RECOGNITION AND CLASSIFICATION OF HUMAN ACTIVITIES USING WEARABLE SENSORS

A THESIS

SUBMITTED TO THE DEPARTMENT OF ELECTRICAL AND

ELECTRONICS ENGINEERING

AND THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE

OF BILKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF SCIENCE

By

Aras Yurtman

September 2012

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Prof. Dr. Billur Barshan (Advisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assoc. Prof. Dr. Nail Akar

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assist. Prof. Dr. Behçet Uğur Töreyin

Approved for the Graduate School of Engineering and Science:

Prof. Dr. Levent Onural
Director of the Graduate School

ABSTRACT

RECOGNITION AND CLASSIFICATION OF HUMAN ACTIVITIES USING WEARABLE SENSORS

Aras Yurtman
M.S. in Electrical and Electronics Engineering
Supervisor: Prof. Dr. Billur Barshan
September 2012

We address the problem of detecting and classifying human activities using two different types of wearable sensors. In the first part of the thesis, a comparative study on the different techniques of classifying human activities using tag-based radio-frequency (RF) localization is provided. Position data of multiple RF tags worn on the human body are acquired asynchronously and non-uniformly. Curves fitted to the data are re-sampled uniformly and then segmented. The effect of varying the relevant system parameters on the system accuracy is investigated. Various curve-fitting, segmentation, and classification techniques are compared and the combination resulting in the best performance is presented. The classifiers are validated through the use of two different cross-validation methods. For the complete classification problem with 11 classes, the proposed system demonstrates an average classification error of 8.67% and 21.30% for 5-fold and subject-based leave-one-out (L1O) cross validation, respectively. When the number of classes is reduced to five by omitting the transition classes, these errors become 1.12% and 6.52%. The system demonstrates acceptable classification performance despite that tag-based RF localization does not provide very accurate position measurements.

In the second part, data acquired from five sensory units worn on the human body, each containing a tri-axial accelerometer, a gyroscope, and a magnetometer, during 19 different human activities are used to calculate inter-subject and inter-activity variations in the data with different methods. Absolute, Euclidean, and dynamic time-warping (DTW) distances are used to assess the similarity of the signals. The comparisons are made using time-domain data and feature vectors. Different normalization methods are used and compared. The "best" subject is defined and identified according to his/her average distance to the other subjects.

Based on one of the similarity criteria proposed here, an autonomous system that detects and evaluates physical therapy exercises using inertial sensors and magnetometers is developed. An algorithm that detects all the occurrences of one or more template signals (exercise movements) in a long signal (physical therapy session) while allowing some distortion is proposed based on DTW. The algorithm classifies the executions in one of the exercises and evaluates them as correct/incorrect, identifying the error type if there is any. To evaluate the performance of the algorithm in physical therapy, a dataset consisting of one template execution and ten test executions of each of the three execution types of eight exercise movements performed by five subjects is recorded, having totally 120 and 1,200 exercise executions in the training and test sets, respectively, as well as many idle time intervals in the test signals. The proposed algorithm detects 1,125 executions in the whole test set. 8.58% of the executions are missed and 4.91% of the idle intervals are incorrectly detected as an execution. The accuracy is 93.46% for exercise classification and 88.65% for both exercise and execution type classification. The proposed system may be used to both estimate the intensity of the physical therapy session and evaluate the executions to provide feedback to the patient and the specialist.

Keywords: radio-frequency localization, radio-frequency identification, human activity recognition, pattern recognition, classification, feature extraction, feature reduction, principal components analysis, linear discriminant analysis, P-fold cross-validation, leave-one-out cross-validation, absolute distance, Euclidean distance, dynamic time warping, subsequence dynamic time warping, dynamic programming, normalization, inertial sensors, accelerometers, gyroscopes, magnetometers, pattern search, movement detection, physical therapy.

ÖZET

GİYİLEBİLİR DUYUCULARLA İNSAN AKTİVİTELERİNİN ALGILANMASI VE SINIFLANDIRILMASI

Aras Yurtman
Elektrik ve Elektronik Mühendisliği, Yüksek Lisans
Tez Yöneticisi: Prof. Dr. Billur Barshan
Eylül 2012

Farklı türde giyilebilir algılayıcılar kullanarak insan aktivitelerinin sezimi ve sınıflandırılması ele alınmaktadır. Tezin ilk bölümünde, etiket tabanlı, radyo frekansına dayalı bir konumlama sistemi ile insan aktivitesi tanımada çeşitli yöntemlerin kullanımı karşılaştırmalı olarak sunulmuştur. İnsan bedeninin farklı bölgelerine yerleştirilen etiketlerin konumları, eşzamansız ve farklı aralıklarla örneklenmiş olarak elde edilmektedir. Bu verilere uyarlanan eğriler, düzgün örneklenmiş ve bölütlenmiştir. İlgili sistem parametrelerinin sistem başarımına etkisi incelenmiştir. Çeşitli eğri uyarlama, bölütleme ve sınıflandırma yöntemleri karşılaştırılmış ve en iyi başarımı veren katışım sunulmuştur. Sınıflandırıcılar, iki farklı bağımsız geçerlilik sınaması yöntemiyle değerlendirilmiştir. 11 sınıftan oluşan sınıflandırma probleminde, sırasıyla P-bölmeli ve birini dışarıda bırak bağımsız geçerlilik sınamaları kullanıldığında ortalama sınıflandırma hataları %8.67 ve %21.30 olarak elde edilmiştir. Geçiş sınıfları dışarıda bırakılarak elde edilen beş sınıflı sınıflandırma probleminde ise, bu hatalar %1.12 ve %6.52'ye düşmektedir. Etiket-tabanlı konumlama sistemlerinin çok hassas konum ölçümleri sağlamamasına karşın sonuçlar, sistemin kabul edilebilir bir sınıflandırma başarımı sunduğunu göstermektedir.

İkinci bölümde, insan bedeninin beş noktasına yerleştirilmiş, her biri üç eksenli ivmeölçer, dönüölçer ve manyetometre içeren beş duyucu ünitesinden 19 farklı günlük aktivite sırasında elde edilen veriler, katılımcılar arası ve aktiviteler arası farklılıkları çeşitli yöntemlerle hesaplamak için kullanılmıştır. İşaretlerin karşılaştırılması için mutlak, Öklit ve dinamik zaman bükmesi (DZB) uzaklıkları kullanılmıştır. Karşılaştırmalar, zaman bölgesindeki veri ve öznitelik vektörleri kullanılarak yapılmıştır. Farklı düzgeleme yöntemleri kullanılmış ve karşılaştırılmıştır. "En iyi" katılımcı, diğer katılımcılara olan ortalama uzaklığa dayalı olarak tanımlanmış ve saptanmıştır. Bu kısımda önerilen benzerlik ölçütlerinden biri seçilerek, eylemsizlik duyucuları ve manyetometreler kullanılarak fizik tedavi egzersizlerini sezen ve değerlendiren özerk bir sistem geliştirilmiştir. DZB yöntemine dayanarak belirli ölçüde bozuluma izin vererek, uzun bir işaretin (fizik tedavi seansı) içinde bir ya da birden fazla şablon işaretin (fizik tedavi hareketkeri) bütün olagelişlerini sezen bir algoritma öne sürülmüştür. Bu algoritma, yürütümleri egzersiz hareketlerinden birisi olarak sınıflandırmakta, doğru/yanlış olarak değerlendirmekte ve eğer varsa yapılan hatanın türünü belirtmektedir. Algoritmanın fizik tedavideki başarımını belirlemek için, beş katılımcı tarafından yürütülen sekiz egzersiz hareketinin üç farklı yapılış biçiminin her biri için bir şablon ve on test yürütümünden oluşan ve böylece eğitim ve test veri kümelerinde sırasıyla 120 ve 1,200 egzersiz yürütümü ile test işaretlerinde birçok boş zaman aralığı içeren bir veri kümesi kaydedilmiştir. Öne sürülen algoritma, bütün test kümesinde 1,125 yürütüm sezmiştir. Yürütümlerin %8.58'i sezilememiş, boş aralıkların %4.91'i yanlışlıkla yürütüm olarak sezilmiştir. Başarım, egzersiz ayırt etmede %93.46, hem egzersiz hem de yapılış şekli ayırt etmede %88.65'tir. Geliştirilen sistem, hem fizik tedavi seansının yoğunluğunu kestirmek için, hem de egzersiz yürütümlerini değerlendirerek hastaya ve uzmana geribildirim vermek için kullanılabilir.

Anahtar sözcükler: radyo-frekanslı konumlama, radyo-frekanslı tanımlama, insan aktivitesi tanıma, örüntü tanıma, sınıflandırma, öznitelik çıkarma, öznitelik indirgeme, ana bileşenler çözümlemesi, doğrusal ayırtaç çözümlemesi, P-bölmeli bağımsız geçerlilik sınaması, birini dışarıda bırak bağımsız geçerlilik sınaması, mutlak uzaklık, Öklit uzaklığı, dinamik zaman bükmesi, altdizi dinamik zaman bükmesi, dinamik programlama, düzgeleme, eylemsizlik duyucuları, manyetometreler, örüntü arama, hareket algılama, fizik tedavi.

Acknowledgement

I would like to express my gratitude to my supervisor Prof. Dr. Billur Barshan for her support in the development of this thesis.

I would like to thank Assoc. Prof. Dr. İlknur Tuğcu for sharing the literature and invaluable assistance in the field of physical therapy.

I would also like to thank Assoc. Prof. Dr. Nail Akar and Assist. Prof. Dr. Behçet Uğur Töreyin for accepting to read and review this thesis.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

With rapidly developing technology, devices such as personal digital assistants (PDAs), smart phones, and tablet computers have made their way to our daily lives. It is now becoming essential for such devices to recognize and interpret human behavior correctly in real time. One aspect of this type of context-aware systems is the recognition and monitoring of activities of daily living such as sitting, standing, lying down, walking, ascending/descending stairs, and most importantly, falling.

1.1 Approaches in Activity Recognition

There exist several different approaches for the recognition of human activities in the literature [1]. The most common three approaches are based on computer vision, radio-frequency localization systems, and inertial sensors. Earlier work in this area is summarized below.

1.1.1 Activity Recognition Using Visual Sensors

A large number of studies employ vision-based systems with multiple video cameras mounted in the environment to recognize the activities performed by a person [2–5]. In many of these studies, points of interest on the human body are pre-identified by placing special, visible markers such as light-emitting diodes (LEDs) at those points and the positions of the markers are recorded by cameras [6]. For example, Kaluža et al. [7] considered a total of six activities including falls using the Smart infrared motion capture system. The attributes they used are the coordinates of the body parts in different coordinate systems and the angles between adjacent body parts. In Luštrek et al. [8], walking anomalies, such as limping, dizziness, and hemiplegia are detected using the same system. Camera systems obviously interfere with privacy since they capture additional information that is not needed by the system but that may easily be exploited with a simple modification on the software. Hence, people act unnaturally and feel uncomfortable when camera systems are used, especially in private places. Other disadvantages of vision-based systems are the high computational cost of processing images and videos, correspondence and shadowing problems, the need for camera calibration, and inoperability in the dark. When multiple cameras are employed, several 2-D projections of the 3-D scene have to be combined. Moreover, this approach imposes restrictions on the mobility of the person since the system operates only in the limited environment monitored by the cameras.

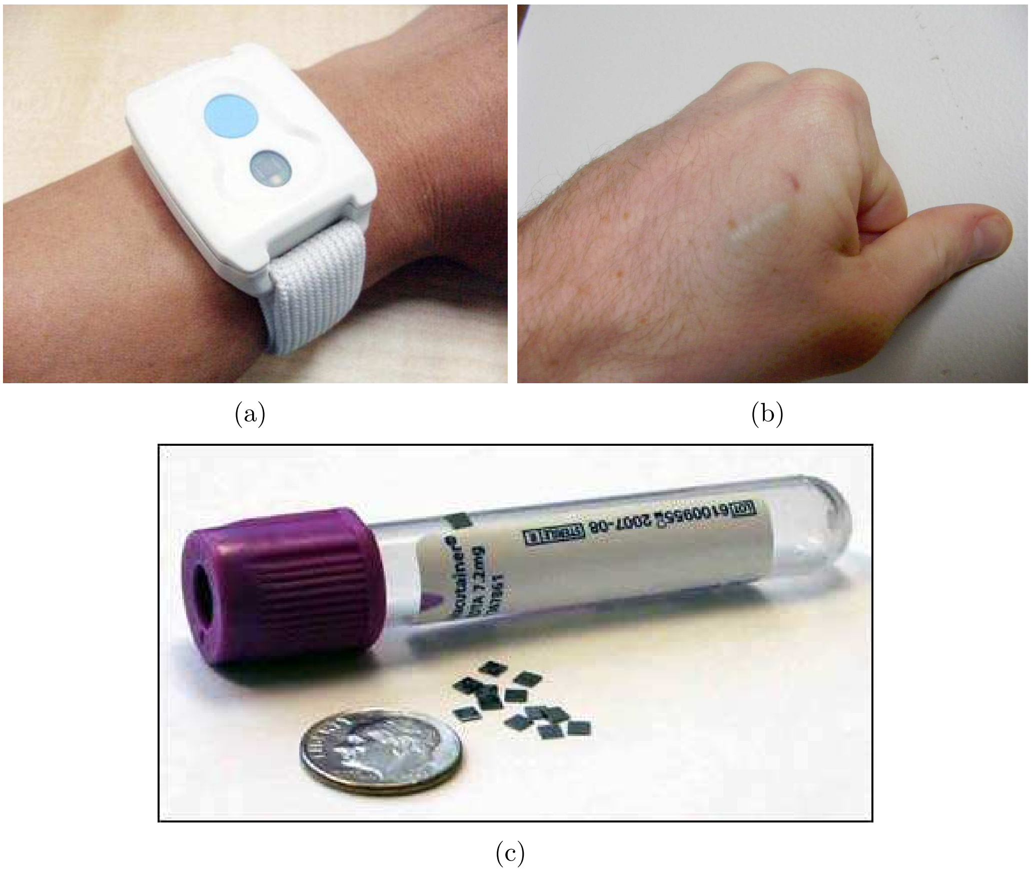
1.1.2 Activity Recognition Using Radio-Frequency Localization

Rather than monitoring human activities from a distance or remotely, we believe that “activity can best be measured where it occurs” as stated in [9]. Unlike the first approach utilizing visual sensors, the use of inertial sensors and the radio-frequency (RF) localization-based approach directly acquire the motion data and position data in 3-D, respectively. In the RF localization technique, the 3-D positions of the RF tags worn on different parts of the body are estimated[1]. Multiple antennas called *readers* are mounted in the environment that detect the relative positions of small devices called RF tags (Figure 1.1). Each tag emits RF pulses containing its unique ID for identification and localization. *Active* RF tags have internal power sources (batteries) to transmit RF pulses, whereas *passive* tags do not contain a power source and rely on the energy of the waves transmitted by the readers [11]. Passive RF tags are small stickers similar to RFID tags that can be as small as 2 mm × 2 mm in size (Figure 1.1(c)), whereas active RF tags are much larger than the passive ones (Figure 1.1(a)). RF tags are very inexpensive and lightweight, and thus comfortable to use on the human body [12]. Unlike bar codes, the tag does not need to be within the line of sight of the reader and may be embedded in the tracked object or even buried under the skin (Figure 1.1(b)).

The operating range of most RF readers is not more than 10 m [13]. In uncluttered, open environments, typical localization accuracy is about 15 cm across 95% of the readings [14]. Since each tag must be detected by multiple readers for localization, this method cannot be used in large areas because in that case, numerous readers are needed, which would be too costly. On the other hand, the number of RF tags that can be worn on the body is limited. In systems that use active tags, the pulses transmitted by the tags may interfere, whereas in systems that use passive tags, it becomes difficult for the system to distinguish the tags that are close together.

RFID technology is a very valuable tool in a variety of applications involving the automatic detection, identification, localization, and tracking of persons, animals, vehicles, baggage, and goods [15]. RFID systems are used for general

[1] *Radio-Frequency Identification (RFID)* is a technique involving the *detection and identification* of tags (deciding which tags exist in the environment), whereas in *RF localization*, the tags are both *identified* and localized. The tags are called "RFID tags" in the former system and "RF tags" in the latter although they can be identical in some cases [10,11]. In this thesis, the tags used for localization are not the same as RFID tags, hence will be called "RF tags." However, in some texts, the term "RFID localization" is used instead of RF localization because there are systems estimating the positions of RFID tags that are designed for identification only [10].



(a)

(b)

(c)

Figure 1.1: Examples of RFID tags. (a) an active RFID tag worn as a bracelet (Syris sytag245-tm, reprinted from `http://blog.aztronics.com/?p=45`), (b) an RFID tag buried under the skin (An RFID Body Mod Most Curious, Ization Labs, reprinted from `http://izationlabs.com/2009/12/22/a-body-mod-most-curious/`), (c) tiny RFID tags of size 2 mm × 2 mm (Chip-size Passive RFID Tag, Health Care News, RFID Journal, reprinted from `http://www.rfidjournal.com/imagecatalogue/imageview/5866/?RefererURL=/article/view/4585`).

transport logistics, toll collection from vehicles and contactless payment, tracking of parcels and baggage, and to avoid theft of the items sold in stores or supermarkets. With the use of RFID tags, assembly lines and inventories in the supply chain can be tracked more efficiently and products become harder to falsify. This is particularly important for the pharmaceutical industry with increasing anti-counterfeit measures. In the identification and tracking of animals, RFID tags are used for tracking pets, farm animals, and rare animal species such as pandas. They are used on contactless identity cards for access management and control of hospitals, libraries, museums, schools and universities, and restricted zones. Machine readable identification and travel documents such as biometric passports that contain RFID tags are becoming very common. RFID tags are also used for key identification in vehicles, for locking/unlocking vehicles from a distance, ticketing of mass events or public transport, and transponder timing of sporting events.

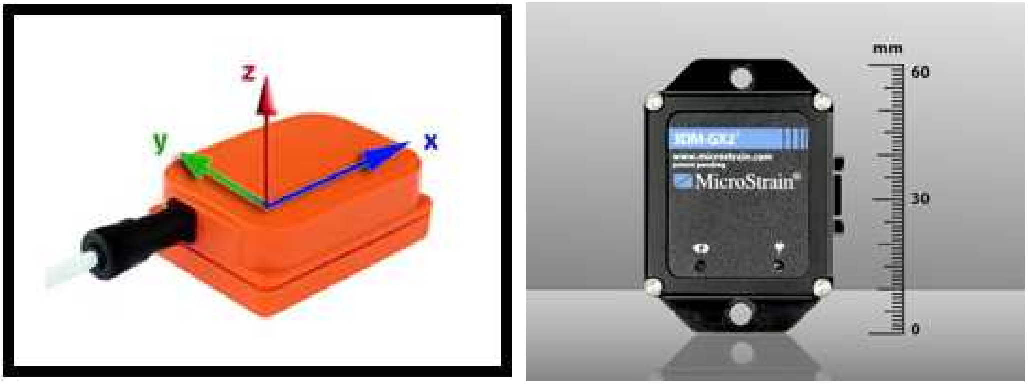
Besides the above uses, RFID systems are also suitable for indoor localization and mapping [16]. Reference [17] analyzes whether the use of RFID technology in the field of robotics can improve the localization of mobile robots and persons in their environment and determines the computational requirements. In [18], many RFID tags are placed on the floor in a grid configuration at known positions and a robot localizes itself by detecting the tags using its antenna. To resolve the issues concerning security and privacy of RFID systems, reference [19] proposes an authentication protocol based on RFID tags.

