line parameter learning of SemiTMC-GMM, a diagram of the training and testing is displayed in Figure 5.2. The model is updated in on-line testing, but not in the batch mode testing. Comparing to the

diagram of TMC-HIST shown in Figure 4.4, there are some differences. The first one is that the estimated hidden state is dependent on the filtering results when using SemiTMC-GMM, but is dependent on the smoothed results of an gait cycle in TMC-HIST. The second one is that there is no gait cycle detection in SemiTMC-GMM, thus the estimated hidden state is given at every sampling time, which does not need to wait the entire gait cycle is completed as TMC-HIST.

5.4 Experimental results

Two datasets are used to validate the proposed algorithm. The first dataset is the Sports and Daily Activities (SDA) dataset [98], in which eight subjects were enrolled to perform 19 daily and sports activities while wearing five Xsens MTx[1] IMUs on their torso, left arm, right arm, left thigh and right thigh (shown in Figure 5.3), all the sensors measure the acceleration and angular rate of where the sensors are placed. The five sensor placements are determined in this dataset because the involved activities are not only lower limb locomotion activities, but also include static activities and upper limb activities, such as sitting, lying, rowing and playing basketball... So they used the 5 sensors to collect the motion data from different part of the body. However, in this paper, the proposed algorithm is designed for recognizing the lower limb locomotion activity with periodic gait or leg cycle performed by healthy people. Since we only care about the movement of the lower limbs and healthy people have a symmetric motion of the two legs, then it is possible to use only one sensor placed on either the left or right leg to recognize the considered activities. Therefore, we only use the sensor placed on the right thigh to validate our algorithm. The sensor sampling rate was set to 25 Hz, the acceleration sensing range was set to $\pm 18g$, the angular rate sensing range was set to $\pm 1200^{\circ}/s$. Because the objective of the proposed algorithm is to detect lower limb locomotion activities that have gait cycle or leg cycle, while the 19 activities consists of both lower limb locomotion activities with and without the cycles, then only 11 suitable activities out of the total are selected in this work: walk in parking

[1]Details of Xsens MTx can be found in http://www.xsens.com/en/general/mtx.

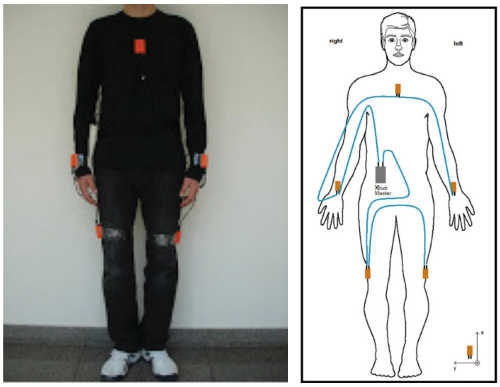


Figure 5.3: The five sensor placements in SDA dataset, the selected sensor is the one placed on the right thigh [98].

lot, walk on treadmill with incline, walk on treadmill on flat, stair descent, stair ascent, run on treadmill, jump, exercise on stepper, exercise of cycling in vertical position, exercise of cycling in horizontal position, exercise on cross trainer. These 11 locomotion activities of SDA dataset are referred to as D1A1 to D1A11 in the remaining of this paper. In the dataset, the subjects performed each activity for about 5 minutes separately, and each activity was divided into 60 segments of 5 seconds. Therefore, there are $60 \times 8 = 480$ segments for each activity. In order to make the dataset available for our algorithm, we firstly combined the 60 segments of one activity from one subject to recover the 5 minutes activity, then combined the data of the same activity from all the subjects to form 40 minutes for each activity, the final data was obtained by combining the 11 activities. Thus, the duration of the data is $11 \times 40 = 440$ minutes.

Unlike the sensor is placed on the foot in the previous chapters, since the IMU sensor is placed on the right thigh in SDA dataset. The segmentation of gait cycle is based on the motion of foot, so similarly we can define 'leg cycle' based on the motion of leg. One leg cycle can be segmented into four leg phases, which are low position, lifting, high position and dropping.

There are only 7500 samplings for each experiment of SDA, the data length is not long enough to use on-line EM recognition. Therefore we utilize the second dataset for the validation of the proposed on-line EM algorithm. This the same dataset that described in Chapter 4.

In Chapter 4, we use both 6D features (angular rate only) and 12D features (angular rate and acceleration) to validate the TMC-HIST model, and showed that only using angular rate is better than using both angular rate and acceleration. So in this Chapter, we also use the 2 kinds of features to see what impacts will be made for different sensor positions. The proposed SemiTMC-GMM model is compared with TMC-GMM to see the advancement of semi-Markov structure in recognizing lower limb locomotion activities. While GMM is implemented by different κ to see the impact of the GMM components number that has on recognition accuracy. TMC-HIST is also compared with SemiTMC-GMM to show the improvement from TMC-HIST to SemiTMC-GMM.

5.4.1 SDA dataset

The batch mode recognition is tested by a leave-one-out cross-validation (LOOCV) strategy, *i.e.* taking one subject for testing and the others for training, then make the test for all the subjects. The sliding window length of feature extraction is set to 5 based on our experience. Both SemiTMC-GMM and TMC-GMM model are involved in the validation, the GMM mixture number κ is set to 1, 3, 6, 9 respectively. Particularly when $\kappa = 1$, the class-conditional observation density yields to the conventional Gaussian distribution.

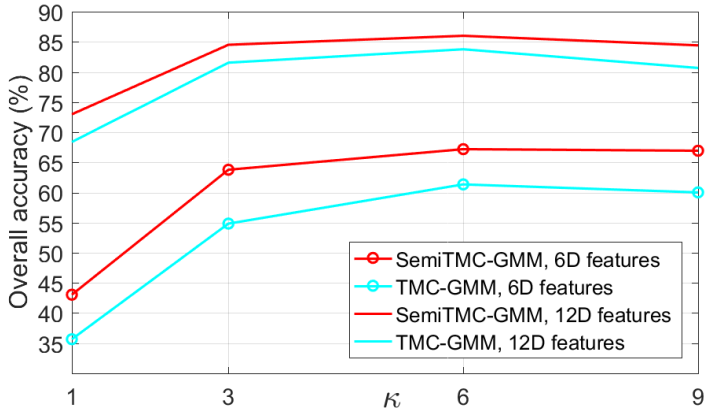


Figure 5.4: The overall batch mode recognition accuracy using different features on SDA dataset, according to different GMM mixture number κ.

The overall accuracy of batch mode recognition on SDA dataset is shown in

Table 5.1: The sensitivity, specificity, F1 score, MCC value of the batch mode recognition using 6D features, for each activity of SDA dataset. **Up**: TMC-HIST; **middle**: TMC-GMM when $\kappa = 6$; **down**: SemiTMC-GMM when $\kappa = 6$.

		Activity					
TMC-HIST		**D1A1**	**D1A2**	**D1A3**	**D1A4**	**D1A5**	**D1A6**
	Sensitivity	0.5877	0.5694	0.6413	0.8175	0.6562	1.0000
	Specifivity	0.9322	0.9520	0.9685	0.9710	0.9053	0.9745
	F1 score	0.5207	0.5571	0.6570	0.7704	0.4955	0.8880
	MCC	0.4698	0.5114	0.6234	0.7486	0.4515	0.8822
		D1A7	**D1A8**	**D1A9**	**D1A10**	**D1A11**	**Total**
	Sensitivity	0.4952	0.4434	0.4865	0.3900	0.1878	0.5705
	Specifivity	0.9906	0.9525	0.9452	0.9529	0.9811	0.9569
	F1 score	0.6229	0.4638	0.4800	0.4207	0.2732	0.5590
	MCC	0.6200	0.4127	0.4266	0.3683	0.2676	0.5257
TMC-GMM		**D1A1**	**D1A2**	**D1A3**	**D1A4**	**D1A5**	**D1A6**
	Sensitivity	0.4549	0.6274	0.6233	0.7901	0.6733	0.9720
	Specifivity	0.9329	0.9772	0.9620	0.9605	0.9152	0.9914
	F1 score	0.4295	0.6772	0.6237	0.7169	0.5256	0.9451
	MCC	0.3689	0.6499	0.5857	0.6907	0.4840	0.9399
		D1A7	**D1A8**	**D1A9**	**D1A10**	**D1A11**	**Total**
	Sensitivity	0.7322	0.4892	0.7492	0.3513	0.3002	0.6148
	Specifivity	0.9940	0.9592	0.9154	0.9765	0.9906	0.9614
	F1 score	0.8170	0.5169	0.5798	0.4439	0.4311	0.6097
	MCC	0.8075	0.4720	0.5444	0.4206	0.4505	0.5831
SemiTMC-GMM		**D1A1**	**D1A2**	**D1A3**	**D1A4**	**D1A5**	**D1A6**
	Sensitivity	0.4707	0.7577	0.5588	0.8636	0.7739	1.0000
	Specifivity	0.9631	0.9657	0.9735	0.9726	0.9071	0.9950
	F1 score	0.5130	0.7226	0.6139	0.8028	0.5634	0.9761
	MCC	0.4708	0.6940	0.5824	0.7850	0.5346	0.9739
		D1A7	**D1A8**	**D1A9**	**D1A10**	**D1A11**	**Total**
	Sensitivity	0.7730	0.6241	0.8797	0.3433	0.3640	0.6735
	Specifivity	0.9891	0.9521	0.9320	0.9903	0.9988	0.9672
	F1 score	0.8211	0.5951	0.6898	0.4772	0.5291	0.6640
	MCC	0.8064	0.5528	0.6709	0.4898	0.5738	0.6486

Table 5.2: Overall performances summary of Table 5.1.

	Sensitivity	**Specificity**	**F1 Score**	**MCC**
TMC-HIST	0.5705	0.9569	0.5590	0.5257
TMC-GMM	0.6148	0.9614	0.6097	0.5831
SemiTMC-GMM	0.6735	0.9672	0.6640	0.6486

Table 5.3: The sensitivity, specificity, F1 score, MCC value of the batch mode recognition using 12D features, for each activity of SDA dataset. **Up**: TMC-HIST; **middle**: TMC-GMM when $\kappa = 6$; **down**: SemiTMC-GMM when $\kappa = 6$.

		Activity					
TMC-HIST		**D1A1**	**D1A2**	**D1A3**	**D1A4**	**D1A5**	**D1A6**
	Sensitivity	0.4900	0.5463	0.6997	0.9017	0.7885	1.0000
	Specificity	0.9392	0.9883	0.9649	0.9839	0.9222	0.9939
	F1 Score	0.4687	0.6574	0.6837	0.8708	0.6057	0.9709
	MCC	0.4128	0.6461	0.6511	0.8587	0.5781	0.9684
		D1A7	**D1A8**	**D1A9**	**D1A10**	**D1A11**	**Total**
	Sensitivity	0.8308	0.7116	0.9489	0.9972	0.6618	0.7797
	Specificity	0.9911	0.9924	1.0000	1.0000	0.9813	0.9779
	F1 Score	0.8654	0.7966	0.9737	0.9986	0.7168	0.7826
	MCC	0.8535	0.7854	0.9715	0.9985	0.6936	0.7652
TMC-GMM		**D1A1**	**D1A2**	**D1A3**	**D1A4**	**D1A5**	**D1A6**
	Sensitivity	0.6784	0.6797	0.5483	0.9146	0.8980	1.0000
	Specificity	0.9322	0.9993	0.9866	0.9689	0.9465	0.9995
	F1 Score	0.5777	0.8059	0.6525	0.8164	0.7305	0.9978
	MCC	0.5353	0.8067	0.6382	0.8025	0.7151	0.9975
		D1A7	**D1A8**	**D1A9**	**D1A10**	**D1A11**	**Total**
	Sensitivity	0.8843	0.8917	0.8602	0.9876	0.8784	0.8383
	Specificity	0.9961	0.9940	0.9987	0.9998	0.9999	0.9838
	F1 Score	0.9197	0.9140	0.9184	0.9930	0.9348	0.8419
	MCC	0.9129	0.9059	0.9132	0.9923	0.9309	0.8319
SemiTMC-GMM		**D1A1**	**D1A2**	**D1A3**	**D1A4**	**D1A5**	**D1A6**
	Sensitivity	0.6672	0.7247	0.6182	0.9638	0.8767	0.9990
	Specificity	0.9457	0.9972	0.9860	0.9773	0.9563	0.9990
	F1 Score	0.6054	0.8273	0.7039	0.8752	0.7509	0.9944
	MCC	0.5644	0.8223	0.6862	0.8666	0.7327	0.9939
		D1A7	**D1A8**	**D1A9**	**D1A10**	**D1A11**	**Total**
	Sensitivity	0.9025	0.9410	0.8561	0.9956	0.9215	0.8606
	Specificity	0.9936	0.9922	0.9996	0.9994	1.0000	0.9860
	F1 Score	0.9175	0.9324	0.9208	0.9948	0.9590	0.8620
	MCC	0.9096	0.9255	0.9165	0.9943	0.9560	0.8516

Table 5.4: Overall performances summary of Table 5.3.