There are many studies involving activity recognition using RFID; however, they are *not* based on RF localization. In the earlier work on human activity recognition using RFID technology, activities of daily living are mostly inferred based on the interactions of a person with the objects in its environment. RFID antennas are worn on the body usually in the form of gloves or bracelets, and RFID tags are fixed to the objects in the environment such as equipment, tools, furniture, or doors (or vice versa). Then, the position of the subject and the activity s/he is performing is estimated from the tag readings in consequence of body-object interactions. The main limitation of these systems is that they provide activity information only when the subject interacts with one of the tagged objects in its environment. Thus, the only recognizable activities are those that involve these objects. This type of approach is followed in [20–24]. Similarly, in [25], the authors employ RFID sensor networks based on wireless identification and sensing platforms (WISPs) that combine passive UHF RFID technology with traditional sensors. Everyday objects are tagged with WISPs to detect when they are in use, and a simple hidden Markov model (HMM) is used to convert object traces into high-level daily activities. In [26], a dynamic Bayesian network model is presented that combines RFID and video data to jointly infer the most likely household activity and object labels. Reference [27] combines data from RFID tag readers and accelerometers to recognize ten housekeeping activities with higher accuracy. A multi-agent system for fall and disability detection of the elderly living alone is presented in [28], based on the commercially available Ubisense smart space platform [14]. On the other hand, in (Chapter 2 of) this thesis, human activities are recognized by using an RF localization system, where the 3-D position estimations of the RF tags are used. This technique does not require any object interactions. Furthermore, a new approach based on curve-fitting is applied to the non-uniform and asynchronous position measurements of the RF tags to segment them and extract their features. This approach also solves the problem of missing data that is encountered in RF localization systems.

1.1.3 Activity Recognition Using Inertial Sensors

The third approach utilizes miniature inertial sensors whose size, weight, and cost have decreased considerably during the last two decades. The availability of lower cost, medium performance inertial sensor units has opened up new possibilities for their use. In this approach, several sensor units are worn on different parts of the body. These units usually contain gyroscopes and accelerometers, and sometimes, magnetometers in addition. Some of these devices are sensitive around a single axis whereas others are multi-axial (usually two- or three-axial). Two examples are shown in Figure 1.2 and a wearable system is illustrated in Figure 1.3.

Inertial sensors do not directly provide linear or angular position information. Gyroscopes provide *angular rate* information around an axis of sensitivity,



(a)

(b)

Figure 1.2: (a) Xsens MTx [29] and (b) 3DM-GX2 [30] sensor units.

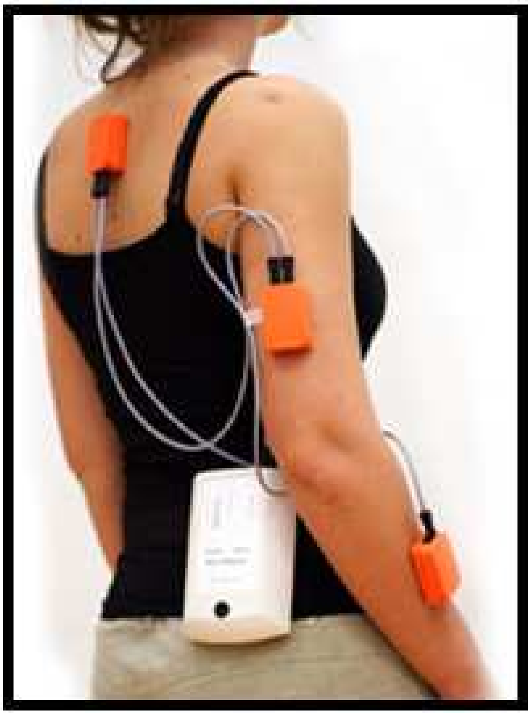


Figure 1.3: Miniature inertial sensor units worn on the body (reprinted from `http://www.xsens.com/en/movement-science/xbus-kit`).

whereas accelerometers provide *linear or angular velocity rate* information. These rate outputs need to be integrated once or twice to obtain the linear/angular position. Thus, even very small errors in the rate information provided by inertial sensors cause an unbounded growth in the error of integrated measurements.

The acquired measurements are either collected and processed in a battery-powered system such as a cellular phone, or wirelessly transmitted to a computer to be processed. Detailed literature surveys on activity recognition using inertial sensors can be found in [31–35]. In the earlier work on human activity recognition, the utilization of inertial sensors [35–38] is also considered. Although this method results in accurate classification, wearing the sensors and the processing unit on the body may not always be comfortable or even acceptable despite how small and lightweight they have become. People may forget or neglect to wear these devices. Furthermore, this approach has certain limitations: Although it is demonstrated that it is possible to recognize activities with high accuracy (typically above 95%), the same is not true for human localization because of the drift problem associated with inertial sensors [39,40]. In [39], it is considered to exploit activity cues to improve the erroneous position estimates provided by inertial sensors and have achieved significantly better accuracies in localization when performed simultaneously with activity recognition.

In many studies on activity recognition with wearable inertial sensors, it is observed that the classification accuracy decreases significantly when the activities of a subject are classified using the classifiers trained with the data of other subjects [35]. However, the reason is not investigated so far. For this purpose, using the previous activity recognition dataset [35], inter-subject and intra-subject variations in the data are investigated in detail in Chapter 3. The signals are normalized with various methods. To compare signals, different distance measures are used. Based on the results, the subject that performs the activities in the best way is also identified.

1.2 Application of Activity Recognition in Physical Therapy

A different aspect of activity recognition may be very useful in detecting and evaluating exercise movements in the physical therapy field. The patients often perform the prescribed exercise movements in a hospital or a rehabilitation center for a while and they continue exercising at home, where they receive no feedback about how correctly they perform [41]. In addition, while exercising under the supervision of a specialist, it is common that the patients receive poor feedback due to the number of personnel being insufficient, the difficulty of tracking several patients at the same time, and/or subjective feedback due to negligence of the specialists and the lack of systematic rules for many exercises [42, 43]. For this purpose, it would be very useful and valuable if the individual exercise movements can be evaluated automatically by an intelligent system utilizing wearable sensors, as done in Chapter 4 in this thesis. For example, wearable miniature inertial sensors or an RF localization system with RF tags placed on the body can be used, both of which are much less expensive and highly portable compared to medical devices such as biofeedback devices [44]. In addition, both systems are easier to be placed on the body by the patient himself compared to tight garments containing strain sensors, which are used, for example, in reference [45].

In (Chapter 4 of) this thesis, we use one of the similarity criteria proposed in Chapter 3 to detect and evaluate the executions of physical therapy exercises automatically using wearable sensing units. An algorithm that detects all the occurrences of one or more template signals (exercise movements) in a long signal (physical therapy session) while allowing some distortion is proposed based on DTW. The algorithm classifies the executions in one of the exercises and evaluates them as correct/incorrect, identifying the error type if there is any. The developed system also estimates the number of (correct or all) executions, providing feedback about not only the correctness of the executions, but also the effectiveness or intensity of physical therapy sessions.

The organization of this thesis is as follows: In Chapter 2, human activities are classified using an RF localization system and various classification methods. Variations in inertial sensor data of human activities performed by different subjects are examined in Chapter 3 and three similarity criteria are proposed. In Chapter 4, a system that detects and evaluates physical therapy exercises by using a novel algorithm applied to inertial sensor and magnetometer data is presented based on one of the similarity criteria proposed in Chapter 3. Finally, in Chapter 5, conclusions are drawn and directions for future work are provided.

Chapter 2

Human Activity Recognition Using Tag-Based Radio-Frequency Localization

In this chapter, human activities are classified using an RF localization system. The work presented in this chapter is an extension of the study in reference [46].

An important issue in most RF systems is that the system measures the tag positions asynchronously and non-uniformly at different, arbitrary time instants. In other words, whenever the readers receive a pulse transmitted by an RF tag, the system records its relative position along the x, y, z axes as well as a unique timestamp and its unique ID. Although each tag transmits pulses periodically, tags cannot be synchronized since their pulses must not interfere with each other and thus the locations of the tags are sampled at different time instants. Furthermore, the readers sometimes cannot detect the pulses due to low signal-to-noise ratio (SNR), interference, or occlusion. Under these circumstances, localization accuracy may drop significantly. The detection ratio of a tag increases when it is close to the antennas and decreases when it is near conductive objects. Thus, it is not possible to treat the raw measurements as ordinary position vectors sampled at a constant rate in time.

In this study, we consider a broad set of daily activity types (11 activities) and recognize these activities with high accuracy without having to take into account the interaction of a person with the objects in its environment. We only keep track of the position data of four RF tags worn on different parts of the body, acquired by the Ubisense platform [14]. In the data pre-processing stage, we propose a method to put the dataset in uniformly and synchronously sampled form. After feature reduction in two different ways, we compare several classifiers through the use of P-fold and subject-based leave-one-out (L1O) cross-validation techniques. The variation of the relevant system parameters on the classification performance is investigated.

In Section 2.1, details of the experimental setup and the dataset are provided. Section 2.2 describes pre-processing of the dataset. Feature extraction and reduction is the topic of Section 2.3. Classifiers used for activity recognition are listed in Section 2.4. Section 2.5 describes the performance evaluation of the classifiers through the use of two cross-validation techniques. In Section 2.6, experimental results are presented and interpreted. Lastly, conclusions are drawn in Section 2.7.

2.1 The System Details

The human activity recognition system employed in this study employs four active RF tags worn on different parts of the body, whose relative positions along the three axes are detected by a computer or a simple microcontroller via four RF antennas mounted in the environment (see Figure 2.1) [47]. The four RF tags are positioned on the left ankle (tag 1), right ankle (tag 2), chest (tag 3) and the belt (tag 4).

The 3-D position data of the four RF tags worn by a subject are measured over time while s/he is performing a fixed sequence of predetermined activities. The operating range of the system is about 46 m. Although each tag transmits a pulse every 0.1 s, the readers may miss some of the pulses (due to occlusion, low SNR at large distances, etc.) and therefore, the data acquisition rate is not

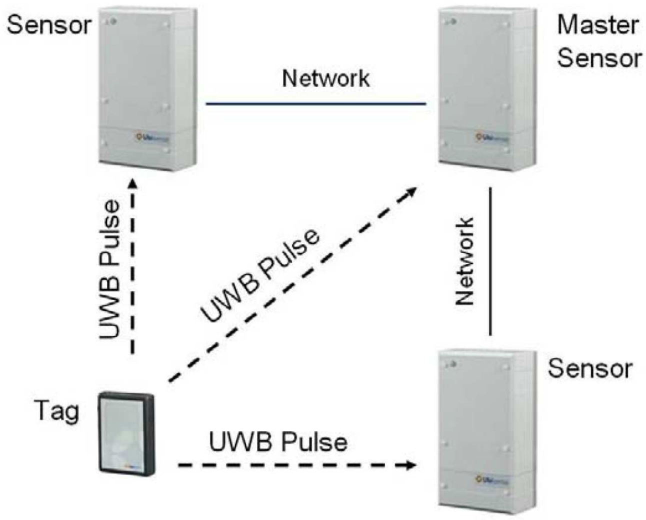


Figure 2.1: Ubisense hardware components [14].

constant. However, the average detection rate does not vary too much and is about 9 Hz most of the time.

The 11 activity types are numbered as follows: (1): walking, (2): falling, (3): lying down, (4): lying, (5): sitting down, (6): sitting, (7): standing up from lying, (8): on all fours, (9): sitting on the ground, (10): standing up from sitting, (11): standing up from sitting on the ground.

Each subject performs a sequence of activities referred as an "experiment" in this chapter. Each experiment consists of the following sequence of activities with different but similar durations:

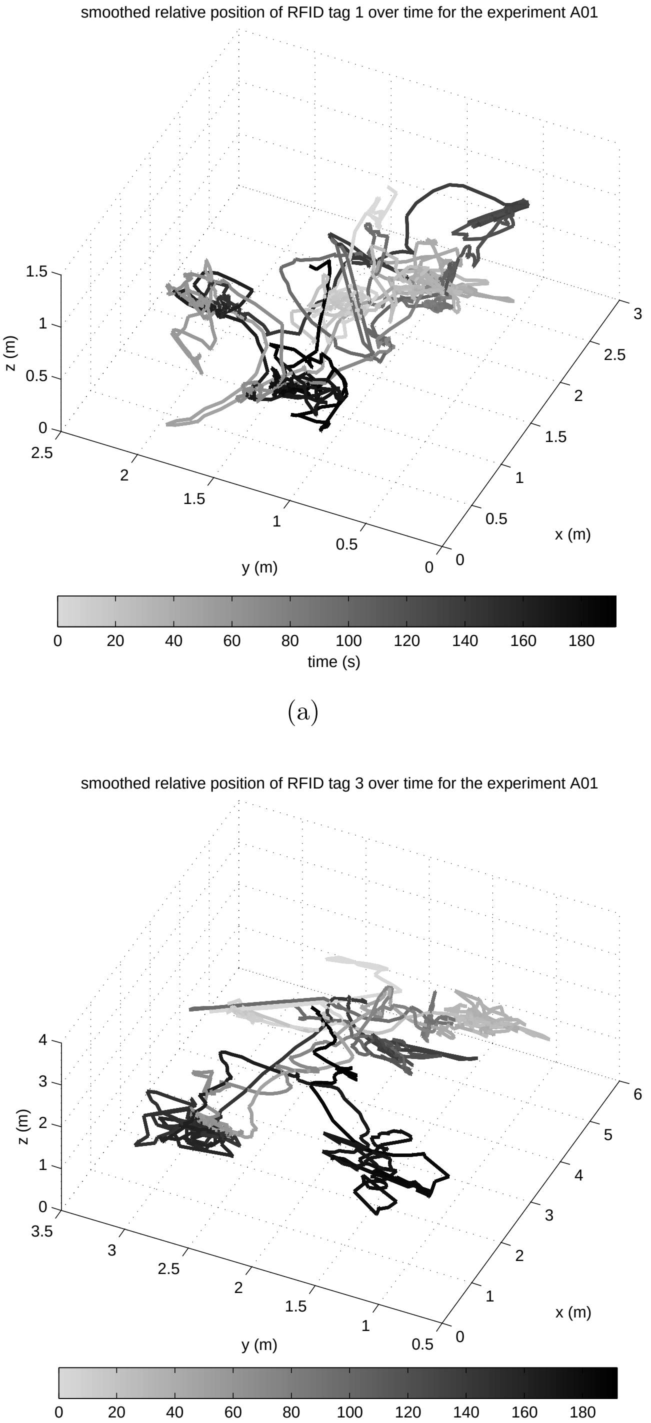
walking—sitting down—sitting—standing up from sitting—
walking—falling—lying—standing up from lying—walking—
lying down—lying—standing up from lying—walking—falling—
lying—standing up from lying—walking—sitting down—sitting—
sitting on the ground—standing up from sitting on the ground—
walking—lying down—lying—standing up from lying—walking—
lying down—on all fours—lying—standing up from lying—walking

There are five subjects, each performing the same experiment five times. Thus, there is a total of $5 \times 5 = 25$ experiments of the same scenario. The dataset described above and used in this study is entitled "Localization Data for Person Activity Data Set" and is publicly available at the University of California, Irvine Machine Learning Repository [47]. The dataset is an extremely long but simple-structured 2-D array of size $164,860 \times 8$ (see Figure 2.2 for sample rows). Each line of the raw data corresponds to one measurement, where the first element denotes the subject code (A-F) and the experiment number (01-05), the second element is the tag ID (the unique ID of one of the four tags), the third column is a unique timestamp, the fourth column is the explicit date and time, the 5th, 6th and the 7th columns respectively contain the relative x, y, z position of the tag, and the 8th column stores the event name, corresponding to one of the 11 activities performed. In the modified dataset, each activity type is represented by its number for simplicity without loss of information. Similarly, the unique IDs of the tags in the raw data are converted to tag numbers 1–4 for the sake of simplicity and without loss of information. Note that, a measurement, corresponding to one of the rows of the dataset, simply defines the relative position of a particular tag at a particular time instant (as well as the true activity type) and is acquired by multiple antennas. The data of each experiment are just a subset of the rows in the raw data array. Therefore, the sequence of activities and their durations can be extracted from the dataset. As an example, the positions of tags 1 and 3 in the first experiment of the first subject are shown in Figure 2.3 as 3-D curves in time. In the figure, the gray level of the curve changes from light gray to black as time passes.