	Sensitivity	**Specificity**	**F1 Score**	**MCC**
TMC-HIST	0.7797	0.9779	0.7826	0.7652
TMC-GMM	0.8383	0.9838	0.8419	0.8319
SemiTMC-GMM	0.8606	0.9860	0.8620	0.8516

Figure 5.4. As it can be seen, SemiTMC-GMM achieves an accuracy improvement of about 6%-7% compared to TMC-GMM when using 6D features, whereas the improvement when using 12D features is 2%-3%. However, using 6D features obtains lower performances than using 12D features. For 6D features, the SemiTMC-GMM model reaches the highest accuracy of 67.19% when $\kappa = 6$, while the one of TMC-GMM is 61.34%. While for the 12D features, SemiTMC-GMM and TMC-GMM obtain the accuracy of 86.00% and 83.76% respectively. As for the TMC-HIST model, it obtains 56.89% and 77.91% recognition accuracy for 6D and 12D features respectively, which are lower than that of SemiTMC-GMM and TMC-GMM. Table 5.1 and 5.3 show the sensitivity, specificity, F1 score, and MCC of each individual activity that using the different features. Particularly for the sensitivity of each individual activity, it equals to the accuracy of corresponding activity. Activities D1A1 to D1A3 are recognized with relatively poor performance, it is because that these three activities are all walking and are very easily misclassified. In a general view of the two Tables, using 12D features is better than using 6D features. It is reasonable for SDA dataset because the selected sensor is placed on the same place (right thigh) and in the same direction of each subject. In Chapter 4, the IMU sensor is placed on the shoes and the distinctions in the front slope of each subject may lead in the distinction of orientation, which may finally have an impact on the acceleration. That is why using 12D features is better for SDA dataset, and using 6D features is better for our own dataset.

As reported in [90], the classifiers of kNN, SVM and decision tree are tested on SDA dataset using all the five sensors. The accuracies are 78.97%, 84.03% and 84.63% respectively. In [91], the authors used SDA dataset and showed single sensor recognition accuracy of four classifiers: kNN, decision tree, discriminant analysis and Naive Bayes. Specifically for the right leg sensor that is used in our paper, the four classifiers obtained accuracies of 81.72%, 78.78%, 87.03%, 76.93%. Therefore, we can state that SemiTMC-GMM outperforms the generic classifiers like kNN, SVM, decision tree and Naive Bayes, and obtains a similar performance of discriminant analysis. On the other hand, the authors in [135] used CNN to recognize human daily activities in OPPORTUNITY dataset [99], which contains

activities such as open (close) door, open (close) drawer, clean table, drink cup... They obtained an accuracy of 85.8% by using 23 body-worn sensors, 12 object sensors and 21 ambient sensors. Also for the OPPORTUNITY dataset, [86] used CNN obtains an accuracy of 77.99% by using the body-worn sensors only. While in [136], CNN obtained an accuracy of 93.75% on six activities: walking, stair ascent, stair descent, sitting, standing and laying. Because of the prevalent CNNs can generate high dimensional features that suit for the recognition task, then CNNs may probably be suited for sophisticated activities. But it requires huge quantity of data to train the network, and it is difficult to make CNN work for adaptive on-line scenario. So, maybe CNN could obtain higher accuracy than SemiTMC-GMM, we still believe that our proposed model is competent in some scenarios.

5.4.2 Our own dataset

For this dataset, the size of sliding window for computing features is set to 15 as the same as Chapter 4. Firstly, the batch mode recognition is performed using LOOCV strategy. Figure 5.5 shows the batch mode recognition accuracy *w.r.t.* different κ that using different features. For the 6D features, the accuracy of SemiTMC-GMM when $\kappa = 9$ is 98.06%, while the one of TMC-GMM is 88.72%. For the 12D features, the accuracy of SemiTMC-GMM when $\kappa = 9$ is 95.16%, while the one of TMC-GMM is 92.57%. Meantime, the choice of κ has less impact on accuracy for SemiTMC-GMM. The recognition accuracy obtained by TMC-HIST are 85.32% and 80.42% respectively for 6D and 12D features, which are lower than the ones of TMC-GMM and SemiTMC-GMM when $\kappa > 1$. Therefore, using SemiTMC-GMM and 6D obtains highest accuracy. Table 5.5 and 5.6 show the sensitivity, specificity, F1 score, and MCC of each individual activity that using different features. By comparing the batch mode recognition shown in Table 5.1 5.3 and 5.5 5.6, both TMC-GMM and SemiTMC-GMM outperform TMC-HIST. It means that considering the observation correlation improves the recognition performance.

There is an irregular behavior when using TMC-GMM. As shown in Figure 5.5 and Tables 5.5, 5.6, the performances of 6D feature is lower than the ones of 12D feature. However in Chapter 4, we have explained why 6D feature is better than

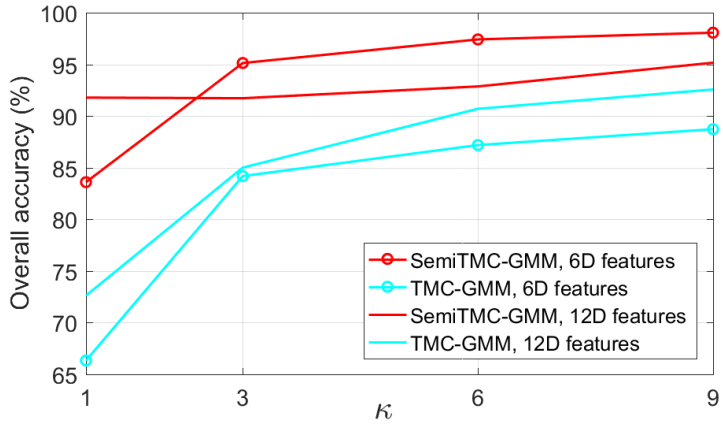


Figure 5.5: The overall batch mode recognition accuracy on LMFIMU dataset, according to different GMM mixture number κ.

Table 5.5: The sensitivity, specificity, F1 score, MCC value of the batch mode recognition using 6D features, for each activity of LMFIMU dataset. Up: TMC-HIST; middle: TMC-GMM when $\kappa = 9$; down: SemiTMC-GMM when $\kappa = 9$.

		Activity				
		Walk	**Run**	**Stair ascent**	**Stair descent**	**Total**
TMC-HIST	**Sensitivity**	0.7900	0.9607	0.8295	0.9307	0.8777
	Specificity	0.9712	0.9063	0.9663	0.9648	0.9521
	F1 Score	0.8681	0.8371	0.8212	0.8683	0.8487
	MCC	0.7763	0.7954	0.7893	0.8474	0.8021
TMC-GMM	**Sensitivity**	0.8967	0.9456	0.8774	0.7750	0.8737
	Specificity	0.9720	0.9996	0.9512	0.9787	0.9754
	F1 Score	0.9536	0.9974	0.9273	0.9706	0.9622
	MCC	0.9220	0.9673	0.7665	0.7938	0.8624
SemiTMC-GMM	**Sensitivity**	0.9887	0.9761	0.9862	0.9537	0.9762
	Specificity	0.9856	0.9992	0.9906	0.9960	0.9929
	F1 Score	0.9869	0.9865	0.9671	0.9644	0.9762
	MCC	0.9743	0.9828	0.9614	0.9587	0.9693

Table 5.6: The sensitivity, specificity, F1 score, MCC value of the batch mode recognition using 12D features, for each activity of LMFIMU dataset. Up: TMC-HIST; middle: TMC-GMM when $\kappa = 9$; down: SemiTMC-GMM when $\kappa = 9$.

		Activity				
		Walk	**Run**	**Stair ascent**	**Stair descent**	**Total**
TMC-HIST	**Sensitivity**	0.7007	0.9721	0.7705	0.9385	0.8454
	Specificity	0.9858	0.8931	0.9174	0.9595	0.9389
	F1 Score	0.8169	0.8258	0.6885	0.8596	0.7977
	MCC	0.7194	0.7833	0.6317	0.8382	0.7431
TMC-GMM	**Sensitivity**	0.9399	0.9475	0.9105	0.8590	0.9142
	Specificity	0.9720	0.9996	0.9512	0.9787	0.9754
	F1 Score	0.9547	0.9723	0.8327	0.8641	0.9060
	MCC	0.9130	0.9654	0.8044	0.8419	0.8812
SemiTMC-GMM	**Sensitivity**	0.9608	0.9829	0.9483	0.8749	0.9417
	Specificity	0.9831	0.9987	0.9634	0.9910	0.9841
	F1 Score	0.9713	0.9891	0.8799	0.9071	0.9368
	MCC	0.9445	0.9861	0.8600	0.8932	0.9210

the 12D feature, and our experiment results show TMC-HIST and SemiTMC-GMM perform better when using 6D feature. But TMC-GMM has an opposite results, *i.e.* using 12D feature is better. It is mainly due to TMC-GMM is not able to detect the gait phases when using 6D features. The Figures 5.6 and 5.7 show the relationship between the estimated gait phases and activity, by using different features, κ is set to 9. Comparing the second line in the two Figures, we can see that the gait detect when using 6D feature is not as good as the one of using 12D feature, and so as to the recognized activity. Since the introduction of the hidden state $\boldsymbol{U}$ (gait cycle) is aiming at improving the accuracy of recognizing activity, then that is why TMC-GMM obtains higher recognition accuracy when using 6D features. In addition, if comparing the second line and third line in Figure 5.6 or Figure 5.7, SemiTMC-GMM obtains better gait cycle than that of TMC-GMM, which results in higher accuracy of activity recognition. Also, the recognition accuracy improvements from TMC-GMM to SemiTMC-GMM are 9.34% and 2.59% respectively for the 6D and 12D features. Because of the highest accuracy and bigger improvement obtained by using 6D features, we can conclude that only considering angular rate is more

proper for the proposed SemiTMC-GMM model.

As a matter of fact, Figures 5.4, 5.5 and 5.8 show that introducing semi-Markov structure into the TMC model can improve the accuracy. Meanwhile, using GMM with $\kappa > 1$ also improves the recognition significantly. But it does not mean that using a larger κ allows higher accuracy to be achieved. In Figure 5.4, the accuracy when $\kappa = 9$ is slightly lower than that obtained when $\kappa = 6$, it is because the observation of SDA dataset is more closer to a GMM mixture of 6 densities. A too much larger κ may probably lead to an overfitting problem. It is sure that κ can be automatically acquired through the methods such as BIC [137] and AIC [138], to make κ consistent with different activities. While for simplification in this thesis, we manually set κ to 6 for all the activities based on the experimental results.

Then, the on-line EM algorithm is performed to validate the adaptive on-line recognition performances. The proposed algorithm is implemented in Matlab code, running on a 64-bit system computer with $3.2GHz$ CPU and 32G RAM. In the dataset, the average experiment time is 32.33 minutes, while the computing time of SemiTMC-GMM when $\kappa = 1, 3, 6, 9$ are 9.72, 14.72, 21.53 and 27.65 minutes respectively. Thus, using on-line EM is applicable in on-line scenarios. The window length W_l for updating the parameters is set to 1000, which means that parameters are updated every 10 seconds.