An important problem in activity recognition is the detection of the activity durations and transition times in a continuous data stream [48–51]. In the dataset, activities 2, 3, 5, 7, 10, and 11 actually correspond to *transitions* between two activities and their duration is much shorter than the others. For example, class 5 called "sitting down" stands for the change from the state of walking to the state of sitting. In the original dataset, these transition activities have been assigned to ordinary activity classes so that there is a total of 11 activities. In addition to the classification problem with 11 classes, a simplified (reduced) case with five classes (corresponding to activities 1, 4, 6, 8, and 9) is also considered by omitting the transition classes.

```
...
A01, 020-000-032-221, 633790227929538017, 27.05.2009 14:06:32:953, 2.1674482822418213, 1.7744109630584717, 0.768406093120575, standing up from lying
A01, 010-000-024-033, 633790227929808316, 27.05.2009 14:06:32:980, 2.083272695541382, 1.605885624885559, -0.019668469205498695, standing upfrom lying
A01, 020-000-033-111, 633790227930348903, 27.05.2009 14:06:33:033, 2.1095681190490723, 1.670161485671997, 1.096860647201538, standing up from lying
A01, 020-000-032-221, 633790227930619202, 27.05.2009 14:06:33:063, 2.1871273517608643, 1.8179994821548462, 0.7751449942588806, standing up from lying
A01, 010-000-024-033, 633790227930889495, 27.05.2009 14:06:33:090, 2.124642848968506, 1.3958414793014526, 0.05755209922790527, standing up from lying
A01, 010-000-030-096, 633790227931159787, 27.05.2009 14:06:33:117, 2.3755478858947754, 1.9687641859054565, -0.02469586208462715, walking
A01, 020-000-033-111, 633790227931430084, 27.05.2009 14:06:33:143, 2.162515878677368, 1.688720703125, 1.1396502256393433, walking
A01, 020-000-032-221, 633790227931700376, 27.05.2009 14:06:33:170, 2.1584346294403076, 1.7068990468978882, 0.7499973177909851, walking
A01, 010-000-024-033, 633790227931970674, 27.05.2009 14:06:33:197, 2.007559299468994, 1.7204831838607788, -0.04071690887212753, walking
A01, 020-000-032-221, 633790227932781551, 27.05.2009 14:06:33:277, 2.1662490367889404, 1.5868778228759766, 0.9013731479644775, walking
...
A01, 020-000-033-111, 633790227969271197, 27.05.2009 14:06:36:927, 3.2222139835357666, 1.994187593460083, 1.0909717082977295, walking
A01, 020-000-032-221, 633790227969541489, 27.05.2009 14:06:36:953, 3.3250277042388916, 2.288264036178589, 1.0457459688186646, walking
A01, 010-000-024-033, 633790227969811780, 27.05.2009 14:06:36:980, 3.237037420272827, 2.085507392883301, 0.31236138939857483, walking
A01, 010-000-030-096, 633790227970082071, 27.05.2009 14:06:37:007, 2.991684675216675, 1.9473466873168945, -0.052446186542510986, walking
A01, 020-000-033-111, 633790227970352368, 27.05.2009 14:06:37:037, 3.235618352890014б, 2.0799317359924316, 1.1940621137619019, walking
A02, 010-000-024-033, 633790230241677177, 27.05.2009 14:10:24:167, 3.8436150550842285, 2.038317918777466, 0.4496184289455414, walking
A02, 010-000-030-096, 633790230241947475, 27.05.2009 14:10:24:193, 3.288137435913086, 1.7760037183761597, 0.2172906994819641, walking
A02, 020-000-033-111, 633790230242217774, 27.05.2009 14:10:24:223, 3.7905502319335938, 2.10894513130188, 1.2178658246994019, walking
A02, 020-000-032-221, 633790230242488064, 27.05.2009 14:10:24:250, 4.826080322265625, 3.061596393585205, 2.016235589981079, walking
A02, 010-000-024-033, 633790230242758363, 27.05.2009 14:10:24:277, 3.889177083969116, 1.9832324981689453, 0.34168723225593567, walking
A02, 010-000-024-033, 633790230243839526, 27.05.2009 14:10:24:383, 3.8668057918548584, 2.037929058074951, 0.4540541470050812, walking
A02, 010-000-030-096, 633790230244109820, 27.05.2009 14:10:24:410, 3.724804401397705, 2.37532639503479, 0.43077021837234497, walking
A02, 020-000-033-111, 633790230244380118, 27.05.2009 14:10:24:437, 3.798861265182495, 2.0785067081451416, 1.1724120378494263, walking
A02, 020-000-032-221, 633790230244650412, 27.05.2009 14:10:24:467, 4.824059009552002, 3.062581777572632, 2.010768413543701, walking
A02, 010-000-024-033, 633790230244920703, 27.05.2009 14:10:24:493, 3.889165163040161, 2.053361177444458, 0.3612026274204254, walking
A02, 020-000-033-111, 633790230245461287, 27.05.2009 14:10:24:547, 3.8123855590820312, 1.9204648733139038, 1.1589833498001099, walking
...
```

Figure 2.2: Sample rows of the original dataset.


smoothed relative position of RFID tag 1 over time for the experiment A01

(a)

smoothed relative position of RFID tag 3 over time for the experiment A01

(b)

Figure 2.3: The positions of (a) tag 1 and (b) tag 3 in the first experiment of the first subject as 3-D curves whose gray level change from light gray to black in time.

2.2 Pre-processing of the Data

2.2.1 Curve-Fitting

Since the tag positions are acquired asynchronously and non-uniformly, feature extraction and classification based on the raw data would be very difficult. Thus, we first propose to fit a curve to the position data of each axis of each tag (a total of $3 \times 4 = 12$ axes) in each experiment and then re-sample the fitted curves uniformly at exactly the same time instants. Provided that the new, constant sampling rate is considerably higher than the average data acquisition rate, the curve-fitting and re-sampling process does not cause much loss of information. We assume that the sample values on the fitted curves (especially those that are far from the actual measurement points) represent the true positions of the tags since the tag positions do not change very rapidly. In general, the positions of the tags on the arms and the legs tend to change faster compared to the chest and the waist.

Three curve-fitting methods are considered in this work:
In *shape-preserving interpolation*, the fitted curve passes through the measurement points around which it is curvy and smooth but looks almost like straight lines in between. Hence, this method is very similar to linear (or first-order) interpolation except that the curve is differentiable everywhere. The fitted curve has high curvature, especially around the peaks.

The second method is *cubic-spline interpolation.* The curve in this method passes through the measurement points, like the previous one, but overall is much smoother. The fitted curve may oscillate unnecessarily in between the measurement points and may go far beyond the peaks of the measurements, in which case, it may not resemble (one axis of) the actual position curve of the tag.

The *smoothing spline* is the third method, having a single parameter adjustable between 0 and 1. It is observed that this method resembles shape-preserving interpolant when the parameter is chosen about 0.5 and cubic-spline interpolant when it is approximately 1. The parameter value should be chosen proportionately large with the complexity of the data, i.e., the number of available position measurements. In this study, we have used a parameter value of $1 - 10^{-6}$ for smoothing spline interpolation.

Although the third method seems to provide better results than the others, it is not feasible for long data as in this study since its computational complexity is much higher than the others. Therefore, we preferred to use shape-preserving interpolation because of its simplicity. Sample curves fitted to synthetic position data using the three methods are plotted in Figure 2.4. In addition, the x position of tag 4 in the fifth experiment of the fifth subject is plotted in Figure 2.5. Once the 12 different curves are fitted to the 12 axes of each experiment independently, the curves are re-sampled uniformly at exactly the same time instants.

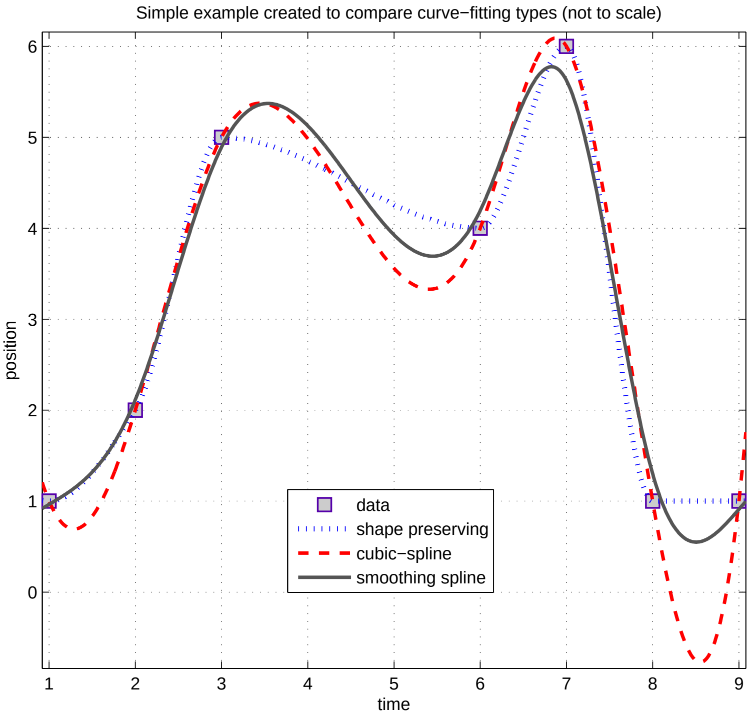
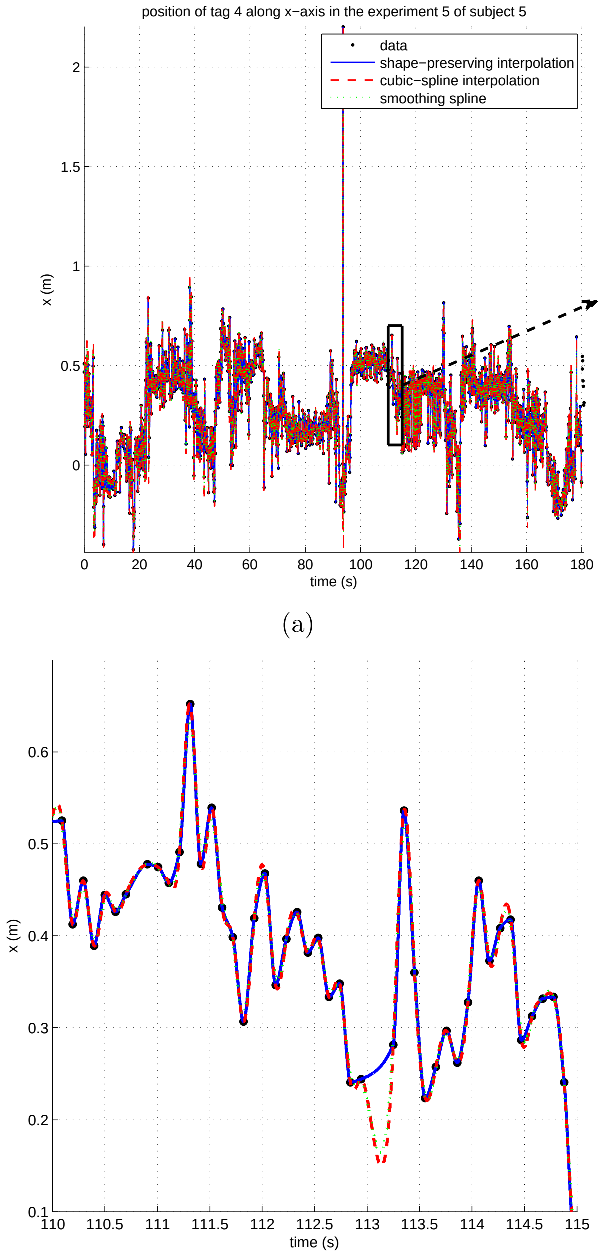


Figure 2.4: The three curve-fitting methods applied to synthetic position data.



(a)

(b)

Figure 2.5: The x position of tag 4 in the fifth experiment of the fifth subject. (a) The whole curve and (b) the zoomed-in version.

2.2.2 Segmentation

After the curve fitting and uniform re-sampling stages, the modified dataset now consists of $5 \times 5 = 25$ 2-D arrays (each corresponding to an experiment) with each line containing the time instant, position values along the 12 axes (three axes per tag) at that instant and the activity being performed at that time. Note that the number of rows is not the same in all of the experiments, since the duration of the experiment, hence the number of equally-spaced samples may differ slightly.

The 2-D array of each experiment is first divided into intervals containing samples corresponding to exactly one activity type. Then, each interval is divided into time segments of equal length, typically about one second. To prevent a segment from containing more than one activity, the following segmentation rule is used: For each experiment, progressing vertically along the 2-D array, a new segment is started only if the desired segment length is reached or a different activity is encountered. Naturally, the segments immediately before the transition points between activities and the very last segment may be shorter in length.

Classification is performed for each segment independently. While testing the classifiers and implementing the system in real time, the system needs to know where a new activity starts (i.e., the activity transition times). If this is not possible so that a constant segment duration is used, a segment may be associated with more than one activity. One can assign the longest activity contained in that segment as the true class, but this would unfairly decrease the classification accuracy. Since techniques for modeling activity durations and detecting the activity transition times are available [48, 52], we performed segmentation using the information on the true transition times so that each segment is associated with only a single activity.

2.3 Feature Extraction and Reduction

2.3.1 Feature Extraction

As stated above, each segment consists of many position samples in the corresponding time interval; each row of the dataset comprising 13 values (one time instant and 12 position values) as well as the true activity class. Thus, it would take a lot of time for a classifier to be trained and evaluated using the whole data. As an alternative, features extracted from the time segments are used for classification.

The features extracted for each of the 12 axes are the minimum, maximum, mean, variance, skewness, kurtosis values, the first few coefficients of the autocorrelation sequence, and the magnitudes of the five largest FFT coefficients. Therefore, there are (12 axes $\times[11 + \text{ceil}(N/2)]$ coefficients per axis) $= 132 + 12 \times \text{ceil}(N/2)$ coefficients in the feature vector, N being the maximum number of samples in a segment ($N = 5$ in this study). Note that the size of the feature vector increases with the maximum number of samples in a segment which, in turn, is the product of the sampling frequency (in Hz) and the segment duration (in s).

2.3.2 Feature Reduction

Because of the large number of features (about 150–200) associated with each segment, we expect feature reduction to be very useful in this scheme. The size of the feature vector is reduced by mapping the original high-dimensional feature space to a lower-dimensional one using principal component analysis (PCA) and linear discriminant analysis (LDA) [53]. PCA is a transformation that finds the optimal linear combination of the features in the sense that they represent the data with the highest variance in a feature subspace, without taking the intra-class and inter-class variances into consideration separately. It seeks a projection that best represents the data in a least-squares sense. On the other hand, LDA seeks a projection that best separates the data in the same sense and maximizes class separability [53]. Whereas PCA seeks rotational transformations that are efficient for representation, LDA seeks those that are efficient for discrimination. The best projection in LDA makes the difference between the class means as large as possible relative to the variance.

2.4 Classification

The following are the 10 different classifiers used in this study, with their corresponding PRTools [54] functions:

(1) `ldc`: Gaussian classifier with the same arbitrary covariance matrix for each class

(2) `qdc`: Gaussian classifier with different arbitrary covariance matrices for each class

(3) `udc`: Gaussian classifier with different diagonal covariance matrices for each class

(4) `mogc`: mixture of Gaussians classifier (with two mixtures)

(5) `naivebc`: naïve Bayes classifier

(6) `knnc`: k-nearest neighbor (k-NN) classifier

(7) `kernelc`: dissimilarity-based classifier

(8) `fisherc`: minimum least squares linear classifier

(9) `nmc`: nearest mean classifier

(10) `nmsc`: scaled nearest mean classifier

Detailed descriptions of these classifiers can be found in [53].

2.5 Performance Evaluation through Cross Validation

Since each of the five subjects repeats the same sequence of activities five times, the procedures used for training and testing affect the classification accuracy. For this reason, two different cross-validation techniques are used for evaluating the classifiers: P-fold and subject-based leave-one-out (L1O) [53].

In P-fold cross validation ($P = 5$ in this thesis), the whole set of feature vectors is divided into P partitions, where the feature vectors in each partition are selected completely randomly, regardless of the subject or the class they belong to. One of the P partitions is retained as the validation set for testing, and the remaining $P - 1$ partitions are used for training. The cross-validation process is then repeated P times (the folds), so that each of the P partitions is used exactly once for validation. The P results from the folds are then averaged to produce a single estimate of the overall classification accuracy.

In subject-based L1O cross validation, partitioning of the dataset is done subject-wise instead of randomly. The feature vectors of four of the subjects are used for training and the feature vectors of the remaining subject are used in turn for validation. This is repeated five times such that the feature vector set of each subject is used once as the validation data. The five correct classification rates are averaged to produce a single estimate. This is same as P-fold cross validation with P being equal to the number of subjects ($P = 5$) and all the feature vectors in the same partition being associated with the *same* subject.

Although these two cross-validation methods use all the data equally in training and testing of the classifiers, there are two factors that affect the results obtained based on the same data. The first one is the random partitioning of the data in the P-fold cross-validation technique that slightly affects the classification accuracy. Secondly, classifier 7 (dissimilarity-based classifier) includes randomness in its nature. Therefore, both cross-validation methods are repeated five times and the average classification accuracy and its standard deviation are calculated over the five executions. This way, we can assess the repeatability of the results and estimate how well the system will perform over newly acquired data from unfamiliar subjects.

2.6 Experimental Results

The following are the adjustable parameters or factors that possibly affect the classification accuracy, with their default values written in square brackets:

(1) `fs`: sampling frequency of the fitted curves in forming the modified data (in Hz) [default: 10 Hz]

(2) `frm_dur`: maximum segment duration (in seconds) [default: 0.5 s]

(3) `curve_fit_type`: the curve-fitting algorithm
(1: shape-preserving interpolation, 2: cubic-spline interpolation,
3: smoothing spline) [default: 1]

(4) `pri`: prior probabilities of the 11 classes (i.e., activities)
(0: equal priors for each class,
1: priors calculated based on the class frequencies) [default: 1]

(5) `reduc`: the feature reduction type if used and the dimension of the reduced feature space
(0: no feature reduction; $+|n|$: PCA with reduced dimension n;
$-|n|$: LDA with reduced dimension n) [default: 0]

All the classifiers are trained and tested using different combinations of the parameters described above. Then, for each classifier, the set of parameters that result in the lowest average classification error are determined. This process is repeated for both cases (the complete and the simplified classification problems with 11 and 5 classes, respectively) and both cross-validation methods (5-fold and subject-based L1O). Average classification errors of the classifiers over the five executions and their standard deviations are tabulated in Table 2.1. It is observed that the k-NN classifier (with $k = 5$) is the best one among the 10 classifiers compared in this study and outperforms the other classifiers in all cases. For the complete classification problem with 11 classes, the k-NN classifier has an average classification error of 8.67% and 21.30%, whereas for the reduced case with five classes, these numbers are 1.12% and 6.52% for 5-fold and subject-based L1O cross-validation, respectively. Note that since the partitions are fixed in subject-based L1O cross validation, this technique gives the same result if the complete cycle over the subject-based partitions is repeated. Therefore, its standard deviation is zero except for classifier 7 that includes randomness. For the k-NN classifier, the cumulative confusion matrices obtained by summing up the confusion matrices of each run in all of the five executions are presented in Tables 2.2 and 2.3 for the 11-class and 5-class problems, respectively, using the two cross-validation techniques.

The parameters listed above significantly affect the classification accuracy. Therefore, for each parameter, the tests are run by varying that parameter while keeping the remaining ones constant at their default values. The variation of the average classification error with each of these parameters is shown in Figures 2.6–2.10 for the two cross-validation methods and for both the complete and simplified classification problems (total of four cases). All the error percentage values presented in these figures are the average values over the five executions. Because the k-NN classifier (classifier 6) outperforms all of the other classifiers, the average classification error of only this classifier is shown in the figures. As expected, the 11-class classification problem results in larger errors compared to the 5-class problem. From the results, it can be observed that in all cases, 5-fold cross validation provides better results than subject-based L1O. This is because in the first case, the system is trained and tested with a random mixture of different subjects' data, whereas in the second, it is trained with the data of four of the subjects and tested with the data of the remaining subject which is totally new to the system.

	classifier	error percentage ± standard deviation (with the corresponding parameter values `fs`, `frm_dur`, `curve_fit_type`, `reduc`)[1]			
		11 classes		5 classes	
		5-fold	subject-based L1O	5-fold	subject-based L1O
1	Gaussian classifier with the same arbitrary covariance matrix for each class	**20.89 ± 0.02** (10, 0.5, 1, –8)	**23.52** (10, 0.5, 3, –8)	**6.06 ± 0.01** (10, 0.5, 1, –4)	**7.63** (10, 0.5, 1, –4)
2	Gaussian classifier with different arbitrary covariance matrices for each class	**20.76 ± 0.04** (10, 0.5, 1, –6)	**23.69** (10, 0.5, 3, –8)	**5.48 ± 0.02** (10, 0.5, 1, –4)	**7.73** (10, 0.5, 1, –4)
3	Gaussian classifier with different diagonal covariance matrices for each class	**21.65 ± 0.05** (10, 0.5, 1, –6)	**24.11** (10, 0.5, 3, –3)	**6.02 ± 0.02** (10, 0.5, 1, –4)	**7.92** (10, 0.5, 1, –4)
4	mixture of Gaussians classifier (with two mixtures)	**20.75 ± 0.07** (10, 0.5, 1, –6)	**23.70** (10, 0.5, 3, –8)	**5.19 ± 0.03** (10, 0.5, 1, –4)	**7.63** (10, 0.5, 1, –4)
5	naïve Bayes classifier	**22.84 ± 0.16** (10–.5, 1, –8)	**24.46** (10, 0.5, 3, –6)	**7.24 ± 0.16** (10, 0.5, 1, –4)	**9.49** (10, 0.5, 1, –4)
6	k-NN classifier ($k = 5$)	**8.67 ± 0.10** (10, 0.5, 3, 0)	**21.30** (10, 0.5, 1, –10)	**1.12 ± 0.04** (10, 0.2, 1, 0)	**6.52** (10, 0.2, 1, 0)
7	dissimilarity-based classifier	**19.41 ± 0.16** (10, 0.5, 1, –8)	**22.10 ± 0.04** (10, 0.5, 3, –10)	**4.66 ± 0.16** (10, 0.5, 1, –4)	**7.33 ± 0.03** (10, 0.5, 1, –4)
8	minimum least squares linear classifier	**25.26 ± 0.52** (10, 0.5, 3, –10)	**26.83** (10, 1, 3, 0)	**7.37 ± 0.17** (10, 0.5, 1, –4)	**9.30** (10, 1, 1, 0)
9	nearest mean classifier	**23.60 ± 0.06** (10, 0.5, 3, –10)	**27.02** (10, 0.5, 3, –10)	**6.07 ± 0.02** (10, 0.5, 1, –4)	**8.21** (10, 0.5, 1, –4)
10	scaled nearest mean classifier	**20.94 ± 0.04** (10, 0.5, 3, –8)	**23.59** (10, 0.5, 3, –8)	**6.04 ± 0.03** (10, 0.5, 1, –4)	**7.58** (10, 0.5, 1, –4)

Table 2.1: The average probability of error (weighted by prior probabilities) of the 10 classifiers, for the 11- and 5-class problems and the two cross-validation techniques. In each case, the combination of parameters leading to the most accurate classifier is given in parantheses.