Figure 5.8 shows the on-line recognition accuracy obtained by LOOCV strategy. The solid lines are higher than the dashed lines which means that the on-line EM algorithm can improve the recognition performance. Also the GMM with $\kappa > 1$ can significantly improve the accuracy. For the 6D feature (Figure 5.8a), when $\kappa = 9$, SemiTMC-GMM has an accuracy improved from 97.22% in the first section to 98.21% in the second section, while TMC-GMM achieves an improvement from 89.58% to 91.12%. For the 12D feature (Figure 5.8b), when $\kappa = 9$, SemiTMC-GMM has an accuracy improved from 95.48% in the first section to 96.93% in the second section, while TMC-GMM achieves an improvement from 93.83% to 95.04%. Therefore, using SemiTMC-GMM and 6D features outperforms the others. By contrast, the adaptive on-line algorithm using the TMC-HIST in Chapter 4, the accuracy was improved from 98.11% to 99.17% (95.32% to 96.93%) using 6(12)D

Figure 5.6: The batch mode estimated activity and gait cycle, using 6D features. κ is set to 9 in TMC-GMM and SemiTMC-GMM.

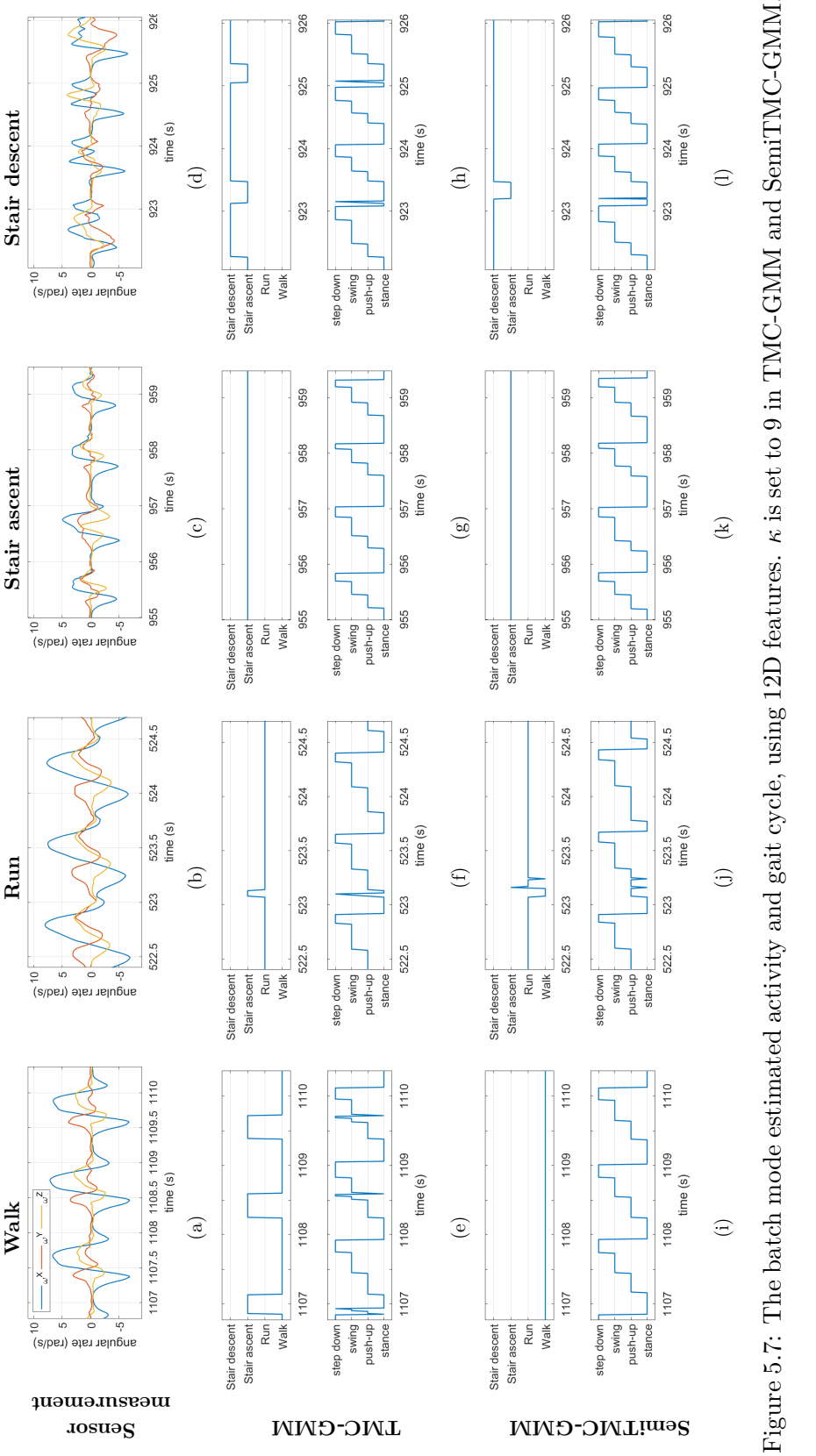
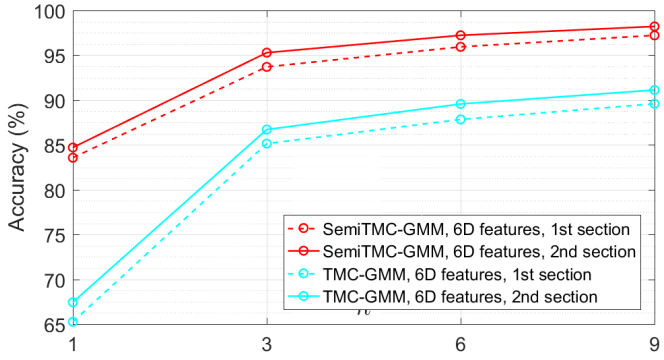
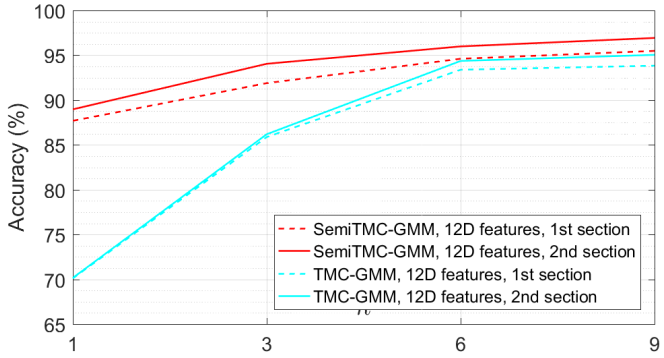


Figure 5.7: The batch mode estimated activity and gait cycle, using 12D features. κ is set to 9 in TMC-GMM and SemiTMC-GMM.



(a)



(b)

Figure 5.8: The on-line mode recognition accuracy of the two experiment sections in our own dataset, according to different GMM mixture number κ. (a): using 6D features; (b): using 12D features.

features. However, this high accuracy is mainly because of the gait cycle complete detection in the adaptive on-line algorithm, which manually sets the activity of all the samplings in one gait cycle to be identical. If without using the gait cycle complete detection, TMC-HIST will fail in the on-line recognition, with the accuracies of 86.37% (78.32%) in the first section and 76.46% (65.20%) in the second section using 6(12)D features. Table 5.7 and 5.8 show the sensitivity, specificity, F1 score, and MCC of the two experiment sections that using different features, the results of each individual activity also indicate that the proposed model can improve the accuracy gradually. Comparing SemiTMC (when $\kappa = 1$) and TMC-HIST, we can conclude that semi-Markov structure is more robust for recognizing the hidden states which have sojourn time. Therefore, the results indicate that both

GMM density and semi-Markov structure improve the on-line recognition, and the combination the two improves the performance the most.

By comparing the two different sensor placements in SDA and our own dataset, the proposed algorithm shows that the sensor is not necessary to be placed at a specific place of the lower limb. In fact, the sensor can be placed in any position that implies the introduced gait phase and leg phase.

In order to understand the dynamic performance of the parameter updating, Figure 5.9 shows the recognition accuracy computed during the latest 10 seconds, using the 6D features. Notice that some lines of $\kappa = 1$ and $\kappa = 3$ are not displayed in Figures 5.9a, 5.9g, 5.9h because the accuracies are lower than 70% for the corresponding activities. SemiTMC-GMM obtains a relatively fast convergence rate when κ equals to 6 and 9. The activities walking and running reach high accuracy within 20 seconds in the first section of the experiment, 99.24% and 98.51% respectively. By contrast, stair ascent and descent are slower (take about 100 seconds) than the former two activities, and obtain lower accuracies of 96.40% and 96.22% respectively. The main reason of this phenomenon is that the activity patterns of stair ascent and descent vary much more differently among the subjects. But in a general view, we can still state that the on-line EM algorithm can dynamically improve the recognition accuracy to a reasonable level.

Figure 5.10 displays the estimated gait cycles of each activity that using 6D features. The gait cycles are obtained when the model converged, κ is set to 1 and 6 for both TMC-GMM and SemiTMC-GMM. ω^x, ω^y and ω^z are the sliding mean of angular rate along the three axes of sensor. The features are 6D, but here we only display the acceleration of along the three axes to show how the gaits proceed. Hence, the estimated gait cycles are displayed *w.r.t.* four models, *i.e.* TMC, SemiTMC, TMC-GMM and SemiTMC-GMM. In fact, the gait phases or leg phases are introduced in the model to improve the recognition accuracy of the lower limb locomotion activity. The Figure shows that SemiTMC-GMM obtains the most regular gait cycle, with no fluctuation in short period and no missing detection. As a consequence, the well estimated gait or leg cycle obtained from SemiTMC-GMM leads to a higher activity recognition performance.

Table 5.7: The sensitivity, specificity, F1 score, MCC value of the on-line mode recognition using 6D features, for each activity of our own dataset. Up: TMC-HIST; middle: TMC-GMM when $\kappa = 9$; down: SemiTMC-GMM when $\kappa = 9$.

Activity (TMC-HIST, with gait complete detection)										
	First section					Second section				
	Walk	**Run**	**Stair ascent**	**Stair descent**	**Total**	**Walk**	**Run**	**Stair ascent**	**Stair descent**	**Total**
Sensitivity	0.9870	0.9957	0.9681	0.9512	0.9755	0.9929	0.9979	0.9872	0.9832	0.9903
Specificity	0.9974	0.9947	0.9894	0.9948	0.9941	0.9975	0.9981	0.9958	0.9978	0.9973
F1 Score	0.9922	0.9884	0.9540	0.9595	0.9735	0.9951	0.9956	0.9819	0.9848	0.9894
MCC	0.9846	0.9852	0.9461	0.9529	0.9672	0.9906	0.9944	0.9787	0.9823	0.9865

Activity (TMC-GMM)										
	First section					Second section				
	Walk	**Run**	**Stair ascent**	**Stair descent**	**Total**	**Walk**	**Run**	**Stair ascent**	**Stair descent**	**Total**
Sensitivity	0.9064	0.9316	0.8902	0.8093	0.8844	0.9029	0.9561	0.9075	0.8738	0.9100
Specificity	0.9626	0.9995	0.9269	0.9737	0.9657	0.9731	0.9995	0.9389	0.9734	0.9712
F1 Score	0.9323	0.9637	0.7695	0.8218	0.8718	0.9348	0.9767	0.8080	0.8596	0.8948
MCC	0.8708	0.9550	0.7330	0.7932	0.8380	0.8799	0.9706	0.7756	0.8359	0.8655

Activity (SemiTMC-GMM)										
	First section					Second section				
	Walk	**Run**	**Stair ascent**	**Stair descent**	**Total**	**Walk**	**Run**	**Stair ascent**	**Stair descent**	**Total**
Sensitivity	0.9926	0.9710	0.9458	0.9296	0.9598	0.9924	0.9851	0.9640	0.9622	0.9759
Specificity	0.9729	0.9996	0.9878	0.9959	0.9891	0.9837	0.9997	0.9936	0.9956	0.9931
F1 Score	0.9827	0.9846	0.9382	0.9512	0.9642	0.9875	0.9920	0.9643	0.9677	0.9779
MCC	0.9656	0.9806	0.9275	0.9438	0.9544	0.9758	0.9897	0.9578	0.9623	0.9714

Table 5.8: The sensitivity, specificity, F1 score, MCC value of the on-line mode recognition using 12D features, for each activity of our own dataset. Up: TMC-HIST; middle: TMC-GMM when $\kappa = 9$; down: SemiTMC-GMM when $\kappa = 9$.