`pri` = 1 throughout the table; that is, prior probabilities are calculated from the class frequencies.

cumulative confusion matrix of classifier 6 (k-NN) for 5-fold (average classification error: 8.67%)

(fs = 10, frm_dur = 0.5, curve_fit_type = 3, pri = 1, reduc = 0)

true	estimated labels											
labels	1	2	3	4	5	6	7	8	9	10	11	total
1	10,057	177	158	171	113	45	298	45	23	56	52	11,195
2	139	606	18	124	16	94	42	0	62	16	3	1,120
3	139	14	1,752	146	3	6	64	65	30	1	15	2,235
4	126	108	131	17,556	14	7	310	45	6	6	11	18,320
5	124	34	6	30	321	96	22	0	4	13	0	650
6	41	59	6	19	60	8,758	3	0	20	67	7	9,040
7	305	45	54	305	10	13	5,691	5	5	1	11	6,445
8	30	1	52	76	0	0	7	1,612	0	0	2	1,780
9	16	42	26	5	2	9	9	0	3,729	0	52	3,890
10	74	19	0	12	9	54	10	0	0	332	0	510
11	50	2	9	11	0	5	3	0	54	0	846	980
total	11,101	1,107	2,212	18,455	548	9,087	6,495	1,772	3,933	492	999	56,165

cumulative confusion matrix of classifier 6 (k-NN) for subject-based L1O (average classification error: 21.30%)

(fs = 10, frm_dur = 0.5, curve_fit_type = 1, pri = 1, reduc = −10)

true	estimated labels											
labels	1	2	3	4	5	6	7	8	9	10	11	total
1	10,645	30	65	15	30	195	190	0	5	10	10	11,195
2	305	340	10	135	10	115	170	5	25	0	5	1,120
3	215	20	510	415	15	80	850	85	0	10	35	2,235
4	40	100	220	16,665	0	30	575	350	285	0	55	18,320
5	180	20	25	5	110	190	95	0	0	25	0	650
6	310	30	45	20	85	8,390	135	0	10	0	15	9,040
7	685	110	355	1,385	50	235	3,110	215	75	25	200	6,445
8	5	10	50	625	5	20	370	630	10	0	55	1,780
9	0	50	0	135	5	175	60	0	3,380	0	85	3,890
10	180	5	30	0	40	150	80	0	5	20	0	510
11	110	0	0	25	20	60	200	5	140	0	420	980
total	12,675	715	1,310	19,425	370	9,640	5,835	1,290	3,935	90	880	56,165

Table 2.2: Cumulative confusion matrices for classifier 6 (k-NN) for the 11-class problem. The confusion matrices are summed up for the five executions of the 5-fold (top) and subject-based L1O (bottom) cross validation.

cumulative confusion matrices of classifier 6 (k-NN)

(`fs` = 10, `frm_dur` = 0.2, `curve_it_type` = 1, `pri` = 1, `reduc` = 0)

	5-fold (avg. classification error: 1.12%)						subject-based L1O (avg. classification error: 6.52%)					
true	estimated labels						estimated labels					
labels	1	4	6	8	9	total	1	4	6	8	9	total
1	27,041	10	226	7	6	27,290	26,850	15	350	50	50	27,290
4	39	45,051	15	206	4	45,315	405	42,905	470	1,040	495	45,315
6	145	3	22,296	10	11	22,465	1,050	25	21,030	285	75	22,465
8	59	361	74	3,885	1	4,380	270	735	795	2,540	40	4,380
9	10	0	29	0	9,591	9,630	90	75	775	0	8,690	9,630
total	27,294	45,425	22,640	4,108	9,613	109,080	28,665	43,755	23,420	3,890	9,350	109,080

Table 2.3: Cumulative confusion matrices for classifier 6 (k-NN) for the 5-class problem. The confusion matrices are summed up for the five executions of the 5-fold (left) and subject-based L1O (right) cross validation.

2.6.1 Effect of the Sampling Frequency (`fs`)

When the sampling frequency is set too low, in particular, 6 Hz, the classification accuracy is surprisingly acceptable. This is because the movement of the RF tags is not very fast, and the classification performance does not degrade much when the high-frequency components are removed.

The average classification error increases slightly with increasing sampling rate (Figure 2.6). For instance, with `fs` = 400 Hz, noting that the dimension of the feature space also increases with the number of samples in a segment, the data becomes too complicated that it misleads most of the classifiers. This is because the position measurements are quite noisy; hence, selecting a high sampling rate may cause over fitting, which in turn degrades the classification accuracy of the system. A suitable value of `fs` is determined as 10 Hz and is set to be the default value.

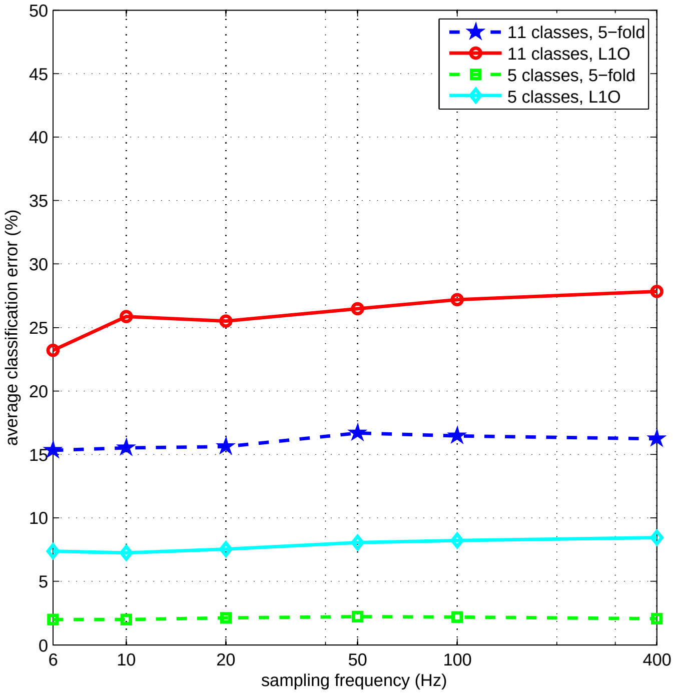


Figure 2.6: Effect of the sampling frequency on the average classification error of the k-NN classifier (classifier 6).

2.6.2 Effect of the Segment Duration (`frm_dur`)

Since a single event or activity is associated with each segment, the segment duration is another parameter that affects the accuracy. Results for segment duration values between 0.2 s and 1 s are shown in Figure 2.7. Although the smallest segment duration gives slightly better results in most cases, the system should make a decision five times in a second with this segment duration, which increases the complexity. In fact, even a very short segment consisting of a single position measurement (one row of the dataset) is sufficient to obtain the body posture information since it directly provides the 3-D positions of the tags on different body parts at that instant. Compromising between complexity and accuracy, a segment duration of 0.5 s is selected as the default value without much loss in the classification accuracy in each of the four cases.

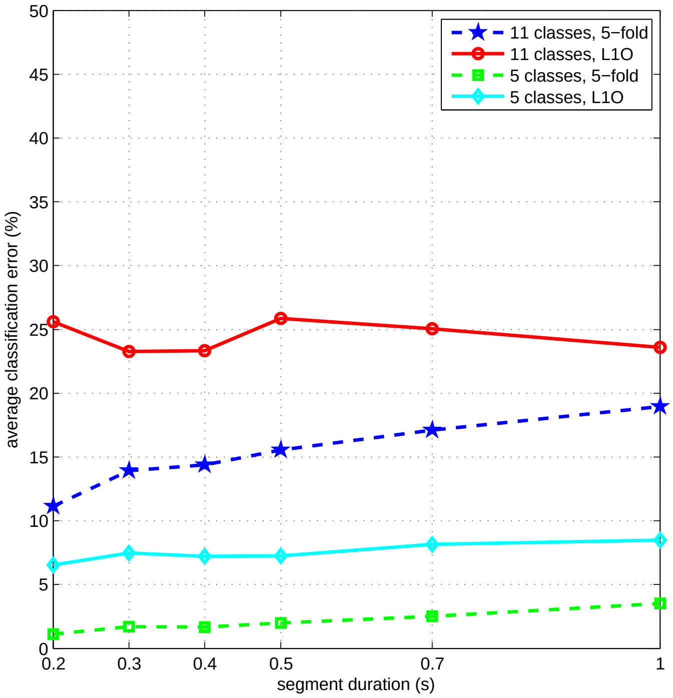


Figure 2.7: Effect of the the segment duration on the average classification error of the k-NN classifier (classifier 6).

2.6.3 Effect of the Curve-Fitting Algorithm

Referring to Figure 2.8, it is observed that shape-preserving interpolation and cubic-spline interpolation give very similar results for subject-based L1O whereas the smoothing spline interpolation leads to poorer classification accuracy. The latter is the best curve-fitting method in the particular case of 5-fold cross-validation with 11 classes. Thus, shape-preserving interpolation is chosen as the default curve-fitting method.

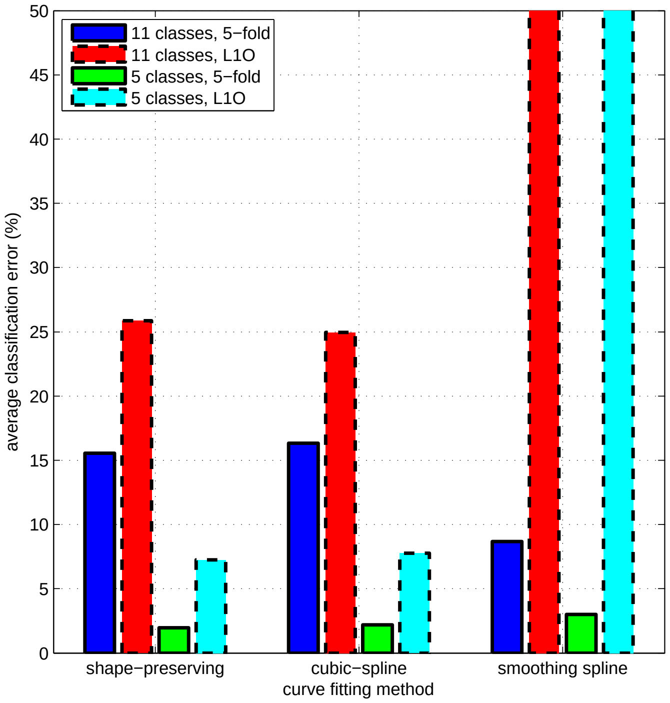


Figure 2.8: Effect of the curve-fitting method on the average classification error of the k-NN classifier (classifier 6).

2.6.4 Effect of the Prior Probabilities

Classification errors for the individual classes can be obtained from the confusion matrices provided in Tables 2.2 and 2.3. The average probability of error is calculated by weighting the classification error of each class with its prior probability. In this study, prior class probabilities have been chosen in two different ways. In the first, prior probabilities are taken equal for each class, whereas in the second, prior probabilities are set equal to the actual occurrence of the classes in the data. Figure 2.9 illustrates the effect of prior probabilities on the average classification error. It is observed in the figure that the error for the case with equal priors is larger. This is because the transition classes (Section 2.2.2) rarely occur in the dataset and their probability of occurrence is extremely low. However, the classification errors for these classes are larger. When a weighted average is calculated using the actual class probabilities, those terms with large classification error contribute relatively less to the total average error.

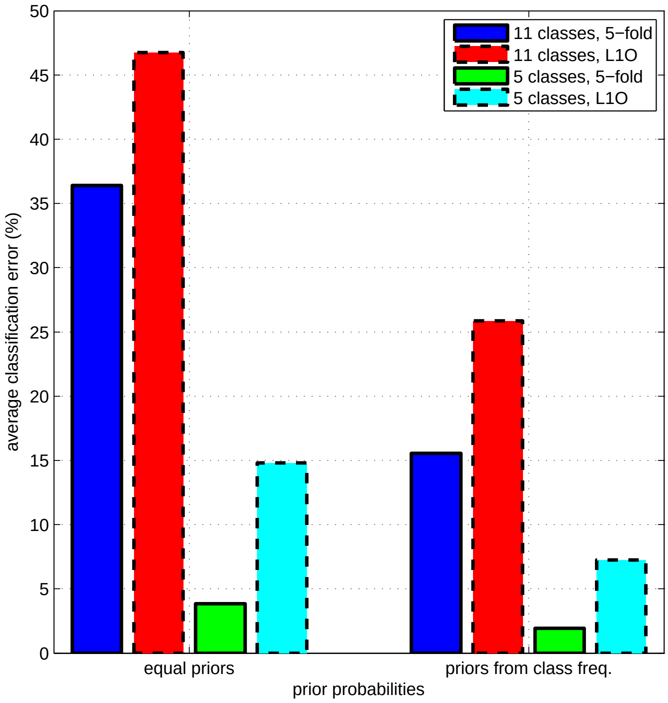


Figure 2.9: Effect of the prior probabilities on the average classification error of the k-NN classifier (classifier 6).

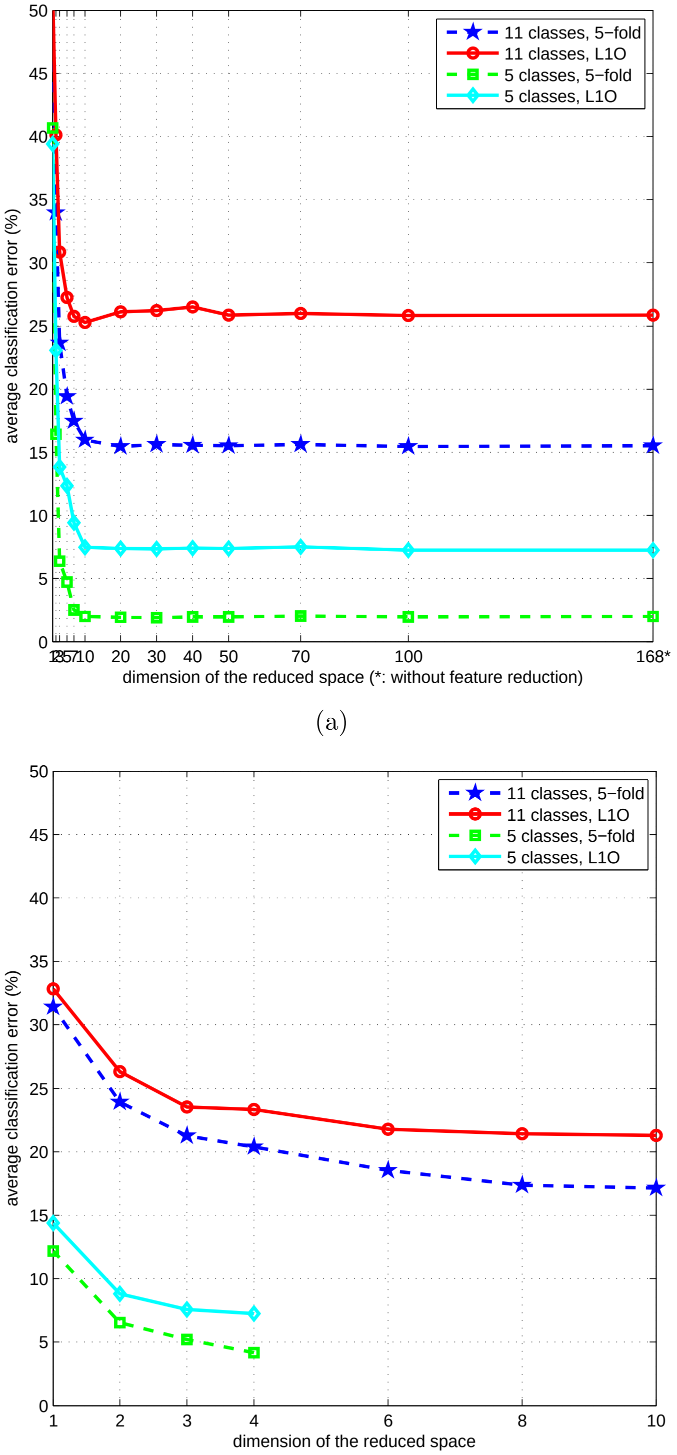
2.6.5 Effect of Feature Reduction

Since there is a large number of features (depending on the sampling frequency and the segment duration), two common methods (PCA and LDA) are used for feature reduction. All the results up to this point, including Figures 2.6–2.9, are obtained without feature reduction.

Figure 2.10(a) shows the average classification error when PCA is used with different reduced dimensionalities (from 1 to 100) as well as the case without feature reduction (168). It can be observed that the intrinsic dimensionality of the feature vectors is about 10, which is much smaller than the actual dimension.

Figure 2.10(b) corresponds to the cases where LDA is used with reduced dimension from 1 to 10 for 11 classes and from 1 to 4 for 5 classes (note that reduced dimension must be less than the number of classes in LDA). For the complete classification problem with 11 classes, LDA with dimension 10 outperforms all other cases including the ones without feature reduction validated by subject-based L1O. Including too many features not only increases the computational complexity of the system significantly, but also confuses the classifiers, leading to a less accurate system (this is known as "the curse of dimensionality").

For the 11-class problem, LDA with dimension 10 performs better than PCA when L1O is used and worse when 5-fold cross validation is employed. For the simplified problem with 5 classes, the results change similarly with feature reduction. With subject-based L1O, LDA with dimension four outperforms PCA with higher dimensions as well as the case without feature reduction. When 5-fold cross-validation is used, PCA with dimension 20 is the best one. Hence, LDA with dimension four is preferable because its performance seems to be less dependent on the subject performing the activities. Therefore, it can be stated that LDA is more reliable if the system is going to be used with subjects who are not involved in the training process. On the other hand, if the system is going to be trained for each subject separately, PCA results in a more accurate classifier, even at the same dimensionality with LDA.



(a)

(b)

Figure 2.10: Effect of feature reduction with (a) PCA and (b) LDA on the average classification error of the k-NN classifier (classifier 6).