Activity (TMC-HIST, with gait complete detection)										
	First section					**Second section**				
	Walk	**Run**	**Stair ascent**	**Stair descent**	**Total**	**Walk**	**Run**	**Stair ascent**	**Stair descent**	**Total**
Sensitivity	0.9848	0.9956	0.9542	0.9250	0.9649	0.9956	0.9986	0.9855	0.9765	0.9891
Specificity	0.9889	0.9975	0.9845	0.9942	0.9913	0.9963	0.9996	0.9949	0.9983	0.9973
F1 Score	0.9867	0.9933	0.9337	0.9436	0.9643	0.9958	0.9987	0.9787	0.9830	0.9890
MCC	0.9737	0.9915	0.9223	0.9348	0.9556	0.9919	0.9983	0.9748	0.9802	0.9863

Activity (TMC-GMM)										
	First section					**Second section**				
	Walk	**Run**	**Stair ascent**	**Stair descent**	**Total**	**Walk**	**Run**	**Stair ascent**	**Stair descent**	**Total**
Sensitivity	0.9689	0.9345	0.9507	0.8239	0.9195	0.9641	0.9536	0.9804	0.8673	0.9413
Specificity	0.9797	0.9996	0.9518	0.9883	0.9799	0.9926	0.9997	0.9571	0.9892	0.9847
F1 Score	0.9740	0.9655	0.8527	0.8694	0.9154	0.9778	0.9758	0.8840	0.8979	0.9339
MCC	0.9487	0.9572	0.8304	0.8511	0.8969	0.9580	0.9695	0.8667	0.8823	0.9191

Activity (SemiTMC-GMM)										
	First section					**Second section**				
	Walk	**Run**	**Stair ascent**	**Stair descent**	**Total**	**Walk**	**Run**	**Stair ascent**	**Stair descent**	**Total**
Sensitivity	0.9793	0.9934	0.9371	0.8282	0.9345	0.9833	0.9988	0.9585	0.8879	0.9571
Specificity	0.9756	0.9995	0.9648	0.9971	0.9843	0.9852	0.9994	0.9760	0.9973	0.9895
F1 Score	0.9773	0.9958	0.8753	0.8975	0.9365	0.9837	0.9984	0.9167	0.9326	0.9579
MCC	0.9549	0.9947	0.8547	0.8864	0.9227	0.9685	0.9979	0.9021	0.9236	0.9480

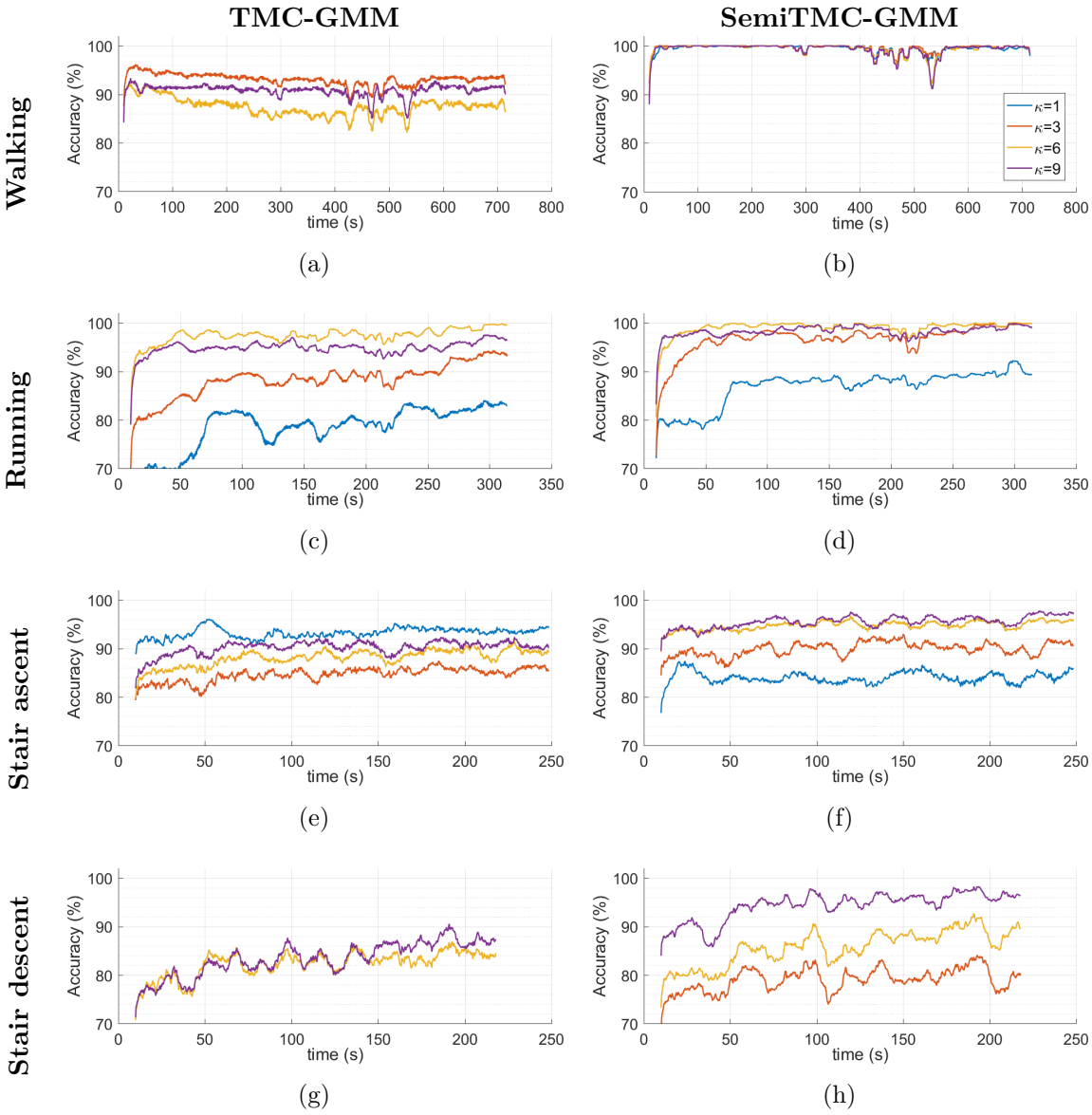
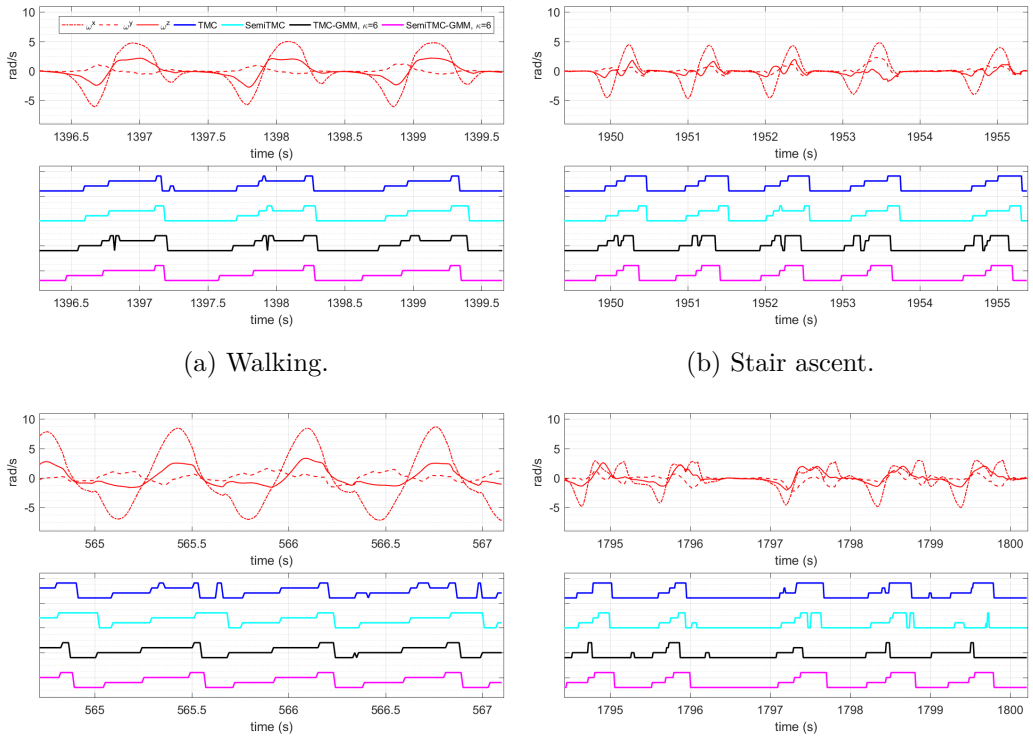


Figure 5.9: Recognition accuracy computed in the latest 10 seconds *w.r.t.* each activity of LMFIMU dataset, using 6D features. Left column: TME-GMM, right column: SemTMC-GMM.



(a) Walking.

(b) Stair ascent.

(c) Running.

(d) Stair descent.

Figure 5.10: Estimated gait cycle of each activity, using 6D features. The blue, cyan, black and magenta represent the gait obtained by TMC, SemiTMC, TMC-GMM and SemiTMC-GMM respectively.

5.5 Conclusion

This Chapter proposes a SemiTMC-GMM model to recognize activities and gait (leg) phases. Comparing with the non-parametric TMC-HIST that proposed in Chapter 4, two different aspects are introduced in SemiTMC-GMM: i), semi-Markov structure is added to more naturally replicate the realistic transition of activity and gait (leg) phase; ii), the non-parametric class-conditional observation density is replaced by a parametric GMM density. Under the scenario of this thesis, our experimental results show that semi-Markov model outperforms the conventional Markov model. Due to the sojourn state $\boldsymbol{D}$ enables the hidden state $\boldsymbol{V}$ to keep the same in a short period of time, the estimated hidden states by semi-Markov model are much more stable than the one of conventional Markov model. On the other hand, GMM densities in SemiTMC-GMM enable the model to take the correlation among

the dimensions of $\boldsymbol{Y}$ into consideration, which is not considered in TMC-HIST. The batch mode recognition over the two datasets proves that the combination of using semi-Markov and GMM can significantly improve the recognition performance. Besides, thanks to the parametric GMM density, on-line EM algorithm is able to be applied for SemiTMC-GMM. The on-line recognition results show that on-line EM can adjust the parameters correctly, the recognition accuracy can be gradually improved to a high level. By comparing the on-line recognition between TMC-HIST and SemiTMC-GMM, both of them can recognize the activities correctly, and the detected gait phases become more regular after parameter update. But TMC-HIST needs the help of gait complete detection to successfully recognize the activity, otherwise it may fail. This phenomenon indicates that semi-Markov model is more suitable for recovering the hidden state with sojourn time.

Chapter 6

Conclusion and perspectives

Quantified-self applications are gaining great research interest recently and Markov models are widely used in many fields. This thesis focuses on using recent Markov model-based methods to study the detection and recognition problems of lower limb locomotion activities, with the help of a unique IMU sensor placed on the lower limb.