2.7 Conclusion

In this study, a novel approach to human activity recognition is presented using a tag-based RF localization system. Accurate activity recognition is achieved without the need to consider the interaction of the subject with the objects in its environment. In this scheme, the subjects wear four RF tags whose positions are measured via multiple antennas (readers) fixed to the environment.

The most important issue is the asynchronous and non-uniform acquisition of the position data since the system records measurements whenever it detects a tag, and the detection frequency is affected by the SNR and interference in the environment. The asynchronous nature of the data acquired introduces some additional problems to be tackled—only one tag can be detected at a given time instant; hence, the measurements of different tags are acquired at different time instants in a random manner. This problem has been solved by first fitting a suitable curve to each measurement axis along time, and then re-sampling the fitted curves uniformly at a higher sampling rate at exactly the same time instants.

After the uniformly-sampled curves are obtained, they are partitioned into segments of maximum duration of one second each such that each segment is associated with only a single activity type. Then, various features are extracted from the segments to be used in the classification process. The number of features are reduced using two feature reduction techniques.

Ten different classifiers are investigated and their average classification errors are calculated for various curve-fitting and feature reduction techniques and system parameters. In calculating the average classification error, two different cross-validation techniques, namely P-fold with $P = 5$ and subject-based L1O, are used. Omitting the transition classes, the complete pattern recognition problem with 11 classes is reduced to a problem with five classes and the whole process is repeated. Finally, the set of parameters and the classifier giving the best result is presented for each problem and for each cross-validation method.

For the complete problem with 11 classes, the proposed system has an average classification error of 8.67% and 21.30% when the 5-fold and subject-based

L1O cross-validation techniques are used, respectively. This relatively large error is caused by the transition activities of much shorter duration that are more difficult to recognize. When these activities are discarded, the reduced system with five classes has an average probability of error of 1.12% and 6.52% when 5-fold and subject-based L1O cross validation are used, respectively. Hence, the performance significantly improves with the removal of the transition activities, as expected. The system proposed here demonstrates acceptable performance for most practical applications.

Chapter 3

Investigation of Personal Variations in Activity Recognition Using Inertial Sensors and Magnetometers

3.1 Introduction and Related Work

The use of wearable miniature inertial sensors and magnetometers in activity recognition has pervaded due to their high portability and low cost. This approach has many advantages over vision-based systems [2–5] and often provides high classification accuracy in activity recognition [35, 36]. On the other hand, independent of the method used for human activity recognition, the acquired data for classification significantly varies between subjects in various ways. More specifically, miniature inertial sensor signals vary in amplitude and speed for different subjects according to their personal styles and anthropometry (i.e., physical attributes). The change in time is often nonlinear and may be difficult for an artificial system to perceive. Therefore, in general, the classification accuracy degrades significantly if activities of a subject are attempted to be recognized using another subjects' training data, as in the subject-based leave-one-out (L1O) test [35].

To the best of our knowledge, inter-subject variability of sensor data in activity recognition has not been examined so far, although it is investigated in detail in vision-based systems [55–57]. In [58], the intra-subject variation of the accelerometer data of the activity checklist performed by seventeen male patients having chronic obstructive pulmonary disease is studied and observed to be low. For this reason, we calculate the average inter-subject variations of the sensor data for each activity by applying different similarity measures to time-domain signals. Based on this inter-subject distance, we propose a method to identify who performs the activities in the best way. This is analogous to the method used in post-stroke rehabilitation where a system providing real-time feedback to neurological patients undergoing motor rehabilitation is developed by applying a modified version of dynamic time warping (DTW) to the data acquired from a sensorized long-sleeve shirt containing strain sensors [41]. In that system, reference data is captured while the subject is performing the movements correctly and incorrectly in a supervised manner. Then, during the exercises, the system is able to distinguish the type of movement as well as how accurately the subject executes it in real-time with a reasonable error.

The work reported in this chapter is the extended form of our earlier research presented in [59, 60]. Data acquired from five sensor units worn on the human body, each containing a tri-axial accelerometer, a tri-axial gyroscope, and a tri-axial magnetometer, during 19 different human activities are used to calculate inter-subject and inter-activity variations. Different methods are used and the results are summarized in various forms. Absolute, Euclidean, and DTW distances are used to assess the similarity of the signals. The comparisons are made based on the raw data, their normalized versions, and feature vectors extracted from the raw data. First, inter-subject distances are averaged out per activity and per subject. Based on these values, the 'best' subject is defined and identified according to his/her average distance to the others. Then, the averages and standard deviations of inter-activity distances are presented per subject, per sensor unit, and per sensor type. The effect of removing the mean value and the use of different distance measures on the results are discussed.

Organization of this chapter is as follows: In Section 3.2, the dataset is described briefly. The distance measures and normalization methods are presented in Sections 3.3 and 3.4. Then, identification of the best subject is explained (Section 3.5.1), average inter-subject distance is calculated per activity (Section 3.5.2), and average means and standard deviations of inter-activity distances are investigated per subject, per sensor unit, and per activity, separately (Section 3.5.3). The chapter is concluded in Section 3.7.

3.2 Dataset

The dataset used in this study is the same as the one used in reference [35]. In the experiments, 8 different subjects wearing 5 miniature sensor units performed 19 activities, each lasting 5 min. The physical characteristics of the subjects can be found in [36]. The activities are the following:

> Sitting (A1), standing (A2), lying on back side (A3), lying on right side (A4), ascending stairs (A5), descending stairs (A6), standing in an elevator still (A7), moving around (A8), walking in a parking lot (A9), walking on a treadmill with a speed of 4 km/h in flat position (A10), walking on a treadmill with a speed of 4 km/h in 15° inclined position (A11), running on a treadmill with a speed of 8km/s (A12), exercising on a stepper (A13), exercising on a cross trainer (A14), cycling on an exercise bike in, horizontal position (A15), cycling on an exercise bike in vertical position (A16), rowing (A17), jumping (A18), playing basketball (A19).

Therefore, in the dataset, for each of the 8 subjects and 19 activities, there are 45 (5 units × 9 sensors) time-domain signals of length 5 min, sampled at 25 Hz, and consisting of 7,500 samples each. The units of the signals differ because a mixture of different sensor types is used.

Each time-domain signal is represented with $x_{\dots}(t)$ or its sampled version $x_{\dots}[n]$, i.e.,

$$x_{p,a,u,s}[n] = x_{p,a,u,s}(t)\Big|_{t=\frac{n}{25}} \tag{3.1}$$

where $0 \leq t \leq 300$ sec, $1 \leq n \leq 7{,}500$, $p \in [1, 8]$ is the subject index, $a \in [1, 19]$ is the activity index, $u \in [1, 5]$ is the unit index, and $s \in [1, 9]$ is the sensor index. The number of subjects, activities, sensors, and units are $N_p = 8$, $N_a = 19$, $N_u = 5$, and $N_s = 9$, respectively.

Feature vectors are calculated based on the time-domain signals to reduce the amount of data, in exactly the same way as in [35]. Since each activity of each subject is recorded for 5 min, and the recording is divided into 5-second segments, a total of $N_k = 60$ $(= \frac{5\times 60}{5})$ feature vectors are extracted for each activity of each subject. Each feature vector consists of specific properties of the same 5-second segment of all the 45 time-domain signals of a particular activity of a particular subject. The features for (each segment of) each axis are the following: the minimum, maximum, mean, variance, skewness, kurtosis, autocorrelation sequence, and the peaks of the discrete Fourier transform (DFT) with the corresponding frequency values. Feature vectors are denoted by $\mathbf{v}_{p,a}\{k\}$, where $k \in [1, 60]$ is the segment index.

3.3 Distance Measures

Three common distance measures are used to calculate the distance between the two sequences $x[n]$ and $y[n]$:

(1) absolute (taxicab) distance:

$$d_{\text{abs}}\left(x[n], y[n]\right) = \sum_{i=1}^{N} |x[i] - y[i]| \tag{3.2}$$

(2) Euclidean distance:

$$d_{\text{Euc}}\left(x[n], y[n]\right) = \sqrt{\sum_{i=1}^{N} \left(x[i] - y[i]\right)^2} \tag{3.3}$$

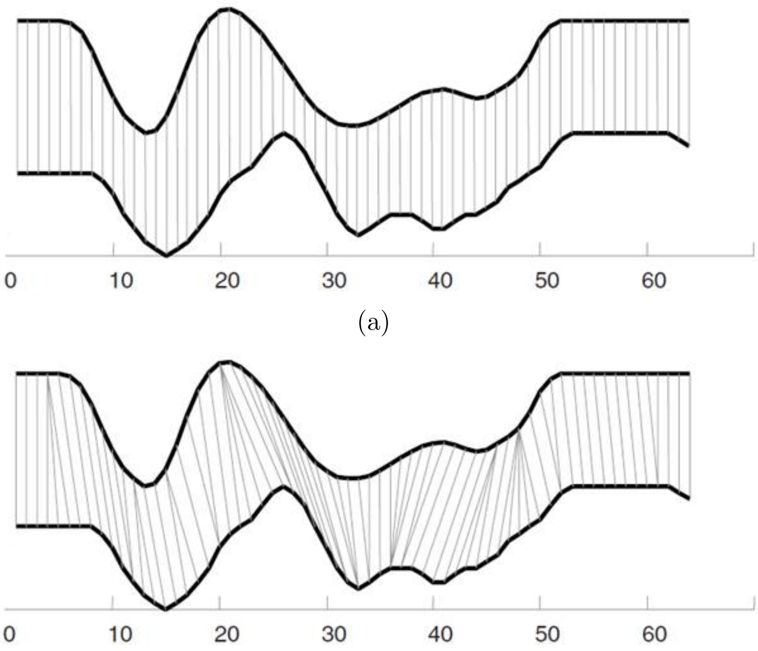
(3) DTW distance[1]:

$$d_{\text{DTW}}\left(x[n], y[n]\right) = \text{DTW}\left(x[n], y[n]\right) \tag{3.4}$$

Here, $x[n]$ and $y[n]$ are the discrete-time sequences with $1 \leq n \leq N$, and $d_{\mathfrak{D}}(x[n], y[n])$ is the distance between them, $\mathfrak{D}$ being one of the three distance measures. Note that the two sequences must be of the same length for absolute and Euclidean distances, whereas there is no such constraint in DTW distance. Therefore, DTW distance is applicable to the more general case where $x[n]$ and $y[n]$ have different lengths.

In the DTW distance measure, the sequences $x[n]$ and $y[n]$ are matched by "elastically" transforming their time (or sample) axes such that they are most similar to each other. In this way, the local minima, the local maxima, and similar shapes in the sequences are matched to each other as much as possible. Then, the Euclidean (or another type of) distance is calculated between the matched

[1] See Appendix A for the DTW algorithm.



(a)

(b)

Figure 3.1: Comparison of the Euclidean and DTW distance measures. (a) The Euclidean distance compares the samples at the same time instants, whereas (b) the DTW distance compares the samples that belong to similar shapes with each other to minimize the distance.

sequences to obtain the DTW distance. The DTW algorithm is summarized in Appendix A. See Figure 3.1 for an illustration that compares the Euclidean and DTW distance measures.

3.4 Normalization

Although the sensors used in data acquisition are calibrated, the measurements may still be biased, resulting in a constant error in the time-domain signals. Hence, even if the mean values of the signals provide information about the activities, they may not be correct due to sensor biases. In addition, the DTW distance measure depends mostly on the shape of signals. Therefore, the first type of normalization involves removing the mean values:

$$\overline{x}_{p,a,u,s}[n] = x_{p,a,u,s}[n] - \operatorname*{mean}_{n} x_{p,a,u,s}[n] \tag{3.5}$$

If a signal is compared with a biased version of itself, the distance between them may be very different depending on which distance measure is used. Suppose that the aforementioned sequences are $x[n]$ and $y[n] = x[n] + e[n]$, $1 \leq n \leq N$, where $e[n] \equiv E > 0$ is the constant bias error between the sequences. Then, according to the three distance functions, the distance between $x[n]$ and $y[n]$ is

$$\begin{align}
d_{\text{abs}}\left(x[n], y[n]\right) &= \sum_{i=1}^{N} |E| = NE \tag{3.6}\\
d_{\text{Euc}}\left(x[n], y[n]\right) &= \sqrt{\sum_{i=1}^{N} E^2} = \sqrt{NE^2} = \sqrt{N}E \tag{3.7}\\
d_{\text{DTW}}\left(x[n], y[n]\right) &\leq NE \tag{3.8}
\end{align}$$

Therefore, the distance between a sequence and its biased version is directly proportional to the amount of bias E and also depends on the sequence size. As a numerical example, for $N = 100$ and $E = 0.01$, $d_{\text{abs}}\left(x[n], y[n]\right) = 1$, $d_{\text{Euc}}\left(x[n], y[n]\right) = 0.1$, and $d_{\text{DTW}}\left(x[n], y[n]\right) \leq 1$.

In addition to removing the mean values, the variance of the signals can also be normalized to 1 by scaling them with their standard deviation. Thus, the second normalization type is obtained by

$$\widetilde{x}_{p,a,u,s}[n] = \frac{1}{\operatorname{std}_n \overline{x}_{p,a,u,s}[n]} \overline{x}_{p,a,u,s}[n]. \tag{3.9}$$

The third normalization type corresponds to limiting the sequence to the interval $[-1, 1]$ by shifting and scaling the signal, which is common especially before applying DTW:

$$\underline{x}_{p,a,u,s}[n] = 2\frac{x_{p,a,u,s}[n] - \min_n x_{p,a,u,s}[n]}{\max_n x_{p,a,u,s}[n] - \min_n x_{p,a,u,s}[n]} - 1 \tag{3.10}$$

3.5 Inter- and Intra-Subject Variations in Activity Recognition Using Inertial Sensors and Magnetometers

3.5.1 Identifying the 'Best' Subjects

In this section, the average distance between the whole data of one subject and all other subjects is calculated to identify the 'best' subjects in terms of their similarities to the others. The three different distance measures are applied to both raw (i.e., unnormalized) and normalized time-domain signals in different ways as well as raw and zero-mean feature vectors. The complete data is used unless the calculations last too long; i.e., if the relatively slow distance function, namely DTW, is used (the percentage of data used in the calculations is always explicitly specified).

3.5.1.1 Comparison Algorithm

A measure of similarity of two subjects based on their activity data in the time domain is proposed as follows: For each activity and for each sensor of each unit, the distances between the time-domain signals of the subjects are calculated and averaged out for all activities, sensors, and units. Then, their mean is considered as 'the distance between the two subjects,' resulting in:

$$d^{\text{time-domain}}_{\text{intra-subject, }\mathfrak{D}}\left(p_1, p_2\right) = \frac{1}{N_a N_u N_s} \sum_a \sum_u \sum_s d_{\mathfrak{D}}\left(x_{p_1,a,u,s}[n], x_{p_2,a,u,s}[n]\right) \tag{3.11}$$

where $\mathfrak{D}$ is one of the distance measures.

To compare two subjects based on their feature vectors, the distance between the feature vectors of the two subjects are averaged out for all instances, i.e.,

$$d^{\text{features}}_{\text{intra-subject, }\mathfrak{D}}\left(p_1, p_2\right) = \frac{1}{N_k} \sum_k d_{\mathfrak{D}}\left(\mathbf{v}_{p_1,a}\{k\}, \mathbf{v}_{p_2,a}\{k\}\right). \tag{3.12}$$

To identify those subjects that are most similar to the others in terms of the average distance sense, the distances between all the subject pairs are calculated. Then, for each subject, the distances from him/her to all other subjects are averaged out, resulting in the average distance of the subject to the others. If the time-domain signals are used, this is given by:

$$d_{\text{avg-subject, } \mathfrak{D}}^{\text{time-domain}}(p) = \frac{1}{N_p - 1} \sum_{p_1 \neq p} d_{\text{intrasubject, } \mathfrak{D}}^{\text{time-domain}}(p, p_1) \tag{3.13}$$

If the feature vectors are used, the corresponding expression is:

$$d_{\text{avg-subject, } \mathfrak{D}}^{\text{features}}(p) = \frac{1}{N_p - 1} \sum_{p_1 \neq p} d_{\text{intrasubject, } \mathfrak{D}}^{\text{features}}(p, p_1) \tag{3.14}$$

The subject with the smallest distance to all the others in the average distance sense, i.e., the person who performs the activities most similar to the others in the average distance sense is considered to be the ‘best’ person in this scheme:

$$\text{Best Subject}_{\mathfrak{D}}^{\text{time-domain}} = \arg\min_{p} d_{\text{avg-subject, } \mathfrak{D}}^{\text{time-domain}}(p) \tag{3.15}$$

or

$$\text{Best Subject}_{\mathfrak{D}}^{\text{features}} = \arg\min_{p} d_{\text{avg-subject, } \mathfrak{D}}^{\text{features}}(p) \tag{3.16}$$

if time-domain signals or feature vectors are used, respectively.

Although the best subject may not be the person who performs the activities most correctly, s/he is the one in the middle; at least s/he is not performing them in any extreme way. For instance, if the dataset contains only the walking activity and the subjects differ only in their walking speed, this approach would identify the subject who walks nearest to the average speed among all subjects as the ‘best’ one.

3.5.1.2 Percentage of Data to Use

When DTW is used to calculate the distances, the calculations last approximately 100 times longer than the cases where the absolute or Euclidean distances are used. For that reason, and because there are many instances of each activity in the dataset, calculations are sped up by using a small percentage of the data in the DTW case.

In time-domain signals, noting that the computational complexity of DTW is proportional to the product of the lengths of the two sequences to be compared (and thus the length square if the two sequences have the same length), only the initial part (a particular percentage in length) of the time-domain sequences are considered in the distance comparisons. For instance, if 5% of the data will be used, only the first 375 samples (i.e., the first 15 seconds) of the signals will be used in calculating DTW distances since the original time-domain signals are all 5 min or 7,500 samples long.

On the other hand, when feature vectors are used, only the first l feature vectors of each subject and activity are considered, where $l = \left\lceil 60 \frac{p}{100} \right\rceil$ with p being the percentage of data to be used.

If some subset of data will be used as explained above, all the formulas given in Section 3.5.1.1 need to be modified slightly: If time-domain signals are used, only the index n needs to be restricted to crop the signal. If feature vectors are used, the index k will change from 1 to the number of segments l to be used (instead of from 1 to N_k).