The main contributions of this thesis are divided in three points:

(1) A specific Left-Right Hidden Markov Chain (LR-HMC) is developed to detect the four gait phases of lower limb locomotion activities, *i.e.* stance, push-up, swing and step down. In LR-HMC, the hidden state $\boldsymbol{U}$ represents the gait phase, the observation $\boldsymbol{Y}$ comes from the feature extraction of sensor acceleration and angular rate. An unsupervised EM (Expectation-Maximization algorithm)-based parameter learning algorithm is described. Four experiments on four individual activities (walking, running, stair ascent and stair descent) are conducted to validate the performance of LR-HMC *w.r.t.* a competitive method called threshold method. The LR-HMC model outperforms the threshold method in detecting the gait phases of the four individual activities with two advantages: (i), LR-HMC can detect four gait phases whereas the threshold method can only detects two (stance and swing); (ii), LR-HMC does not need to set the parameters manually, while threshold method needs to manually tune the parameters for each activity. But there are still some limitations for LR-HMC:

   (i) The model is stationary. Thus, one trained model is only able to detect the gait phases when the activity pattern keeps the same, *i.e.* from

the same subject, at the same speed, stride length... Otherwise, the performance will be reduced.

(ii) One LR-HMC model is only able to detect the gait phases of one individual activity, it is not suitable to use LR-HMC in a multi-activity scenario.

(iii) The Gaussian density used for representing class-conditional observation density is not very suitable, especially for the task of detecting gait phases over a wide range of activity patterns.

(2) Hence, a non-parametric triplet Markov chain (TMC-HIST) is then proposed to recognise lower limb locomotion activity and gait phases simultaneously. An auxiliary hidden state $\boldsymbol{X}$ is introduced into the model to represent the activity, the couple $\boldsymbol{V} = (\boldsymbol{X}, \boldsymbol{U})$ is then the hidden state to be recovered from observations $\boldsymbol{Y}$. A specific transition graph is designed to mimic the real transition among activities and gait phases. Histograms are used in TMC-HIST to represent the class-conditional observation density. This kind of density allows the class-conditional density to suit for a wide range of activity patterns. Unlike parametric density, since the EM algorithm is not suitable for non-parametric density, the ICE principle is adopted to learn the stationary model. A TMC-HIST-based adaptive on-line algorithm is then proposed to enable the model to recognise activity and gait phases at run-time, and to adaptively adjust model parameters. This adaptation property allows the model to be performed under a non-stationary scenario (speed up or slow down).

The algorithm is validated over a dataset from 10 subjects conducting four activities: walking, running, stair ascent and stair descent. The results show that the proposed algorithm obtains high activity recognition accuracy after the parameters are adjusted to the pattern of activity. Meantime, the gait phases are also well detected.

However, there are two drawbacks of using TMC-HIST:

(i) The remaining sojourn time of the hidden state ($\boldsymbol{V}$) in TMC-HIST follows a geometric distribution, what seems to be inconsistent with the intended application.

(ii) Because of the massive The histograms used in TMC-HIST do not consider the correlation between each dimension of the observation $\boldsymbol{Y}$, which can lead to a loss of information of the class-conditional observation density.

(3) Finally, a parametric triplet semi Markov chain that uses GMM (Gaussian mixture model) density as class-conditional observation density (SemiTMC-GMM) is proposed. In SemiTMC-GMM, semi-Markov structure is utilized to better replicate the transition of hidden states that has non-geometrical sojourn time, an auxiliary state $\boldsymbol{D}$ is introduced to modelize semi-Markov structure. GMM density is utilized in SemiTMC-GMM to not only approximate the non-Gaussian density, but also to take the correlation among the dimensions of $\boldsymbol{Y}$ into consideration. Another auxiliary state $\boldsymbol{H}$ is introduced to modelize the mixture component in GMM density. Because of the parametric density, an on-line EM algorithm is adapted to the model. The on-line EM algorithm enables the model to recognize activity and gait (leg) phase at run-time, and adaptively adjust the model as well.

The algorithm is validated over two datasets, the results show that SemiTMC-GMM outperforms the TMC-HIST in both batch mode and on-line mode recognition. Also it shows that semi-Markov model provides better performance than the conventional Markov model, meanwhile GMM is more suitable for representing the class-conditional density than conventional Gaussian density.

Our work in this thesis achieves good results, but our proposed algorithms still have some limitations:

(1) Only motion activities are considered in this thesis. The static activity is not considered, such as standing, sitting and lying... These kind of activities are

also very important in our daily life. Adding static activities into the model is quite straight forward, as we only need to calculate features that enable to distinguish between mobility and immobility. For example, the standard deviation of acceleration and angular rate is close to zero when a person is static, otherwise it varies according to the motion [139].

(2) Non-periodic lower limb locomotion activities are not considered. Even though this kind of activity does not take place very often, it is still worth of investigating. For example, fall detection monitoring device can help people when there is an emergency. Therefore, distinguishing periodic and non-periodic motions could be of interest. Possible periodic pattern mining methods suit for this task are fast Fourier transform-based [140] and principle component analysis-based [141] approaches.

(3) In this thesis, the sensors are tightly fixed on the body part (on the shoe or the thigh). The proposed algorithms assume that the orientation are not changing during the motion, because a change in position and/or orientation may significantly affect the performances of the algorithms. Besides, tightly fixing the sensor on our body is not very practical in daily life. On contrary, the sensors are not placed on the same part of body and are normally not tightly fixed, such as a smartphone placed in our pocket. So, it is of great interest to investigate how to make it insensitive to orientation. A feasible method consists in using a coordination transformation method to transfer the signals measured in sensor coordinate system to a global coordinate system, which does not change during the motion. Then feature extraction will be conducted on the transferred signals.

(4) Only one single IMU sensor is considered in our proposed algorithms. However, more sensors are suitable in the algorithms, which may provide a possibility to recognize the imbalanced motion between the two lower limbs of impaired people, such as investigating the freezing of gait of patients who have Parkinson's disease [110]. Or, data fusion can be introduced into the system that has different types of sensors [142], which can contain more information

from different aspects, such as kinematics from IMU sensor, EMG signals, pressure sensor in the insole.

This thesis proposes some algorithms to recognize activities and gait phases, this fundamental functionality of motion analysis is of great importance and can be further used in many other research fields. The pedestrian navigation algorithms [60] detect the stance gait phase to reduce the integration error of positioning. In these algorithms, the stance phase provides the ground truth of velocity (zero velocity) in a short period of one gait cycle, then it is possible to reduce the integration error according to the ground truth. Normally, using stance phase to detect zero velocity duration and reducing the positioning error is called zero velocity update (ZUPT) [58]. However, to the best of our knowledge, most of the pedestrian navigation algorithms only considered walking, while our algorithms provides a way to navigate over different activities. In another research field, precise gait detection and analysis have been used in sports training. For example, assessing the equilibrium between the two lower limbs of a runner, and evaluating the pose of gait. This kind of application can be helpful to refine the training effect, but normally the assessment is done after the training and the entire data is collected, the refinement can only be done in the next time of training. However, our adaptive on-line algorithms may probably enable the athlete to adjust the motion timely, make sports training more efficient and productive.

Appendix A

Likelihood maximization in Baum-Welch algorithm

Based on Equation (3.1), assuming that $\theta \in \Theta$, the we derive the claimed calculations of the Baum-Welch algorithm suited for the parameter updating,

$$\mathcal{L}_{\theta'}(u_1^N, \boldsymbol{y}_1^N, \theta) = \mathbf{E}\left(\ln p(u_1^N, \boldsymbol{y}_1^N|\theta)|\boldsymbol{y}_1^N, \theta'\right), \tag{A.1}$$

in which

$$\begin{aligned}\ln p(u_1^N, \boldsymbol{y}_1^N|\theta) &= \ln\left(p(u_1)\cdot p(y_1|u_1)\prod_{n=1}^{N-1} p(u_{n+1}|u_n)\cdot p(\boldsymbol{y}_n|u_n)\right)\\ &= \ln p(u_1) + \sum_{n=1}^{N-1}\ln p(u_{n+1}|u_n) + \sum_{n=1}^{N}\ln p(\boldsymbol{y}_n|u_n).\end{aligned} \tag{A.2}$$

Then the likelihood $\mathcal{L}_{\theta'}$ can be reformed as $\mathcal{L}_{\theta'} = \mathcal{L}_{\theta'0} + \mathcal{L}_{\theta'1} + \mathcal{L}_{\theta'2}$, where

$$\mathcal{L}_{\theta'0} = \mathbf{E}\left(\ln p(u_1)|\boldsymbol{y}_1^N, \theta'\right) = \sum_{k=1}^{N-1}\gamma_1(k)\log\zeta_k, \tag{A.3}$$

$$\mathcal{L}_{\theta'1} = \mathbf{E}\left(\ln\prod_{n=1}^{N-1} p(u_{n+1}|u_n)\right) = \sum_{n=1}^{N-1}\sum_{k=1}^{\tau}\sum_{l=1}^{\tau}\xi_n(l,k)\ln a_{lk}, \tag{A.4}$$

$$\begin{aligned}\mathcal{L}_{\theta' 2} &= \mathbf{E}\left(\ln \prod_{n=1}^{N} p(\boldsymbol{y}_n|u_n)\right) \\ &= \sum_{n=1}^{N}\sum_{k=1}^{\tau} \gamma_n(k) \ln\left(\frac{1}{\sqrt{(2\pi)^w|\Sigma_k|}} \exp\left(-\frac{1}{2}(\boldsymbol{y}_n - \boldsymbol{\mu}_k)^\intercal \Sigma_k^{-1}(\boldsymbol{y}_n - \boldsymbol{\mu}_k)\right)\right) \\ &= -\frac{1}{2}\sum_{n=1}^{N}\sum_{k=1}^{\tau} \gamma_n(k)\left(w\ln(2\pi) + \ln|\Sigma_k| + (\boldsymbol{y}_n - \boldsymbol{\mu}_k)^\intercal \Sigma_k^{-1}(\boldsymbol{y}_n - \boldsymbol{\mu}_k)\right).\end{aligned} \tag{A.5}$$

in which, $l, k \in \Gamma$, and w is the observation dimension. $\gamma_n(k)$ and $\xi_n(l,k)$ are the expectations of u_n and (u_n, u_{n+1}) conditionally to $\boldsymbol{y}_1^N$ obtained in the expectation step of Baum-Welch algorithm, through Equations (3.9) and (3.10). $|\Sigma_k|$ is the determinant of matrix Σ_k.

In fact, the calculations of log-likelihood function, using the Equations (A.1) to (A.5), involve the expectation step. Then we are able to solve the maximization problematic through partial differential over θ.

For $\boldsymbol{\mu}_k$:

Through the equilibrium formula of

$$\frac{\partial \mathcal{L}_{\theta'}}{\partial \boldsymbol{\mu}_k} = 0, \tag{A.6}$$

there is

$$\begin{aligned}&\frac{\partial \mathcal{L}_{\theta' 2}}{\partial \boldsymbol{\mu}_k} = 0 \\ \Rightarrow \qquad &\boldsymbol{\mu}_k = \frac{\sum_{n=1}^{N} \gamma_n(k)\boldsymbol{y}_n}{\sum_{n=1}^{N} \gamma_n(k)}.\end{aligned} \tag{A.7}$$

For Σ_k:

Similar as $\boldsymbol{\mu}_k$, Σ_k can be obtained through

$$\begin{aligned} & \frac{\partial \mathcal{L}_{\theta' 2}}{\partial \Sigma_k} = 0 \\ \Rightarrow \qquad \Sigma_k \quad &= \frac{\sum\limits_{n=1}^{N} \gamma_n(k) \boldsymbol{y}_n \boldsymbol{y}_n^{\intercal}}{\sum\limits_{n=1}^{N} \gamma_n(k)} - \boldsymbol{\mu}_k \boldsymbol{\mu}_k^{\intercal}. \end{aligned} \tag{A.8}$$

For ζ_k:

Given the constraint $\sum_{k=1}^{\tau} \zeta_k = 1$, we can then introduce Lagrange multiplier λ to obtain the Lagrange function, and the partial differential of the function is

$$\frac{\partial \left\{ \mathcal{L}_{\theta'} + \lambda(1 - \sum\limits_{k=1}^{\tau} \zeta_k) \right\}}{\partial \zeta_k} = 0. \tag{A.9}$$

From the equation above we obtain $\zeta_k = \gamma_1(k)$.