3.5.1.3 Results

The results are summarized in Figure 3.2. It is observed that when the time-domain signals normalized between -1 and 1 are compared by using the absolute and Euclidean distance measures, subject 1 is identified as the best subject [Figure 3.2(a) and (b)]. When they are compared by using the DTW distance measure, subject 2 is identified as the best [Figure 3.2(c)]. On the other hand, when the feature vectors normalized between -1 and 1 are used, subjects 3 and 5 are identified as the best subjects [Figure 3.2(d)–(f)]. The effect of normalizing the data is also observed in the figure. Normalization decreases the average distance between the data of one subject with the others by a factor of about 5,000. This

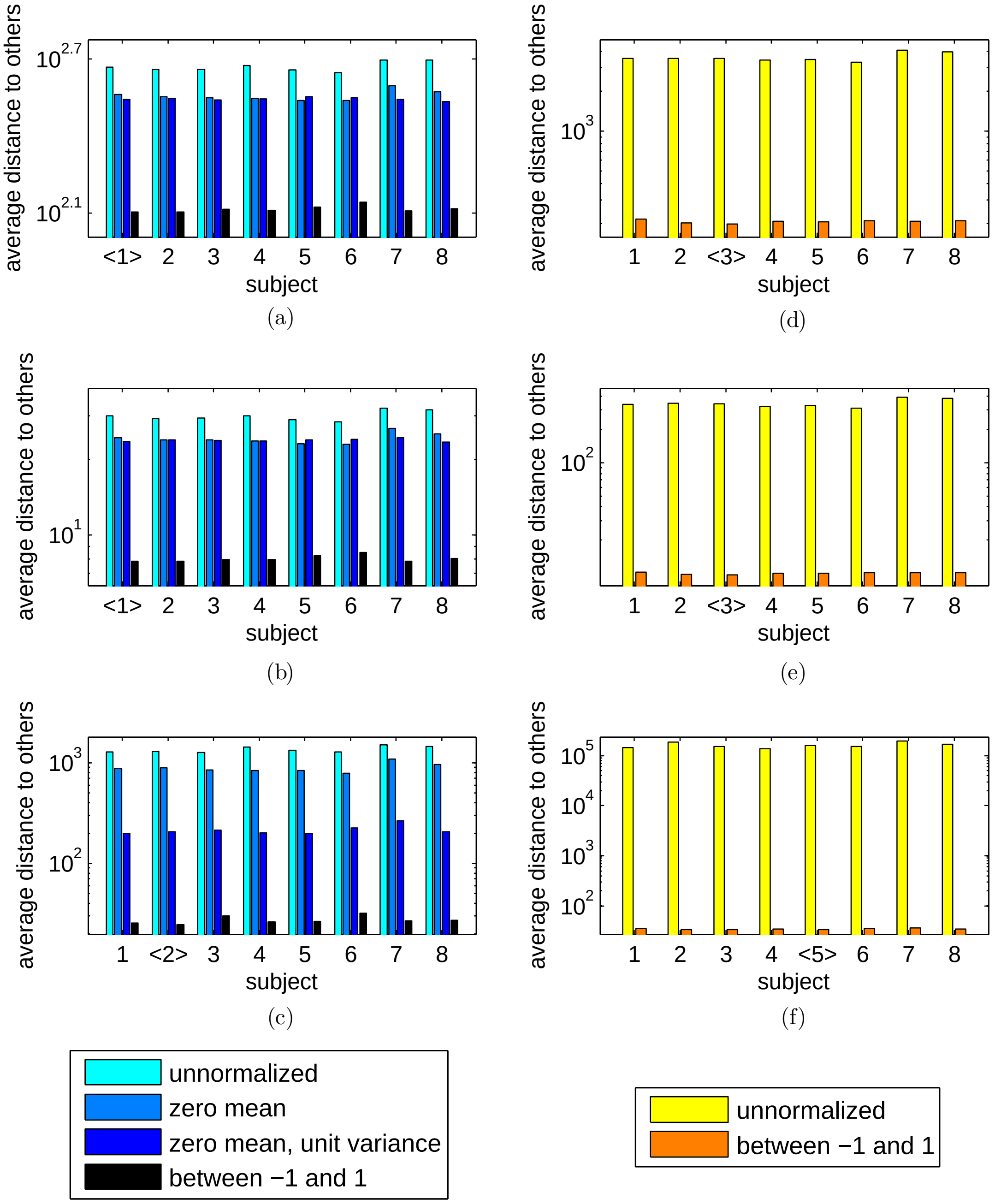


Figure 3.2: Average distance of each subject to the others in terms of the (a), (d): absolute, (b), (e): Euclidean, and (c), (f): DTW distances. Raw and normalized (a)–(c): time-domain signals, (d)–(f): feature vectors are used. The subject number with the smallest distance (for the signals normalized between −1 and 1) is enclosed in brackets.

is more visible when the DTW distance is used because for the DTW algorithm to wrap the two signals correctly, they must be of the same scale. When the mean values and amplitudes of the signals differ too much, DTW cannot match their similar parts because DTW cannot scale or shift the signal's amplitude values, it only wraps their time (or sample) axes. For this reason, the DTW distance measure applied to the raw signals is not expected to provide accurate results, but it is still shown in the figure for completeness.

3.5.2 Average Inter-Subject Distance per Activity

In this section, the distances between all distinct subject pairs are calculated and averaged out for each activity. The three distance functions are applied to both raw and normalized time-domain signals. That is, the average inter-subject distance for the activity a in terms of the distance measure $\mathfrak{D}$ is

$$d_{\text{avg-activity}, \mathfrak{D}}\left(a\right) = \frac{1}{N_p}\frac{1}{N_p - 1}\sum_{p_1}\sum_{p_2 \neq p_1}\left[\frac{1}{N_u}\frac{1}{N_s}\sum_{u}\sum_{s} d_{\mathfrak{D}}\left(x_{p_1,a,u,s}[n], x_{p_2,a,u,s}[n]\right)\right] \tag{3.17}$$

where the term in the square brackets is the average distance between the two subjects p_1 and p_2 for the activity a and is then averaged out for each distinct subject pair, resulting in the average inter-subject distance for the activity a.

The results are shown in Figure 3.3 in terms of the three distance measures applied to the raw and three types of normalized data. Average inter-subject distances of the zero-mean signals of activities A1–A4 are smaller than the other activities even though their unnormalized versions have larger distance values than some of the other activities. The reason is that these activities are completely stationary unlike the others; thus, the inter-subject differences in the signals are mostly caused by the bias and drift errors of the sensors. Hence, when the mean values are removed from the signals, they become very similar and the distance significantly decreases.

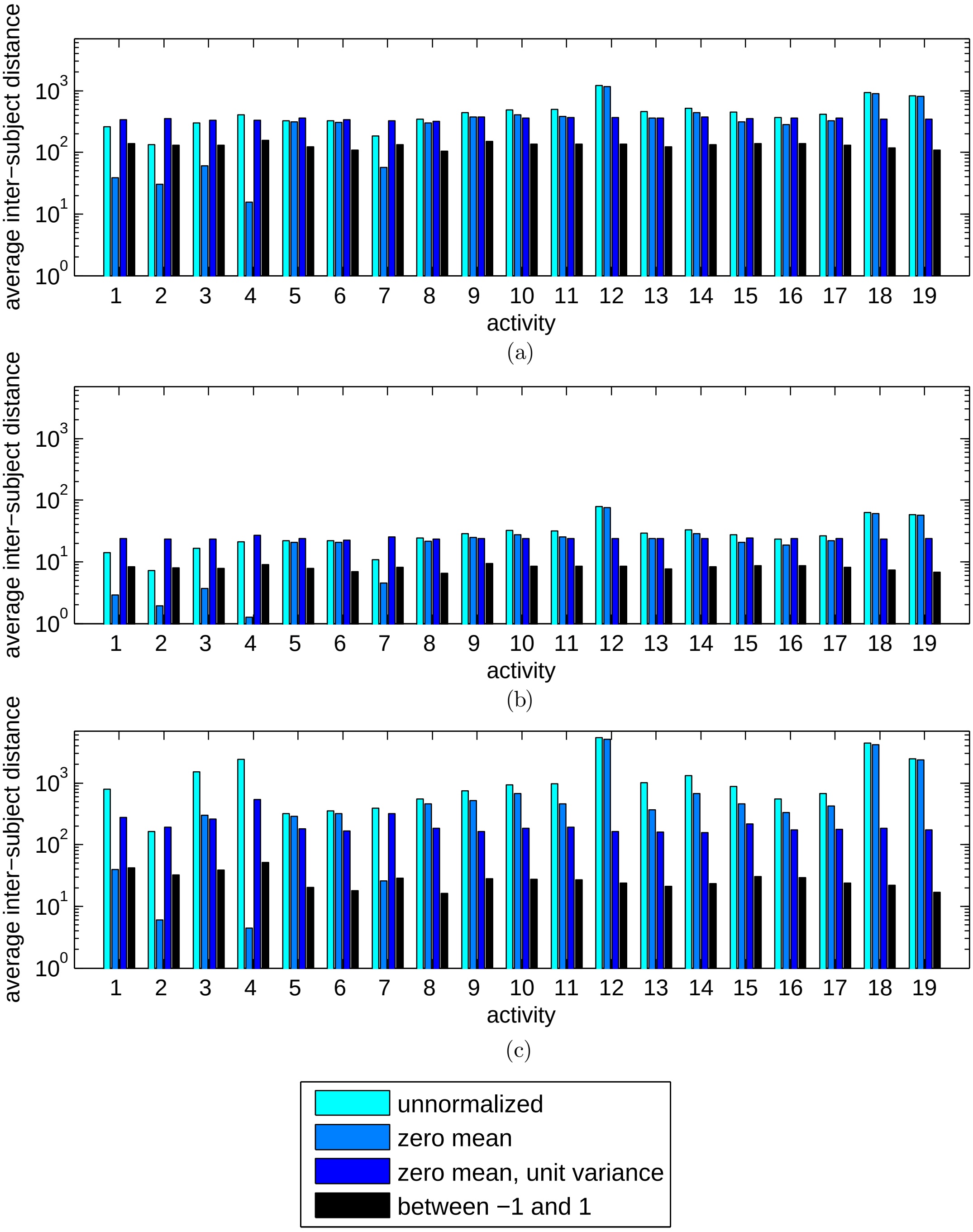


Figure 3.3: Average distance between all distinct subject pairs for each activity in terms of the (a) absolute, (b) Euclidean, and (c) DTW distances calculated using the raw and three types of normalized time-domain data.

In the figure, the distances of activities A18 and A19 (corresponding to jumping and playing basketball, respectively) are much larger than the other activities, because they have a random nature and thus cannot be compared well in the time-domain—it is better to use the statistical properties of the signals for comparison. Consequently, the distances of these activities are much larger than the others in the DTW case, because DTW cannot match these random activities of different subjects (even if it can match to some extent, it gives high penalty to big differences in time shifts, ending up with a large distance), while it can easily match quasi-periodic activities, leading to smaller distances, as expected.

3.5.3 Average Mean and Standard Deviation of Inter-Activity Distances for Each Subject, Unit, Sensor

In this section, inter-activity distances (i.e., distances between different activities) are calculated using normalized time-domain signals only, with subjects, units, and sensors kept the same in the distance calculations. Thus, all the distance values used in this section are calculated by

$$d_{\text{inter-activity, } \mathfrak{D}}\left(p, a_1, a_2, u, s\right) = d_{\mathfrak{D}}\left(\overline{x}_{p,a_1,u,s}[n], \overline{x}_{p,a_2,u,s}[n]\right), \tag{3.18}$$

where $\mathfrak{D}$ is the distance function, (a_1, a_2) is the activity pair, and p, u, s are the subject, unit, and sensor numbers, respectively. That is, the average distance between time-domain signals belonging to one activity and another are calculated over all subjects, units, and sensors.

The results will be separately summarized to compare subjects, units, and sensors because there are 61,560 distance values (8 subjects $\times$ 171 activity pairs $\times$ 5 units $\times$ 9 sensors) in total. (Note that there are normally 19×19 activity pairs, forming a 19×19 matrix. However, since distance functions are commutative, the matrix is symmetric. In addition, the diagonal elements are zero because the distance of a signal to itself is always zero. Hence, the essential part is only the upper-triangle with $\frac{19(19-1)}{2} = 171$ elements, corresponding to the distances of the 171 distinct activity pairs.)

The rows and columns of the 19 × 19 activity pair matrix $\mathbf{D}_{\text{activity pair, }p,u,s}$, of a particular subject p, a particular unit u, a particular sensor s are indexed by the different activities. Then, the (a_1, a_2)th element of the matrix is

$$\left(\mathbf{D}_{\text{activity pair, }p,u,s}\right)_{a_1,a_2} = d_{\text{inter-activity, }\mathfrak{D}}\left(p, a_1, a_2, u, s\right) \tag{3.19}$$

Note that, because of the redundancy explained above, only the upper-triangular part of the matrix needs to be taken into account in the calculations. Then, the mean and standard deviation of (the elements in) the upper-triangular part of the activity pair matrix can be calculated for each subject, sensor and unit. The mean is given by

$$\overline{D}_{\text{activity pair, }p,u,s} = \operatorname*{mean}_{a_1,a_2}\left(\text{upper-triangular part of }\mathbf{D}_{\text{activity pair, }p,u,s}\right) \tag{3.20}$$

and the expression for the standard deviation is

$$\widetilde{D}_{\text{activity pair, }p,u,s} = \operatorname*{std}_{a_1,a_2}\left(\text{upper-triangular part of }\mathbf{D}_{\text{activity pair, }p,u,s}\right). \tag{3.21}$$

3.5.3.1 Average Mean and Standard Deviation of Inter-Activity Distances per Subject

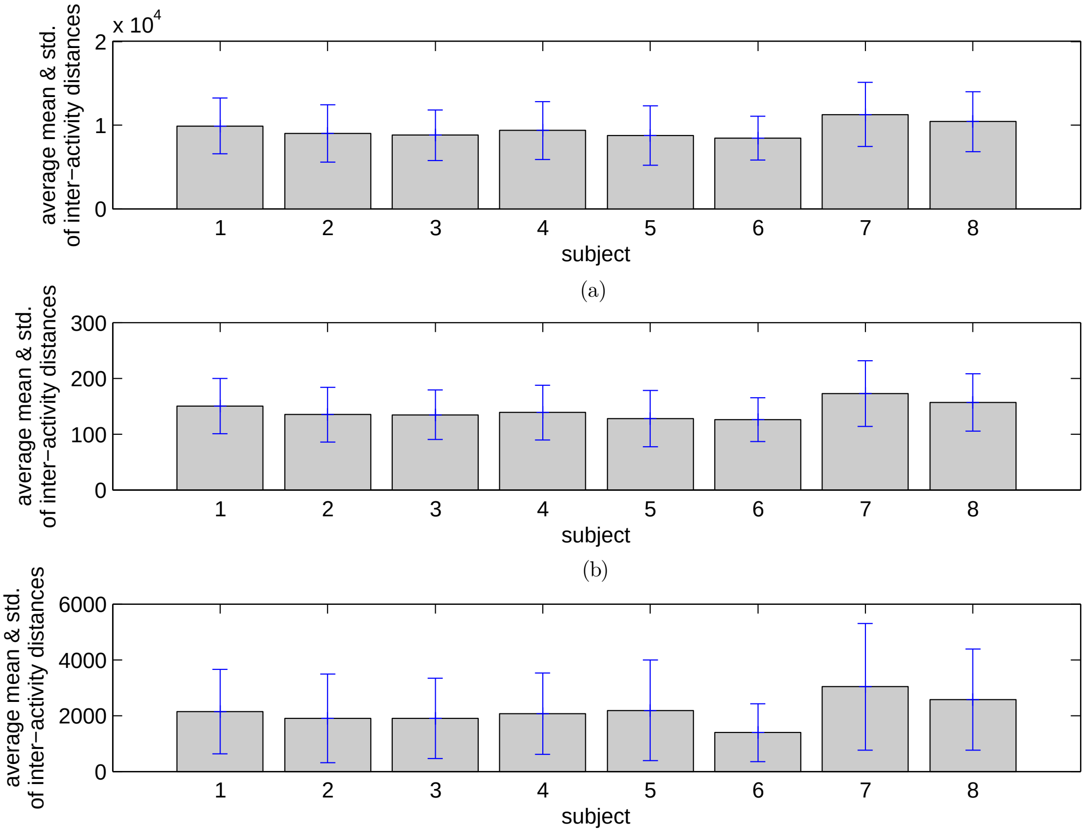
To summarize the results with respect to the subjects, the mean and the standard deviation values belonging to different sensors and units are averaged out for each subject. That is,

$$\overline{D}^{\text{avg., subject}}_{\text{activity pair, }p} = \operatorname*{mean}_{u,s}\left(\overline{D}_{\text{activity pair, }p,u,s}\right) \tag{3.22}$$

and

$$\widetilde{D}^{\text{avg., subject}}_{\text{activity pair, }p} = \operatorname*{mean}_{u,s}\left(\widetilde{D}_{\text{activity pair, }p,u,s}\right). \tag{3.23}$$

The results are shown in Figure 3.4. It is observed that the 6th subject has the smallest average inter-activity distance (i.e., the average distance between different activities). It is interesting to note that the 6th subject also performs



(a)

(b)

(c)

Figure 3.4: Average mean and standard deviation of inter-activity distances for each subject in terms of the (a) absolute, (b) Euclidean, and (c) DTW distances calculated using zero-mean time-domain data.

the activities in the “best” way on the average according to the description given in Section 3.5.1.

3.5.3.2 Average Mean and Standard Deviation of Inter-Activity Distances per Unit

To summarize the results with respect to the units, the mean and standard deviation values belonging to different subjects and sensors are averaged out for each unit. That is,

$$\overline{D}^{\text{avg., unit}}_{\text{activity pair, }u} = \operatorname*{mean}_{p,s}\left(\overline{D}_{\text{activity pair, }p,u,s}\right) \tag{3.24}$$

and

$$\widetilde{D}^{\text{avg., unit}}_{\text{activity pair, }u} = \operatorname*{mean}_{p,s}\left(\widetilde{D}_{\text{activity pair, }p,u,s}\right). \tag{3.25}$$

The results are shown in Figure 3.5. As expected, the sensor measurements vary the most in the units placed on the legs (RL and LL) where the acceleration can be large, and the least in the torso (T) unit.