For a_{lk}:

Similar, given the constraint $\sum_{k=1}^{m} a_{lk} = 1$, then there is

$$\frac{\partial \left\{ \mathcal{L}_{\theta'} + \lambda(1 - \sum\limits_{k=1}^{\tau} a_{lk}) \right\}}{\partial a_{lk}} = 0. \tag{A.10}$$

From the equation above we also get

$$a_{lk} = \frac{\sum\limits_{n=1}^{N-1} \xi_n(l,k)}{\sum\limits_{n=1}^{N-1} \gamma_n(l)}.$$

All the re-estimation equations lead to a possibly local maximum of the likelihood.

Publication

– **Journals**

- H. Li, S. Derrode, and W. Pieczynski. Lower limb locomotion activity recognition using semi-Markov model and single wearable inertial sensor. *Sensors*, 19:4242, 2019
- H. Li, S. Derrode, and W. Pieczynski. An adaptive and on-line IMU-based locomotion activity classification method using a triplet Markov model. *Neurocomputing*, 362:94 – 105, 2019

– **Conferences**

- H. Li, S. Derrode, L. Benyoussef, and W. Pieczynski. Free-walking 3D pedestrian large trajectory reconstruction from IMU sensors. In *26th European Signal Processing Conference (EUSIPCO)*, pages 657–661, Rome, Italy, September 2018. IEEE
- H. Li, S. Derrode, L. Benyoussef, and W. Pieczynski. Unsupervised pedestrian trajectory reconstruction from IMU sensors. Hammamet, Tunisia, Apr 2018

List of Tables

List of Figures

Bibliography

[1] Deborah Lupton. *The quantified self*. John Wiley & Sons, 2016. 1

[2] Heikkinen Mikko-Pekka. Self-tracking trending up in finland. *Helsinki Times*, Jun 2014. 1

[3] I. Gorynin, H. Gangloff, E. Monfrini, and W. Pieczynski. Assessing the segmentation performance of pairwise and triplet Markov models. *Signal Processing*, 145:183–192, 2018. 3

[4] I. Gorynin, E. Azeraf, W. Sabbagh, E. Monfrini, and W. Pieczynski. Optimal filtering in hidden and pairwise Gaussian Markov systems. *International journal of mathematical and computational methods (IJMCM)*, 1:259–263, 2016. 3

[5] M. E. Y. Boudaren, E. Monfrini, W. Pieczynski, and A. Aissani. Phasic triplet Markov chains. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(11):2310–2316, 2014. 3

[6] U. Martinez-Hernandez and A. A. Dehghani-Sanij. Adaptive Bayesian inference system for recognition of walking activities and prediction of gait events using wearable sensors. *Neural Networks*, 102:107 – 119, 2018. 8, 15, 23, 74, 76

[7] H. Li, S. Derrode, and W. Pieczynski. An adaptive and on-line IMU-based locomotion activity classification method using a triplet Markov model. *Neurocomputing*, 362:94 – 105, 2019. 8, 15, 34, 83, 119

[8] T. T. Ngo, Y. Makihara, H. Nagahara, Y. Mukaigawa, and Y. Yagi. Similar gait action recognition using an inertial sensor. *Pattern Recognition*, 48(4):1289–1301, 2015. 8, 15

[9] M. Derawi and P. Bours. Gait and activity recognition using commercial phones. *Computers & Security*, 39:137–144, 2013. 8, 15

[10] M. Langer, S. Kiesel, C. Ascher, and G. F. Trommer. Deeply coupled GPS/INS integration in pedestrian navigation systems in weak signal conditions. In *2012 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–7. IEEE, 2012. 9, 20

[11] M. Alfakih, M. Keche, and H. Benoudnine. A new Wi-Fi/GPS fusion method for robust positioning in urban environments. *Physical Communication*, 31:10–20, 2018. 9, 20

[12] K. Shelley, S. Shelley, and C. Lake. Pulse oximeter waveform: photoelectric plethysmography. *Clinical Monitoring*, pages 420–428, 2001. 9

[13] H. Djelouat, A. Amira, F. Bensaali, C. Kotronis, E. Politi, M. Nikolaidou, and G. Dimitrakopoulos. Real-time ECG monitoring using compressive sensing on a heterogeneous multicore edge-device. *Microprocessors and Microsystems*, 2019. 9, 21

[14] J. Connolly, J. Condell, B. O'Flynn, J. T. Sanchez, and P. Gardiner. IMU sensor-based electronic goniometric glove for clinical finger movement analysis. *IEEE Sensors Journal*, 18(3):1273–1281, 2017. 9, 17

[15] T. Tanaka, S. Guo, and N. Xiao. Development of a doctor's finger motion measurement device for a remote catheter operating system. In *2013 IEEE International Conference on Mechatronics and Automation*, pages 963–967. IEEE, 2013. 9, 10, 17

[16] M. S. Totty and E. Wade. Muscle activation and inertial motion data for noninvasive classification of activities of daily living. *IEEE Transactions on Biomedical Engineering*, 65(5):1069–1076, 2018. 10, 15

[17] M. Tavakoli, C. Benussi, P. A. Lopes, L. B. Osorio, and A. T de Almeida. Robust hand gesture recognition with a double channel surface EMG wearable armband and SVM classifier. *Biomedical Signal Processing and Control*, 46:121–130, 2018. 10, 15

[18] Q. Ju, R. Chalon, and S. Derrode. Assisted music score reading using fixed-gaze head movement: Empirical experiment and design implications. *Proceedings of the ACM on Human-Computer Interaction*, 3(EICS):3, 2019. 10

[19] M. Wang, C. Luo, B. Ni, J. Yuan, J. Wang, and S. Yan. First-person daily activity recognition with manipulated object proposals and non-linear feature fusion. *IEEE Transactions on Circuits and Systems for Video Technology*, 28(10):2946–2955, 2017. 10

[20] T. Aşuroğlu, K. Açıcı, Ç. B. Erdaş, M. Toprak, H. Erdem, and H. Oğul. Parkinson's disease monitoring from gait analysis via foot-worn sensors. *Biocybernetics and Biomedical Engineering*, 38(3):760–772, 2018. 11, 15

[21] N. Khoury, F. Attal, Y. Amirat, L. Oukhellou, and S. Mohammed. Data-driven based approach to aid Parkinson's disease diagnosis. *Sensors*, 19(2):242, 2019. 11, 21

[22] M. Liparoti, M. Della Corte, R. Rucco, P. Sorrentino, M. Sparaco, R. Capuano, R. Minino, L. Lavorgna, V. Agosti, G. Sorrentino, et al. Gait abnormalities in minimally disabled people with multiple sclerosis: A 3D-motion analysis study. *Multiple sclerosis and related disorders*, 2019. 12, 15

[23] S. Qiu, Z. Wang, H. Zhao, K. Qin, Z. Li, and H. Hu. Inertial/magnetic sensors based pedestrian dead reckoning by means of multi-sensor fusion. *Information Fusion*, 39:108–119, 2018. 12, 43

[24] H. Wang, Y. Yan, J. Hua, Y. Yang, X. Wang, X. Li, J. R. Deller, G. Zhang, and H. Bao. Pedestrian recognition in multi-camera networks using multilevel important salient feature and multicategory incremental learning. *Pattern Recognition*, 67:340–352, 2017. 12, 16

[25] Z. Ren, J. Meng, and J. Yuan. Depth camera based hand gesture recognition and its applications in human-computer-interaction. In *2011 8th International Conference on Information, Communications & Signal Processing*, pages 1–5. IEEE, 2011. 12

[26] M. L. Gavrilova, Y. Wang, F. Ahmed, and P. P. Paul. Kinect sensor gesture and activity recognition: new applications for consumer cognitive systems. *IEEE Consumer Electronics Magazine*, 7(1):88–94, 2017. 12

[27] A. Haria, A. Subramanian, N. Asokkumar, S. Poddar, and J. S. Nayak. Hand gesture recognition for Human computer interaction. *Procedia computer science*, 115:367–374, 2017. 12, 16

[28] N. Ç. Kılıboz and U. Güdükbay. A hand gesture recognition technique for human–computer interaction. *Journal of Visual Communication and Image Representation*, 28:97–104, 2015. 12, 16

[29] S. Shukri, L. M. Kamarudin, and M. H. F. Rahiman. Device-free localization for Human activity monitoring. In *Intelligent Video Surveillance*. IntechOpen, 2018. 13, 123

[30] L. Yao, Q. Z. Sheng, X. Li, T. Gu, M. Tan, X. Wang, S. Wang, and W. Ruan. Compressive representation for device-free activity recognition with passive RFID signal strength. *IEEE Transactions on Mobile Computing*, 17(2):293–306, 2017. 13, 17

[31] X. Huang and M. Dai. Indoor device-free activity recognition based on radio signal. *IEEE Transactions on Vehicular Technology*, 66(6):5316–5329, 2016. 13, 17

[32] L. Li, R. Bai, B. Xie, Y. Peng, A. Wang, W. Wang, B. Jiang, J. Liang, and X. Chen. R&P: An low-cost device-free activity recognition for E-health. *IEEE Access*, 6:81–90, 2017. 13, 17

[33] P. Torteeka, X. Chundi, and Y. Dongkai. Hybrid technique for indoor positioning system based on Wi-Fi received signal strength indication. In *2014 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 48–57. IEEE, 2014. 13, 20

[34] D. Plets, N. Podevijn, J. Trogh, L. Martens, and W. Joseph. Experimental performance evaluation of outdoor TDoA and RSS positioning in a public

LoRa network. In *2018 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–8. IEEE, 2018. 13, 20

[35] J. Luomala and I. Hakala. Analysis and evaluation of adaptive RSSI-based ranging in outdoor wireless sensor networks. *Ad Hoc Networks*, 87:100–112, 2019. 13, 20

[36] K. Moriya, E. Nakagawa, M. Fujimoto, H. Suwa, Y. Arakawa, A. Kimura, S. Miki, and K. Yasumoto. Daily living activity recognition with echonet lite appliances and motion sensors. In *2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, pages 437–442. IEEE, 2017. 14, 17

[37] K. N. Swaroop, K. Chandu, R. Gorrepotu, and S. Deb. A health monitoring system for vital signs using IoT. *Internet of Things*, 5:116–129, 2019. 14, 21

[38] H. Mshali, T. Lemlouma, M. Moloney, and D. Magoni. A survey on health monitoring systems for health smart homes. *International Journal of Industrial Ergonomics*, 66:26–56, 2018. 14, 21

[39] A. Moschetti, L. Fiorini, D. Esposito, P. Dario, and F. Cavallo. Daily activity recognition with inertial ring and bracelet: An unsupervised approach. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pages 3250–3255. IEEE, 2017. 15

[40] Y. Hsu, S. Yang, H. Chang, and H. Lai. Human daily and sport activity recognition using a wearable inertial sensor network. *IEEE Access*, 6:31715–31728, 2018. 16, 74

[41] F. S. Ayachi, H. P. Nguyen, E. G. de Brugiere, P. Boissy, and C. Duval. The use of empirical mode decomposition-based algorithm and inertial measurement units to auto-detect daily living activities of healthy adults. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 24(10):1060–1070, 2016. 16, 74

[42] Y. Dai, X. Wang, P. Zhang, and W. Zhang. Wearable biosensor network enabled multimodal daily-life emotion recognition employing reputation-driven imbalanced fuzzy classification. *Measurement*, 109:408–424, 2017. 16

[43] E. Garcia-Ceja, C. E. Galván-Tejada, and R. Brena. Multi-view stacking for activity recognition with sound and accelerometer data. *Information Fusion*, 40:45–56, 2018. 16

[44] C. Dhiman and D. K. Vishwakarma. A review of state-of-the-art techniques for abnormal Human activity recognition. *Engineering Applications of Artificial Intelligence*, 77:21–45, 2019. 16

[45] M. Wang, W. Y. Chen, and X. D. Li. Hand gesture recognition using valley circle feature and Hu's moments technique for robot movement control. *Measurement*, 94:734–744, 2016. 16

[46] D. Martinez and J. Singleton. Individual differences in lexical learning across two language modalities: Sign learning, word learning, and their relationship in hearing non-signers. *Acta Psychologica*, 198:102892, 2019. 16

[47] R. Elakkiya and K. Selvamani. Subunit sign modeling framework for continuous sign language recognition. *Computers & Electrical Engineering*, 74:379–390, 2019. 16

[48] P. Martins, H. Rodrigues, T. Rocha, M. Francisco, and L. Morgado. Accessible options for deaf people in e-learning platforms: technology solutions for sign language translation. *Procedia Computer Science*, 67:263–272, 2015. 16

[49] B. Dingenen, P. Malliaras, T. Janssen, L. Ceyssens, R. Vanelderen, and C. Barton. Two-dimensional video analysis can discriminate differences in running kinematics between recreational runners with and without running-related knee injury. *Physical Therapy in Sport*, 2019. 16

[50] C. Meng, T. Wang, W. Chou, S. Luan, Y. Zhang, and Z. Tian. Remote surgery case: robot-assisted teleneurosurgery. In *IEEE International Conference on*

Robotics and Automation, 2004. Proceedings. ICRA'04. 2004, volume 1, pages 819–823. IEEE, 2004. 17

[51] J. Suthakorn. A concept on cooperative tele-surgical system based on image-guiding and robotic technology. In *2012 Pan American Health Care Exchanges*, pages 41–45. IEEE, 2012. 17, 18

[52] J. E. G. Villarruel and B. T. Corona. Proposal for a remote surgery system based on wireless communications, electromyography and robotics. In *2008 Electronics, Robotics and Automotive Mechanics Conference (CERMA'08)*, pages 93–98. IEEE, 2008. 17

[53] O. Jansen, D. Grasmuecke, R. C. Meindl, M. Tegenthoff, P. Schwenkreis, M. Sczesny-Kaiser, M. Wessling, T. A. Schildhauer, C. Fisahn, and M. Aach. Hybrid assistive limb exoskeleton hal in the rehabilitation of chronic spinal cord injury: proof of concept; the results in 21 patients. *World neurosurgery*, 110:e73–e78, 2018. 18

[54] B. Chen, B. Zi, Z. Wang, L. Qin, and W. Liao. Knee exoskeletons for gait rehabilitation and Human performance augmentation: A state-of-the-art. *Mechanism and Machine Theory*, 134:499–511, 2019. 18

[55] M. A. Fikri, S. C. Abdullah, and M. H. M. Ramli. Arm exoskeleton for rehabilitation following stroke by learning algorithm prediction. *Procedia Computer Science*, 42:357–364, 2014. 18

[56] Q. Wu, X. Wang, B. Chen, and H. Wu. Development of an RBFN-based neural-fuzzy adaptive control strategy for an upper limb rehabilitation exoskeleton. *Mechatronics*, 53:85–94, 2018. 18

[57] W. Kim, S. Lee, H. Lee, S. Yu, J. Han, and C. Han. Development of the heavy load transferring task oriented exoskeleton adapted by lower extremity using quasi-active joints. In *2009 Iccas-Sice*, pages 1353–1358. IEEE, 2009. 18

[58] E. Foxlin. Pedestrian tracking with shoe-mounted inertial sensors. *IEEE Comput. graph. and app.*, 25(6):38–46, 2005. 19, 113

[59] A. Norrdine, Z. Kasmi, and J. Blankenbach. Step detection for ZUPT-aided inertial pedestrian navigation system using foot-mounted permanent magnet. *IEEE Sensors J.*, 16(17):6766–6773, 2016. 19

[60] H. Li, S. Derrode, L. Benyoussef, and W. Pieczynski. Free-walking 3D pedestrian large trajectory reconstruction from IMU sensors. In *26th European Signal Processing Conference (EUSIPCO)*, pages 657–661, Rome, Italy, September 2018. IEEE. 19, 31, 113, 119

[61] Q. Wang, J. Yin, A. Noureldin, and U. Iqbal. Research on an improved method for foot-mounted inertial/magnetometer pedestrian-positioning based on the adaptive gradient descent algorithm. *Sensors*, 18(12):4105, 2018. 19

[62] Y. Hsu, J. Wang, and C. Chang. A wearable inertial pedestrian navigation system with quaternion-based EKF for pedestrian localization. *IEEE Sensors J.*, 17(10):3193–3206, 2017. 19

[63] S. Beauregard, M. Klepal, et al. Indoor PDR performance enhancement using minimal map information and particle filters. In *2008 IEEE/ION Position, Location and Navigation Symposium*, pages 141–147. IEEE, 2008. 19

[64] Y. Wang, X. Li, and J. Zou. A foot-mounted inertial measurement unit (IMU) positioning algorithm based on magnetic constraint. *Sensors*, 18(3):741, 2018. 19

[65] S. Knauth and A. Koukofikis. Smartphone positioning in large environments by sensor data fusion, particle filter and FCWC. In *2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–5. IEEE, 2016. 20

[66] Z. Zhu, Y. Lu, C. You, and C. Chiang. Deep learning for sensor-based rehabilitation exercise recognition and evaluation. *Sensors*, 19(4), 2019. 20

[67] C. Y. Chiang, K. H. Chen, K. C. Liu, S Hsu, and C. T. Chan. Data collection and analysis using wearable sensors for monitoring knee range of motion after total knee arthroplasty. *Sensors*, 17(2):418, 2017. 20

[68] DS Andreasen, SK Alien, and DA Backus. Exoskeleton with EMG based active assistance for rehabilitation. In *9th International Conference on Rehabilitation Robotics, 2005. ICORR 2005.*, pages 333–336. IEEE, 2005. 21

[69] E. Akdogan, K. Shima, H. Kataoka, M. Hasegawa, A. Otsuka, and T. Tsuji. The cybernetic rehabilitation aid: Preliminary results for wrist and elbow motions in healthy subjects. *IEEE transactions on neural systems and rehabilitation engineering*, 20(5):697–707, 2012. 21

[70] E. Ambrosini, S. Ferrante, J. Zajc, M. Bulgheroni, W. Baccinelli, E. d'Amico, T. Schauer, C Wiesener, M Russold, M. Gfoehler, et al. The combined action of a passive exoskeleton and an EMG-controlled neuroprosthesis for upper limb stroke rehabilitation: First results of the retrainer project. In *2017 International Conference on Rehabilitation Robotics (ICORR)*, pages 56–61. IEEE, 2017. 21

[71] Y. Zhou, D. Vongsa, Y. Zhou, Z. Cheng, and L. Jing. A healthcare system for detection and analysis of daily activity based on wearable sensor and smartphone. In *2015 IEEE 12th Intl Conf on Ubiquitous Intelligence and Computing and 2015 IEEE 12th Intl Conf on Autonomic and Trusted Computing and 2015 IEEE 15th Intl Conf on Scalable Computing and Communications and Its Associated Workshops (UIC-ATC-ScalCom)*, pages 1109–1114. IEEE, 2015. 21

[72] J. K. Lee, S. N. Robinovitch, and E. J. Park. Inertial sensing-based pre-impact detection of falls involving near-fall scenarios. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 23(2):258–266, 2014. 21

[73] S. C. Agrawal, R. K. Tripathi, and A. S. Jalal. Human-fall detection from an indoor video surveillance. In *2017 8th International Conference on Com-*

puting, Communication and Networking Technologies (ICCCNT), pages 1–5. IEEE, 2017. 21

[74] A. Parri, K. Yuan, D. Marconi, T. Yan, S. Crea, M. Munih, R. M. Lova, N. Vitiello, and Q. Wang. Real-time hybrid locomotion mode recognition for lower limb wearable robots. *IEEE/ASME Transactions on Mechatronics*, 22(6):2480–2491, 2017. 23

[75] Z. Chen, Q. Zhu, Y. C. Soh, and L. Zhang. Robust Human activity recognition using smartphone sensors via CT-PCA and online SVM. *IEEE Transactions on Industrial Informatics*, 13(6):3070–3080, 2017. 23

[76] K. Safi, F. Attal, S. Mohammed, M. Khalil, and Y. Amirat. Physical activity recognition using inertial wearable sensors—a review of supervised classification algorithms. In *2015 International Conference on Advances in Biomedical Engineering (ICABME)*, pages 313–316. IEEE, 2015. 24

[77] J. Wannenburg and R. Malekian. Physical activity recognition from smartphone accelerometer data for user context awareness sensing. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 47(12):3142–3149, 2017. 24

[78] E. Fullerton, B. Heller, and M. Munoz-Organero. Recognising Human activity in free-living using multiple body-worn accelerometers. *IEEE Sensors Journal*, 17(16):5290–5297, 2017. 24, 40

[79] J. Wen and Z. Wang. Sensor-based adaptive activity recognition with dynamically available sensors. *Neurocomputing*, 218:307–317, 2016. 24, 77

[80] R. San-Segundo, J.-M. Montero, R. Barra-Chicote, F. Fernández, and J.-M. Pardo. Feature extraction from smartphone inertial signals for Human activity segmentation. *Signal Processing*, 120:359 – 372, 2016. 24, 74

[81] H. Zhao, Z. Wang, S. Qiu, J. Wang, F. Xu, Z. Wang, and Y. Shen. Adaptive gait detection based on foot-mounted inertial sensors and multi-sensor fusion. *Information Fusion*, 52:157–166, 2019. 24

[82] TLM Van Kasteren, G. Englebienne, and B. JA Kröse. Activity recognition using semi-Markov models on real world smart home datasets. *Journal of Ambient Intelligence and Smart Environments*, 2(3):311–325, 2010. 25

[83] D. Trabelsi, S. Mohammed, F. Chamroukhi, L. Oukhellou, and Y. Amirat. An unsupervised approach for automatic activity recognition based on hidden Markov model regression. *IEEE Transactions on automation science and engineering*, 10(3):829–835, 2013. 25

[84] M. A. Alsheikh, A. Selim, D. Niyato, L. Doyle, S. Lin, and H-P Tan. Deep activity recognition models with triaxial accelerometers. In *Workshops at the Thirtieth AAAI Conference on Artificial Intelligence*, 2016. 25

[85] P. Mamoshina, A. Vieira, E. Putin, and A. Zhavoronkov. Applications of deep learning in biomedicine. *Molecular Pharmaceutics*, 13(5):1445–1454, 2016. 25

[86] F. Ordóñez and D. Roggen. Deep convolutional and LSTM recurrent neural networks for multimodal wearable activity recognition. *Sensors*, 16(1):115, 2016. 25, 96

[87] Andrey Ignatov. Real-time Human activity recognition from accelerometer data using convolutional neural networks. *Applied Soft Computing*, 62:915 – 922, 2018. 25

[88] M. M. Hassan, M. Z. Uddin, A. Mohamed, and A. Almogren. A robust Human activity recognition system using smartphone sensors and deep learning. *Future Generation Computer Systems*, 81:307–313, 2018. 25

[89] S. Ramasamy Ramamurthy and N. Roy. Recent trends in machine learning for Human activity recognition—a survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 8(4):1254, 2018. 26

[90] Y. Bao and W. Chen. Automatic model construction for activity recognition using wearable devices. In *IEEE Int. Conf. on Pervasive Computing and Communications Workshops (PerCom Workshops)*, pages 806–811, 2018. 26, 95

[91] S. A. Rokni and H. Ghasemzadeh. Autonomous training of activity recognition algorithms in mobile sensors: a transfer learning approach in context-invariant views. *IEEE Transactions on Mobile Computing*, 17(8):1764–1777, 2018. 26, 95

[92] R. Ding, X. Li, L. Nie, J. Li, X. Si, D. Chu, G. Liu, and D. Zhan. Empirical study and improvement on deep transfer learning for Human activity recognition. *Sensors*, 19(1):57, 2019. 26