3.5.3.3 Average Mean and Standard Deviation of Inter-Activity Distances per Sensor

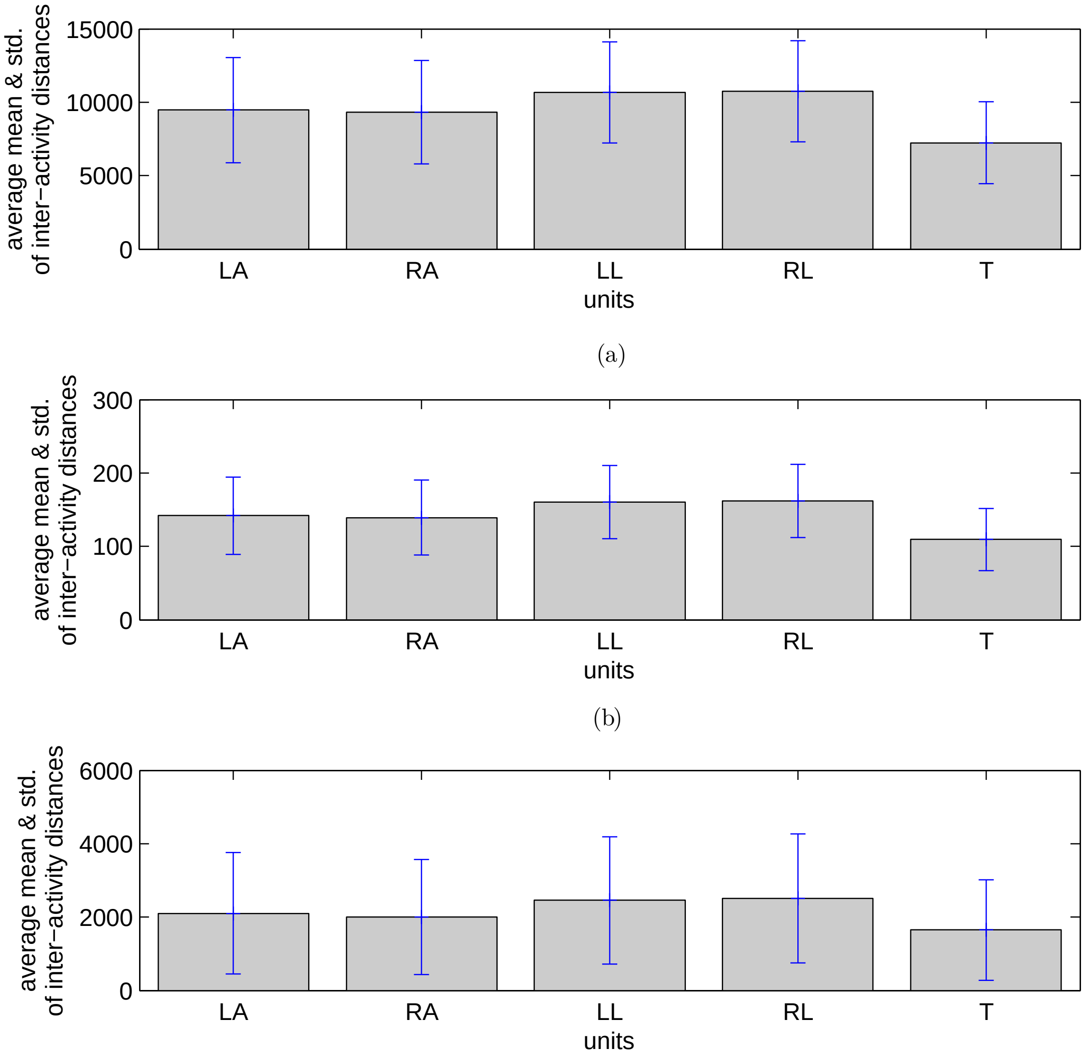
To summarize the results with respect to the sensors, the mean and standard deviation values belonging to different subjects and units are averaged out for each sensor. That is,

$$\overline{D}^{\text{avg., sensor}}_{\text{activity pair, }s} = \operatorname*{mean}_{p,u}\left(\overline{D}_{\text{activity pair, }p,u,s}\right) \tag{3.26}$$

and

$$\widetilde{D}^{\text{avg., sensor}}_{\text{activity pair, }s} = \operatorname*{mean}_{p,u}\left(\widetilde{D}_{\text{activity pair, }p,u,s}\right). \tag{3.27}$$

The results are shown in Figure 3.6. It is observed that there are differences up to the order of 100 in the average intra-class distances, because the measurement units, sensitivities, and the operating ranges of the three sensor types



(a)

(b)

(c)

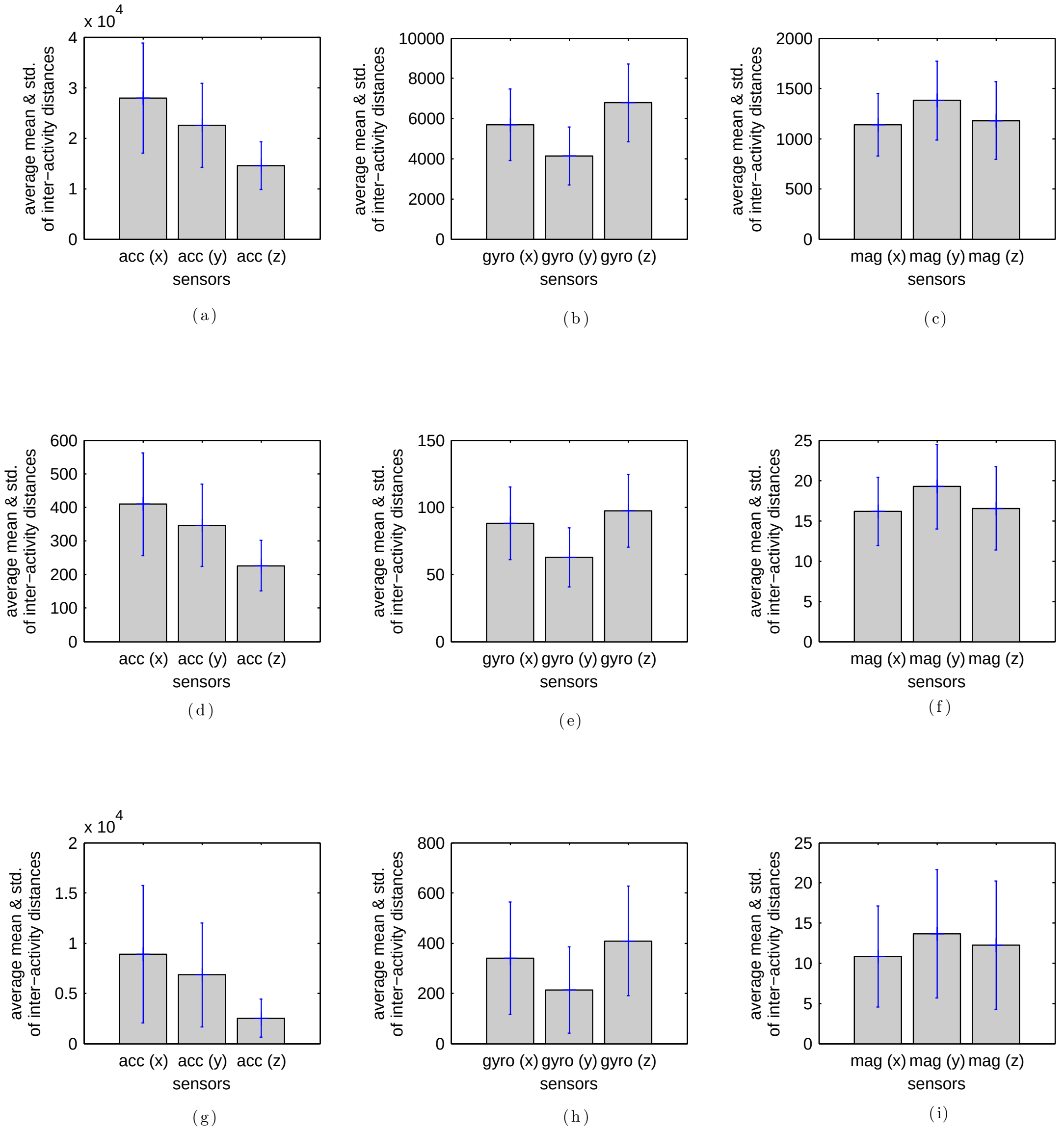
Figure 3.5: Average mean and standard deviation of inter-activity distances for each unit in terms of the (a) absolute, (b) Euclidean, and (c) DTW distances calculated using zero-mean time-domain data.

(accelerometer, gyroscope, and magnetometer) are different. In addition, the figure gives information about the axes of the sensors. For the accelerometers, the x-axis has greater difference among the activities, whereas the z- and y-axes of the gyroscopes and magnetometers are the most varying ones, respectively.

3.6 Discussion

It is observed that the type of normalization used significantly affects the results of the comparisons. For instance, the method proposed for identifying the best subject (see Section 3.5.1) results in different 'best' subjects for the three normalization types and the raw time-domain data. In addition, the average similarity of the data between different activities also differs for the three normalization types (see Section 3.5.3). For example, activity A8 (moving around) has the smallest inter-subject distance in terms of signals normalized between -1 and 1, whereas activity A4 (lying on right side) has the smallest inter-subject distance in terms of the zero mean signals for the absolute distance measure. These results also differ depending on whether the time-domain signals or their features are used. For instance, subjects 2 and 5 are identified as the 'best' subjects in terms of the DTW distance measure applied to the time-domain signals and feature vectors, respectively, both normalized between -1 and 1. The use of time-domain signals can be considered to be more suitable for quasi-periodic or stationary activities such as sitting (A1) or ascending stairs (A5), and the feature vectors are suitable for those activities with random elements such as moving around (A8) or playing basketball (A19). Although the average distance values that provide information about the similarity between the signals are very different in terms of the three distance measures, the sorting of the distance values rarely changes, indicating that the distance measures do not alter the comparison results as much as the normalization type.

The data of the standing activity (A2) varies the least between the subjects in the aforementioned dataset because this activity is quite stationary, the body posture is mostly the same in all subjects, and the anthropometry of the subjects



(a) (b) (c) (d) (e) (f) (g) (h) (i)

Figure 3.6: Average mean and standard deviation of inter-activity distances for each sensor in terms of the (a), (d), (g): absolute, (b), (e), (h): Euclidean, and (c), (f), (i): DTW distances calculated using zero-mean time-domain data. The sensor numbers are 1–9 from left to right in the whole figure.

does not affect the sensor signals significantly (see Section 3.5.2). On the contrary, relatively random activities such as jumping (A18) and playing basketball (A19) have much greater inter-subject distance because the signals in the time domain are not very suitable for comparing random activities. This fact supports the argument that the random signals are compared better in terms of their features. Another reason is that the sub-activities in these random activities such as dribbling in playing basketball do not happen at the same time instants in the different subjects' experiments, hence yielding a large distance value in comparing them in the time domain. Although the DTW distance tries to match them to obtain the highest similarity, it gives penalty to the unmatched subsequences and large amounts of warping, still resulting in a relatively large distance.

The average distance between the activities of subject 6 is the smallest among all the eight subjects, whereas that distance is the largest for subject 7 (see Section 3.5.3.1). This shows that subject 7 performs the activities in a more exaggerated way; i.e., with a larger amplitude, whereas subject 6 does the opposite. The inter-activity distances of the sensor units on the legs are the largest, whereas the torso unit has the smallest variation among the activities. The obvious reason is that the legs move much more than the arms and the torso in the activities of the aforementioned dataset (see Section 3.5.3.2). The inter-activity variations of the three axes of each of the three sensors are also quite different (see Section 3.5.3.3). For example, the z-axes of the accelerometers do not vary much when compared to the x-axes. However, different sensors cannot be compared with each other because of their different scales and measurement units. For instance, it is not reasonable to claim that the gyroscope signals vary less than the accelerometer signals from the provided results.

Note that, although the proposed method to identify the best subject seems to evaluate the performances of the subjects, it may not always be the case because physical attributes of subjects and their personal styles in performing the activities significantly affect the results. In other words, these results mainly show how similar each subject performs the activities compared to all the others.

3.7 Conclusion

In this part of the thesis, inter-activity and inter-subject distances are investigated based on the dataset that our research group has acquired in [35]. Distances between the signals of distinct subjects in the dataset are presented by averaging out for each activity and subject. A description of the 'best' subject is provided and the best subject is identified according to the inter-subject distances. Absolute, Euclidean, and DTW distance measures are used comparatively. The calculations are repeated for raw signals, zero-mean signals, and feature vectors. The effects of the three distance measures and removing the mean values are discussed. Moreover, the inter-activity distances are presented by averaging out for each subject, unit, and sensor.

Measuring the similarity between two or more subjects in a particular activity may be useful in a setting where a subject is training the others to perform some activity, such as in sports, dance figures, teaching to use a tool or an instrument, or teaching rehabilitation exercises to a patient. The trainer performs the activity properly but during the learning stage, the trainees will frequently deviate from the proper motion. The approach presented here could be used as a measure of the errors or deviations of the trainees during the learning process. In particular, in post-stroke rehabilitation, DTW is applied to the measurements obtained from strain sensors placed on the upper limb in order to measure the accuracy of the movements [41]. However, since physical attributes of the subjects may significantly affect the results, the trainees can be forced to perform the correct movements once if possible, and those measurements can be taken as the reference. In this way, during the training session, the measurements of the trainee will be compared with his/her own reference data, and the similarity (opposite of distance in this method) will be given as a feedback so as to improve his/her performance. In the next chapter, we investigate this problem and apply one of the distance measures provided here to physical therapy exercises.

Chapter 4

Automated Evaluation of Physical Therapy Exercises Using Multi-Template Dynamic Time-Warping on Wearable Sensor Signals

Physical therapy is an important type of rehabilitation in the treatment of various disorders. It usually requires exercising in a hospital or a rehabilitation center under the supervision of a specialist [61]. In many situations, after learning how to do the exercise movements correctly, the patients need to perform the exercises at home because they may not be able to go to the hospital frequently, they may not want to be in the hospital environment, or the physical or personnel capacity of the hospital may not allow them to do so [41]. Even if the patients are able to undergo physical therapy sessions in the hospital, the specialists cannot follow each patient continuously during their exercise sessions. This is because the specialists often alternate between at least several patients or they may have other tasks to do in between, resulting in poor feedback [42]. Moreover, different specialists often provide different feedback to the patients due to their subjective evaluation and the lack of systematic rules and guidelines for performing exercises [43]. For instance, some specialists may allow a larger amount of deviation from the ideal movement the patient needs to execute. Therefore, the problem is not only the lack of feedback in at-home physical therapy, but also the lack of an objective and accurate feedback mechanism in physical therapy even if performed under the supervision of a specialist [41].

Another aspect in physical therapy is the amount of exercise the patient performs. The "amount" is often measured in terms of the number of executions of an exercise, or worse, the duration of the exercise session, both of which may be misleading. In the former, the patient may perform the exercises in lower amplitude or in a quicker way resulting in less effective therapy. In the latter, different patients usually perform different number of executions in the same duration. In the past studies, to obtain more accurate information about the effectiveness of an exercise session, the intensity is estimated based on the duration of the active time [62] or the energy expenditure [63] of the patient. Both of these methods fail if the patient performs the exercises incorrectly. The exercises need to be evaluated objectively to assess the effectiveness of the session. Furthermore, it is also very difficult for a specialist to estimate the therapy intensity of a patient because s/he needs to count the number of correct executions of the exercises, which is difficult even when s/he is responsible from only a single patient and impossible when there are several patients monitored by the same specialist. Hence, accurate estimation of the intensity of an exercise session is also an important problem.

To solve the aforementioned problems, an autonomous system is developed that detects all the executions of one or multiple exercise(s) in an exercise session, evaluates each execution as correct or incorrect, and classifies the type of error if there is any, based on one of the comparison methods of different inertial sensor and accelerometer signals presented in Chapter 3. The system also quantifies the similarity between each execution and the ideal execution using a modified version of the DTW distance measure used in Chapter 3, and outputs how good the patient performs. For this reason, small and lightweight sensor units that contain inertial sensors and magnetometers need to be worn by the patient during the therapy sessions. The sensors are inexpensive and can be easily carried and worn by the patient, allowing at-home rehabilitation. For the system to detect the correct and incorrect executions of the exercise movements and the error types, the patient first executes the exercises in the correct way and in two different incorrect ways under the supervision of a specialist. Then, the system compares the detected executions with the supervised recordings to evaluate them. Once individual executions are detected and evaluated, statistical information including but not limited to the total number of executions the patient performs, the number of correct executions, the accuracy of the executions, and the active and the idle intervals in the physical therapy session can easily be determined. This can be used as feedback for the patient as well as the specialist, to whom the results can be sent remotely.

The most important advantage of the proposed methodology is that the patient does not need to push a button before each execution, or even select the exercise that s/he will perform. This makes the system usable in a physical therapy session of any duration, theoretically consisting of unlimited number of executions. In the experiments, it is observed that this autonomous methodology causes rare misdetections (MD) and false alarms (FA), which are tolerable considering that the system is much easier to use compared to the systems in other studies such as [64]. In addition, since the system proposed in this thesis does *not* use information on the sensor types, the number of sensors and units, and the sensor placement on the body, any sensor configuration that captures the movements sufficiently can be directly used with this algorithm without any modifications or adjustments. This is a significant advantage over many previous studies employing a 3-D human body model to evaluate the exercises, because in these studies such as [42, 65–67], the types, positions, and orientations of the sensors on the body are previously determined—the system needs to be modified considerably in case of a change in the sensor configuration. In addition, these types of systems often require separate rule-based methods to evaluate the correctness of the exercise executions, which makes them extremely difficult to configure for newly added exercise types. However, in our system, if a new exercise needs to be added, the only requirement is to record the templates of the different execution types of that exercise performed by the patient. This can be easily done by a physiotherapist, not an engineer who develops the system. Considering that there are numerous physical therapy exercises for different disorders, the "unsupervised" architecture of the algorithm allows flexibility in hardware type, hardware configuration, and exercise type.

The proposed methodology can be applied to all the patients who are assigned one or more exercise(s). Cardiopulmonary, neurology, orthopedics, and pediatry are the most common areas where rehabilitation treatment is used [68]. In particular, about the 110,000 people who experience a stroke, more than 75% require rehabilitation in the whole of U.K. [67]. Hence, there are many patients who can benefit from such a system.

The subsequence DTW algorithm (see Section A.3) applies the DTW to the template signal and the best-matching subsequence of the test signal. In order to search for different templates in a continuous and long test signal while allowing some flexibility, a novel algorithm, namely *multi-template multi-match dynamic time-warping (MTMM-DTW)* is developed as an extension of the subsequence algorithm. The algorithm allows template signals of different durations, and a test signal of any duration. It is highly adaptable to be used in different schemes because how much flexibility it will allow in the signals, how aggressive it will be in detecting the occurrences, how much overlapping it will allow in between the occurrences, and how short the matched subsequences may be are all adjustable as desired. Moreover, the algorithm only uses the two outputs (the distance and sample indices of the matched subsequence) of the subsequence DTW algorithm; thus, MTMM-DTW can be executed with any modifications and variations in the subsequence DTW, allowing even more flexibility.

This chapter is organized as follows: In Section 4.1, the related work is summarized. In Section 4.2, the extensions of the standard DTW algorithm (provided in Appendix A) including the MTMM-DTW algorithm are explained. Experiments and their results are presented in Section 4.3. Finally, conclusions are drawn in Section 4.4.

4.1 Related Work

Several different sensor technologies are used in physical rehabilitation, including inertial [42, 43, 61, 62, 64–66, 69–79], visual [65, 66, 70, 71, 80], strain [41, 45] and medical [43] sensors. However, many studies are based on estimating the activity/therapy intensity [61, 62, 74] or the energy expenditure [63] using the sensors rather than determining the accuracy of the physical therapy exercises. In numerous studies, a 3-D real-time human body model is built in order to observe the movements [42,65–67,74,80,81]. A major purpose of monitoring the patients' body movements is that the patients are aimed to do the given exercises to complete necessary tasks in video-game-like virtual environments, making exercise sessions more enjoyable [67, 76].

Another approach is *biofeedback*, which helps the patient and the specialist better observe the patient's body [82]. Biofeedback devices transform the sensor measurements of the body to hearable sound, a blinking LED, or an observable shape on the screen [83]. For example, for muscle strengthening or relaxation, electromyography signals measuring muscle tension are used [84]. Although there are portable solutions, most biofeedback devices are immobile and costly and are mostly used in hospitals or rehabilitation centers [44]. In addition, most devices do not evaluate the performance of the patient, or evaluate the results using a simple threshold, hence require the evaluation of the feedback by a specialist or sometimes the patient himself [44], both of which can be highly subjective. Therefore, biofeedback devices cannot take place of the specialist in most situations. In addition, since they provide the result of the performed action as feedback, the patient may not know what to do in order to obtain the desired result. In particular, biofeedback devices for muscle strengthening or relaxation do not guide the patients to perform a recommended movement or evaluate the movement itself; instead, they output the state of the muscle. Hence, biofeedback devices cannot be used to monitor/evaluate the accuracy or the amount of exercises the patient does.