[93] T. Schneider, N. Helwig, and A. Schütze. Automatic feature extraction and selection for condition monitoring and related datasets. In *IEEE Int. Instrumentation and Measurement Technology Conf. (I2MTC)*, pages 1–6, 2018. 26

[94] H. Rezaie and M. Ghassemian. An adaptive algorithm to improve energy efficiency in wearable activity recognition systems. *IEEE Sensors Journal*, 17(16):5315–5323, 2017. 26

[95] M. S. Dao, T. A. Nguyen-Gia, and V. C. Mai. Daily human activities recognition using heterogeneous sensors from smartphones. *Procedia computer science*, 111:323–328, 2017. 26

[96] Y. Arakawa, K. Yasumoto, K. Pattamasiriwat, and T. Mizumoto. Improving recognition accuracy for activities of daily living by adding time and area related features. In *2017 Tenth International Conference on Mobile Computing and Ubiquitous Network (ICMU)*, pages 1–6. IEEE, 2017. 26

[97] C. Zhu and W. Sheng. Motion-and location-based online human daily activity recognition. *Pervasive and Mobile Computing*, 7(2):256–269, 2011. 26

[98] B. Barshan and M. C. Yüksek. Recognizing daily and sports activities in two open source machine learning environments using body-worn sensor units. *The Computer Journal*, 57(11):1649–1667, 2014. 29, 90, 91, 124

[99] D. Roggen, A. Calatroni, M. Rossi, T. Holleczek, K. Förster, G. Tröster, P. Lukowicz, D. Bannach, G. Pirkl, A. Ferscha, et al. Collecting complex

activity datasets in highly rich networked sensor environments. In *2010 Seventh international conference on networked sensing systems (INSS)*, pages 233–240. IEEE, 2010. 29, 95

[100] M. Zhang and A. S. Sawchuk. USC-HAD: A daily activity dataset for ubiquitous activity recognition using wearable sensors. In *ACM International Conference on Ubiquitous Computing (Ubicomp) Workshop on Situation, Activity and Goal Awareness (SAGAware)*, pages 1036–1043, Pittsburgh, Pennsylvania, USA, September 2012. ACM. 29

[101] M. Shoaib, S. Bosch, O. Incel, H. Scholten, and P. Havinga. Fusion of smartphone motion sensors for physical activity recognition. *Sensors*, 14(6):10146–10176, 2014. 29

[102] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz. A public domain dataset for Human activity recognition using smartphones. In *Esann*, 2013. 29

[103] D. Ravi, C. Wong, B. Lo, and G. Z. Yang. A deep learning approach to on-node sensor data analytics for mobile or wearable devices. *IEEE journal of biomedical and health informatics*, 21(1):56–64, 2016. 29

[104] J. R. Kwapisz, G. M. Weiss, and S. A. Moore. Activity recognition using cell phone accelerometers. *ACM SigKDD Explorations Newsletter*, 12(2):74–82, 2011. 30

[105] J. W. Lockhart, G. M. Weiss, J. C. Xue, S. T. Gallagher, A. B. Grosner, and T. T. Pulickal. Design considerations for the WISDM smart phone-based sensor mining architecture. In *Proceedings of the Fifth International Workshop on Knowledge Discovery from Sensor Data*, pages 25–33. ACM, 2011. 30

[106] H. Gjoreski, M. Ciliberto, L. Wang, F. J. O. Morales, S. Mekki, S. Valentin, and D. Roggen. The university of Sussex-Huawei locomotion and transportation dataset for multimodal analytics with mobile devices. *IEEE Access*, 6:42592–42604, 2018. 30

[107] A. Reiss and D. Stricker. Creating and benchmarking a new dataset for physical activity monitoring. In *Proceedings of the 5th International Conference on PErvasive Technologies Related to Assistive Environments*, page 40. ACM, 2012. 30

[108] P. Zappi, C. Lombriser, T. Stiefmeier, E. Farella, D. Roggen, L. Benini, and G. Tröster. Activity recognition from on-body sensors: accuracy-power trade-off by dynamic sensor selection. In *European Conference on Wireless Sensor Networks*, pages 17–33. Springer, 2008. 30

[109] T. T. Ngo, Y. Makihara, H. Nagahara, Y. Mukaigawa, and Y. Yagi. The largest inertial sensor-based gait database and performance evaluation of gait-based personal authentication. *Pattern Recognition*, 47(1):228–237, 2014. 30

[110] M. Bächlin, M. Plotnik, D. Roggen, I. Maidan, J. M. Hausdorff, N. Giladi, and G. Tröster. Wearable assistant for Parkinson's disease patients with the freezing of gait symptom. *IEEE Trans. Information Technology in Biomedicine*, 14(2):436–446, 2010. 30, 112

[111] J. Perry and J. R. Davids. Gait analysis: normal and pathological function. *Journal of Pediatric Orthopaedics*, 12(6):815, 1992. 31

[112] N. Shetty and S. Bendall. Understanding the gait cycle, as it relates to the foot. *Orthopaedics and Trauma*, 25(4):236 – 240, 2011. 31

[113] Tayfun Efe Ertop, Tolga Yuksel, and Erhan ilhan Konukseven. Realization of human gait in virtual fluid environment on a robotic gait trainer for therapeutic purposes. *Robotics and Autonomous Systems*, 105:59–68, 2018. 32

[114] N. M. Mueske, S. Õunpuu, D. D. Ryan, B. S. Healy, J. Thomson, P. Choi, and T. AL Wren. Impact of gait analysis on pathology identification and surgical recommendations in children with spina bifida. *Gait & posture*, 67:128–132, 2019. 32

[115] H. Zhao, Z. Wang, S. Qiu, Y. Shen, and J. Wang. IMU-based gait analysis for rehabilitation assessment of patients with gait disorders. In *2017 4th*

International Conference on Systems and Informatics (ICSAI), pages 622–626. IEEE, 2017. 32

[116] W. Pieczynski, C. Hulard, and T. Veit. Triplet Markov chains in hidden signal restoration. In *Image and Signal Processing for Remote Sensing VIII*, volume 4885, pages 58–69. Int. Society for Optics and Photonics, 2003. 49

[117] S. Derrode and W. Pieczynski. Signal and image segmentation using pairwise Markov chains. *IEEE Transactions on Signal Processing*, 52(9):2477–2489, Sept 2004. 49, 52

[118] I. Gorynin, H. Gangloff, E. Monfrini, and W. Pieczynski. Assessing the segmentation performance of pairwise and triplet Markov models. *Signal Processing*, 145:183 – 192, 2018. 49

[119] P. Lanchantin, J. Lapuyade-Lahorgue, and W. Pieczynski. Unsupervised segmentation of randomly switching data hidden with non-Gaussian correlated noise. *Signal Processing*, 91(2):163–175, 2011. 49

[120] J. P. Delmas. An equivalence of the EM and ICE algorithm for exponential family. *IEEE Transactions on Signal Processing*, 45(10):2613–2615, Oct 1997. 52

[121] J. Gorodkin. Comparing two K-category assignments by a K-category correlation coefficient. *Computational biology and chemistry*, 28(5-6):367–374, 2004. 60

[122] H. L. Bartlett and M. Goldfarb. A phase variable approach for IMU-based locomotion activity recognition. *IEEE Transactions on Biomedical Engineering*, 65(6):1330–1338, June 2018. 74, 77

[123] A. Jain and V. Kanhangad. Human activity classification in smartphones using accelerometer and gyroscope sensors. *IEEE Sensors Journal*, 18(3):1169–1177, 2018. 74

[124] F. Sikder and D. Sarkar. Log-sum distance measures and its application to Human-activity monitoring and recognition using data from motion sensors. *IEEE Sensors Journal*, 17(14):4520–4533, July 2017. 74

[125] S. Lin, Y. Lai, C. Hsia, P. Su, and C. Chang. Validation of energy expenditure prediction models using real-time shoe-based motion detectors. *IEEE Transactions on Biomedical Engineering*, 64(9):2152–2162, 2017. 74

[126] P. Li, Y. Wang, Y. Tian, T. Zhou, and J. Li. An automatic user-adapted physical activity classification method using smartphones. *IEEE Transactions on Biomedical Engineering*, 64(3):706–714, 2017. 74, 77

[127] M. A. Uddin, J. B. Joolee, A. Alam, and Y. Lee. Human action recognition using adaptive local motion descriptor in Spark. *IEEE Access*, 5:21157–21167, 2017. 77

[128] L. Cheng, Y. Yu, X. Liu, J. Su, and Y. Guan. Recognition of Human activities using fast and adaptive sparse representation based on wearable sensors. In *16th IEEE Int. Conf. on Machine Learning and Applications (ICMLA)*, pages 944–949, Dec 2017. 77

[129] Z. Sheng, C. Hailong, J. Chuan, and Z. Shaojun. An adaptive time window method for Human activity recognition. In *IEEE 28th Canadian Conf. on Electrical and Computer Engineering (CCECE)*, pages 1188–1192, May 2015. 77

[130] H. K. Hameed, W. Z. W. Hasan, S. Shafie, S. A. Ahmad, and H. Jaafar. An amplitude independent muscle activity detection algorithm based on adaptive zero crossing technique and mean instantaneous frequency of the sEMG signal. In *IEEE Regional Symp. on Micro and Nanoelectronics (RSM)*, pages 183–186, Aug 2017. 77

[131] V. S. Barbu and N. Limnios. *Semi-Markov chains and hidden semi-Markov models toward applications: their use in reliability and DNA analysis*, volume 191. Springer Science & Business Media, 2009. 83

[132] S. Z. Yu. *Hidden Semi-Markov models: theory, algorithms and applications*. Morgan Kaufmann, 2015. 83

[133] J. Lapuyade-Lahorgue and W. Pieczynski. Unsupervised segmentation of hidden semi-Markov non-stationary chains. *Signal Processing*, 92(1):29–42, 2012. 83

[134] O. Cappé. Online EM algorithm for hidden Markov models. *Journal of Computational and Graphical Statistics*, 20(3):728–749, 2011. 88, 89

[135] J. Yang, M. N. Nguyen, P. P. San, X. Li, and S. Krishnaswamy. Deep convolutional neural networks on multichannel time series for Human activity recognition. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015. 95

[136] C. A. Ronao and S. Cho. Human activity recognition with smartphone sensors using deep learning neural networks. *Expert systems with applications*, 59:235–244, 2016. 96

[137] H. S. Bhat and N. Kumar. On the derivation of the Bayesian information criterion. *School of Natural Sciences, University of California*, 2010. 99

[138] T. W. Arnold. Uninformative parameters and model selection using Akaike's information criterion. *The Journal of Wildlife Management*, 74(6):1175–1178, 2010. 99

[139] D. Trabelsi, S. Mohammed, Y. Amirat, and L. Oukhellou. Activity recognition using body mounted sensors: An unsupervised learning based approach. In *The 2012 International Joint Conf. on Neural Networks (IJCNN)*, pages 1–7. IEEE, 2012. 112

[140] Zhenhui. Li, B. Ding, J. Han, R. Kays, and P. Nye. Mining periodic behaviors for moving objects. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1099–1108. ACM, 2010. 112

[141] Z. He, X. S. Wang, B. S. Lee, and A. C. Ling. Mining partial periodic correlations in time series. *Knowledge and Information systems*, 15(1):31–54, 2008. 112

[142] Y. Nam and J. W. Park. Child activity recognition based on cooperative fusion model of a triaxial accelerometer and a barometric pressure sensor. *IEEE journal of biomedical and health informatics*, 17(2):420–426, 2013. 112

[143] H. Li, S. Derrode, and W. Pieczynski. Lower limb locomotion activity recognition using semi-Markov model and single wearable inertial sensor. *Sensors*, 19:4242, 2019. 119

[144] H. Li, S. Derrode, L. Benyoussef, and W. Pieczynski. Unsupervised pedestrian trajectory reconstruction from IMU sensors. Hammamet, Tunisia, Apr 2018. 119