Here, we provide a summary of those studies aiming to estimate the accuracy of physical therapy exercises or classify them as correct/incorrect:

Fergus et al. [42] developed a system that collects and stores the motion data of the patient, utilizing body area and sensor networks including inertial sensors. The system shows the body motions on a 3-D human body model either in real-time or using the stored data. The proposed approach for telerehabilitation is that the physiotherapists monitor the body motions remotely to measure the patient's progress [42]. However, the proposed solution is impractical and does not significantly improve the inspection time since the system itself does not provide any information about movement capability, movement accuracy or the patient's progress [42].

Taylor et al. [73] built a classifier that labels incorrect exercises prescribed for knee osteoarthritis, a degenerative disease of the knee joint, using five body-worn tri-axial accelerometers. Three exercises are performed by nine healthy subjects rather than patients who have the disease. The exercises are performed in the correct way as well as with a particular error such as "performing fast" or "knee not fully extended." The errors are different for each exercise and are mostly labeled by non-experts [73]. Several features extracted from the accelerometer data are used in the AdaBoost classifier to classify the exercises as being correct or having a particular error. Classification results are presented by using within-subject, across subjects, and subject-based L1O cross-validation techniques. However, multiple errors are not allowed by the methodology used and the classification accuracy is about 70% in most cases, which is not very good.

In [78], an Android application estimating the accuracy (i.e., score) of balancing board exercises using the internal accelerometer and magnetometer of a smartphone is developed. In this project, a complex rule-based algorithm is proposed to obtain a score value closest to the score given by an expert and the difference between the human and automatic scores is less than 10 points in more than 75% of the exercises on a 0 to 100 scale [78]. However, the proposed methodology does not yield an optimal solution, and different rule-based scoring algorithms are used for different exercise types.

In myHeart neurological rehabilitation concept [45], the accuracy of the arm movements in physical therapy are determined by using strain sensors. Healthy subjects wearing garments with printed strain sensors imitated how post-stroke patients perform both correct and incorrect movements for each of the seven exercise types under the supervision of a doctor. Open-end variant of the DTW algorithm is used to compute the similarity between the recorded signal and prerecorded correct template. The system decides that the activity is performed correctly if the similarity is greater than a threshold, which can be determined individually for each subject [78]. During the exercise, the system continuously compares the measured signals with the first part of the template and gives real-time feedback to the patient about the movement accuracy. The classification accuracy is 85% on average. The disadvantage of the system is the difficulty of wearing the garment—since the garment must be tight-fitting, it would usually be very difficult for a post-stroke patient to wear, especially when compared to the inertial sensor units that can be worn as bracelets or other accessories.

In [41], strain sensors worn on the arm are used again to provide real-time feedback to neurological patients undergoing motor rehabilitation. Seven exercises are executed by a healthy subject wearing a left-handed sensorized long-sleeve shirt both correctly and incorrectly at various speeds. The system checks whether the measured signals "match *at most once* a prefix of one of several stored references, used as templates" [41] in order to detect which activity is performed. Then, the classifier selects a class among the correct class (including different execution speeds) and the incorrect classes "movement too small," "typical compensatory action 1," and "typical compensatory action 2." The dissimilarities between the strain signals are measured as DTW distances using the open-end DTW algorithm and they are fed to a 1-NN classifier both in exercise recognition and correctness/error type classification. The disadvantage of the proposed system is again the difficulty of wearing the sensorized shirt.

m-Physio platform [64] classifies the physical rehabilitation activities as correct/incorrect using accelerometers. In m-Physio, the smartphone iPhone containing a tri-axial accelerometer is mounted on the patient's leg or arm depending on the exercise he performs. The accuracy of the exercises is determined by the ordinary DTW algorithm applied to the accelerometer signals to compare the exercise performed with the pre-recorded template. Four steps are involved in the m-Physio platform [64]:

- *exercise capture:* The patient performs the rehabilitation exercises correctly under the supervision of a specialist and the accelerometer signals of (one instance of) each exercise are stored as templates. The specialist also determines four parameters for each exercise type: the movement's minimum and maximum exercise duration, the sampling frequency of the accelerometer, and the amount of smoothing applied to the measured signals.

- *exercise training:* The patient performs the exercises under the supervision of a specialist several times and the system provides feedback as *correct/incorrect/too short in time/too long in time* on the iPhone screen for each repetition. The specialist checks whether the patient performs the exercises accurately and whether the system provides accurate feedback. S/he returns to the previous step to revise the parameters or record a new template if necessary.

- *personal rehabilitation:* In this step, the patient performs the exercises without the need for a specialist and the system provides feedback to the patient on the iPhone screen as *correct/incorrect/too short in time/too long in time.* This way, the patient is able to learn if he is performing the movements correctly. When he performs incorrectly, he has an opportunity to improve during the training session. In addition, the system records and uploads the patient's statistics to a centralized database.

- *web application:* The specialist remotely checks the patient's status using the web interface connected to the centralized database and contacts him/her when necessary.

However, before and after the patient performs each execution of an exercise, he has to touch the iPhone screen to mark the start and the end of each exercise, which is not practical. This is not only necessary to determine the movement duration, but also because the similarity measure used to compare the signals

(DTW) penalizes the additional unmatched parts in the beginning and at the end of the signals. Another disadvantage of the m-Physio platform is the need to determine the aforementioned four parameters in the exercise capture phase. The specialist determines them by trial and error, which may easily alter the accuracy of the system.

In summary, in the previous studies, either the executions of the exercises are cropped manually, the subject marks each execution by pressing a button, or the subject performs each execution when s/he is informed by the system by a sound or on-screen notification. In addition, no idle time periods are involved in the studies evaluating the executions. There exist studies that estimate the active duration of the patient in an exercise session (for example, [62, 63]), but they neither detect the executions of activities nor evaluate them; they simply estimate the intensity of the session. On the other hand, once the exercise movements are recorded, the system proposed in this thesis automatically detects the movements as well as the idle time periods, if there are any, during an exercise session, independent of the number of exercise types. The system also classifies each movement as one of the exercise types and evaluates it, indicating the error type if there is any. The patient neither needs to press a button in the beginning and the end of each execution nor select the exercise he is going to perform. A physiotherapist is needed only while recording the movements in order to make sure that the patient performs the exercise correctly or with a predetermined error. Then, the patient can perform the exercises anywhere provided that he properly wears the sensors, and can observe how well he performs. Since the system also counts the executions, it can be used to notify the patient when he completes the advised number of repetitions in a given time interval. The results may also be checked by an expert to observe the patient's progress.

4.2 Modifications to the DTW Algorithm

In this chapter, we propose to use a modified version of the DTW algorithm, namely *multi-template multi-match DTW (MTMM-DTW)*, which is suitable for detecting the occurrences of multiple exercise templates in a recorded signal. This makes it possible to identify correct and incorrect executions of an exercise (including two commonly occurring types of error), the counting of the exercises, and their classification over 8 most commonly assigned arm and leg exercises, which are selected by consulting a physiotherapist at the Gülhane Military Medical Academy, Turkish Armed Forces Rehabilitation Centre [85].

For the detection of multiple occurrences of multiple templates, an approach based on DTW is selected because the DTW algorithm (see Appendix A) is much more flexible than the absolute and Euclidean distance measures in comparing two signals since it tries to match the similar parts of the signals. This may be beneficial when the variation in different executions of the same physical therapy exercise is considered. The speeds or durations of some parts of the exercise movement may change, which should be tolerable. For instance, if the exercise contains a phase at which the patient waits for 5 s, the distance should not increase significantly when the patient waits for 4 or 6 s. On the other hand, at the same time, the distance measure should not tolerate differences in the amplitude, which occurs, for example, when the patient waits for 5 s in a different position. If the absolute or Euclidean distance measures are used, both variation types affect the distance value and it is not possible to allow one of them while penalizing the other. However, the DTW algorithm naturally compensates linear or nonlinear changes in the time (or sample) axis but not changes in amplitude, which is desired in this scheme.

Firstly, the *single-template multi-match DTW (STMM-DTW)* algorithm that is developed to detect possibly multiple occurrences of a template signal in a long signal is described. Then, the *MTMM-DTW* algorithm that is developed to detect possibly multiple occurrences of multiple template signals in a long signal is presented.

4.2.1 Single-Template Multi-Match DTW (STMM-DTW)

In the standard DTW and the subsequence DTW algorithms, summarized in Appendix A, the sequences $\mathbf{x}$ and $\mathbf{y}$ do not have to be of the same length. The standard DTW matches their first and last samples to each other, possibly warping the time (or sample) axes in between to obtain maximum overall similarity. See Figure 3.1(b) for an illustration. On the other hand, the subsequence DTW algorithm matches the subsequence of $\mathbf{y}$ that is the most similar to the template signal $\mathbf{x}$. Therefore, the subsequence DTW can be used to search for a segment that resembles the template in a much longer test signal. It may also be desirable to detect *all* of the subsequences in $\mathbf{y}$ similar to $\mathbf{x}$ in a setting where the template signal may occur more than once in the test signal. The STMM-DTW algorithm serves this purpose and is provided below. It is similar to the "Compute Similar Subsequences" algorithm in reference [86].

Algorithm 1 single-template multi-match DTW (STMM-DTW)

```
$N \leftarrow \text{length}(\mathbf{x})$
$M \leftarrow \text{length}(\mathbf{y})$
$M_{\text{left}} \leftarrow M$ {$M_{\text{left}}$ is the maximum number of successive unmatched samples in $\mathbf{y}$}
while $M_{\text{left}} \geq N/\tau$ {$\tau$ is the threshold factor} do
    Compute $\text{DTW}_{\text{subsequence}}(\mathbf{x}, \mathbf{y})$ and save $\text{DTW}_{\text{subsequence}}$, $m_1$, and $m_2$
    $\mathbf{y}(m_1 : m_2) \leftarrow \infty$ {the $\infty$-valued samples can never be used in the next DTW executions}
    if $m_2 - m_1 + 1 \geq N/\tau$ {the last matched subsequence is sufficiently long} then
        Add the last match to the list.
    else
        Ignore the last match. {the last matched subsequence is too short}
    end if
    $M_{\text{left}} \leftarrow$ the maximum number of successive finite samples in $\mathbf{y}$
end while
```

In the STMM-DTW algorithm, the subsequence DTW algorithm is executed multiple times to obtain multiple subsequences. However, a restriction can be imposed on the matches: The length of each matched subsequence must be at least

N/τ, where N is the length of $\mathbf{x}$ and τ is the *threshold factor*. (This restriction may be omitted by setting $\tau = 0$.)

The algorithm first executes subsequence DTW and checks whether the matched subsequence satisfies the length condition. If so, the results are added to the list of matched subsequences and the values of the matched subsequence of $\mathbf{y}$ are set to infinity in line 6 to prevent this part from matching again in the subsequence DTW executions that follow. This procedure is repeated until the maximum number of successive samples in $\mathbf{y}$ that were not matched goes below M/τ, in which case the next matched subsequences will not satisfy the length condition.

In the STMM-DTW algorithm, the matched subsequences are not allowed to overlap with each other. This restriction may be loosened to allow some overlap by replacing line 6 by $\mathbf{y}(\tilde{m_1} : \tilde{m_2}) \leftarrow \infty$ where $\tilde{m_1} = (1-\sigma)m_1 + \sigma m_2$ and $\tilde{m_2} = \sigma m_1 + (1-\sigma)m_2$ with $\sigma \in (0, 1]$ being the ratio of the matched subsequences that are allowed to overlap in the beginning and at the end with another subsequence. If $\sigma = 1$, overlapping is not allowed.

The advantages of the STMM-DTW algorithm are that (1) the number of subsequences, their locations on the sample axis, and the length of the test signal $\mathbf{y}$ need not be known, (2) the template and test signals $\mathbf{x}$ and $\mathbf{y}$ may be multidimensional, (3) trivial false matches with a subsequence of length much smaller than the template signal may be ignored by setting the threshold factor τ, and (4) the amount of overlapping between the matched subsequences can be adjusted as desired. Hence, the STMM-DTW algorithm can be used to determine the busy or idle time periods (in the sense that the known template occurs or not), to estimate the number of occurrences of the template in the test signal, and to determine the time instants and the durations of all the occurrences of the template.

4.2.2 Multi-Template Multi-Match DTW (MTMM-DTW)

The STMM-DTW detects possibly multiple occurrences of a single template signal $\mathbf{x}$ in the test signal $\mathbf{y}$. However, in some applications, it may be necessary to search for a number (K) of different template signals $\mathbf{x}^{(1)}$, $\mathbf{x}^{(2)}$, $\ldots$, $\mathbf{x}^{(K)}$ in $\mathbf{y}$. To this end, the *MTMM-DTW* algorithm is newly proposed in this study.

Algorithm 2 multi-template multi-match DTW (MTMM-DTW)

```
$M \leftarrow \text{length}(\mathbf{y})$
for $k = 1 \rightarrow K$ do
    $M_{\text{left}}(k) \leftarrow M$
        {$M_{\text{left}}$ is the maximum number of successive unmatched samples in $\mathbf{y}$ for the $k$th template $\mathbf{x}^{(k)}$}
    $\tilde{\mathbf{y}}^{(k)} \leftarrow \mathbf{y}$ {the test signal $\mathbf{y}$ replicated as $\tilde{\mathbf{y}}^{(k)}$ to be used for each template $\mathbf{x}^{(k)}$}
end for
while $M_{\text{left}}(k) \geq M/\tau,\ \forall k$ {$\tau$ is the threshold factor} do
    compute $\text{DTW}_{\text{subsequence}}\left(\mathbf{x}^{(k)}, \tilde{\mathbf{y}}^{(k)}\right)$ and save $\text{DTW}^{(k)}_{\text{subsequence}}$, $m_1^{(k)}$, and $m_2^{(k)}$ for $\forall k$
    $k^* = \arg\min_{k' \in \{1,\ldots K\}} \frac{1}{\text{length}\left(\mathbf{x}^{(k)}\right)} \text{DTW}^{(k')}_{\text{subsequence}}$
        {find matched subsequence of the template having the minimum DTW distance *per sample of the template*}
    if $m_2^{(k^*)} - m_1^{(k^*)} + 1 \geq M/\tau$ {the last matched subsequence is sufficiently long} then
        add the last match with template number $k^*$ to the list
        $\tilde{\mathbf{y}}^{(k)}\left(m_1^{(k^*)} : m_2^{(k^*)}\right) \leftarrow \infty,\ \forall k$
            {prevent the matched samples from being matched in the next DTW executions for all templates}
    else
        ignore the last match {the last matched subsequence is too short}
        $\tilde{\mathbf{y}}^{(k^*)}\left(m_1^{(k^*)} : m_2^{(k^*)}\right) \leftarrow \infty$
            {prevent the last matched subsequence from being matched to the same template in the next DTW executions}
    end if
    for $k = 1 \rightarrow K$ do
        $M_{\text{left}}(k) \leftarrow$ the maximum number of successive finite samples in $\mathbf{y}$ for the $k$th template $\mathbf{x}^{(k)}$
    end for
end while
```

The MTMM-DTW algorithm detects possibly multiple occurrences of all of the template signals in the test signal. Since the multiple templates can have very different durations and the DTW distance is the cumulative distance obtained by summing the pairwise costs between the samples of the warped template and the test signals, the DTW distances of the subsequences matched to different template signals should be normalized by the template lengths (i.e., the number of samples in the template signals) in order make a fair comparison. To this end, when the subsequence DTW algorithm is executed separately for each template (in line 7), the DTW distances of the subsequences matched to a particular template are divided by the length of that template (in line 8). Then, the subsequence with the minimum *normalized* DTW distance is selected, and checked if it satisfies the length condition: The matched subsequence length (in samples) must be at least the same as the length of the matching template (in samples) divided by the threshold factor τ, similar to the length criterion in the STMM-DTW algorithm. If the subsequence satisfies the length criterion, the subsequence's DTW distance, sample interval in the test signal and the matching template number are saved as an item in the list of matched subsequences. Then, the matched samples of the test signal are set to ∞ to prevent them from matching again to any template in the DTW executions that follow (in line 11). On the other hand, if the subsequence does not satisfy the length criterion, it is ignored (i.e., not saved to the list), and the samples in the test signal corresponding to this subsequence are set to ∞ *only for the template matched to this subsequence* in order to prevent the same subsequence from being matched to the same template in the DTW executions that follow. Otherwise, exactly the same subsequence is going to be matched to the same template in all of the following iterations since subsequence DTW always finds the best matching subsequence and does not contain any randomness. The sample range of the test signal is not the same for all of the template signals because if the subsequence does not satisfy the length criterion, there are three possibilities in the following DTW execution: another subsequence matching with the same template at a different location in the test signal may be found (1) satisfying or (2) not satisfying the length criterion, or (3) a subsequence matching with another template (at the same or different location) in the test signal may be found. Therefore, the last matched subsequence must be "invisible" to the template signal matching to it (without satisfying the length criterion) in order for it to be able to match to subsequences at different positions [cases (1) and (2)], but at the same time "visible" to other templates to allow them finding another match in the same position (since the previous match is ignored) [case (3)].

In the MTMM-DTW algorithm, the matched subsequences (associated either with the same template or different templates) in $\mathbf{y}$ cannot overlap with each other. Similar to the STMM-DTW algorithm, this restriction can be loosened by replacing the sample range $\left(m_1^{(k^*)} : m_2^{(k^*)}\right)$ in lines 11 and 14 by $\left(\tilde{m}_1^{(k^*)} : \tilde{m}_2^{(k^*)}\right)$ where $\tilde{m}_1^{(k^*)} = (1-\sigma)m_1^{(k^*)} + \sigma m_2^{(k^*)}$ and $\tilde{m}_2^{(k^*)} = \sigma m_1^{(k^*)} + (1-\sigma)m_2^{(k^*)}$ with $\sigma \in (0, 1]$, where σ is the ratio of the matched subsequences that are allowed to overlap in the beginning and at the end with other subsequences. Overlapping is not allowed if $\sigma = 1$.

The advantages of the MTMM-DTW algorithm are similar to the STMM-DTW: (1) The number of templates, the number of subsequences, their positions on the sample axis, and the length of the test signal $\mathbf{y}$ need not be known, (2) the template and the test signals $\mathbf{x}$ and $\mathbf{y}$ may be multi-dimensional, (3) trivial false matches with length much smaller than the matching template signal may be avoided by setting a threshold factor τ, (4) the amount of overlap between the matched subsequences can be adjusted as desired, and (5) the algorithm can detect all the occurrences of the templates in the test signal and classify the detected subsequences. With these properties, the algorithm can be used for several different purposes: classification of a signal given multiple template signals (pattern recognition), detecting the occurrences of all the templates in the test signal with their time instants and durations, estimating the number of repetitions of all the templates in the test signal, or all of them. Note that, knowing only the template signals and the test signal, MTMM-DTW automatically extracts the samples inside the test signal, forming the test dataset, and also classifies the detected subsequences. To reduce possible false alarms or misdetections, the local weights w_d, w_h, w_v (see Section A.2), the threshold factor τ (see algorithm 2), and the overlap ratio σ (explained in the preceding paragraph) can be adjusted specific to the application.

A similar algorithm is proposed in reference [87] that recognizes multiple templates in a lengthy signal. However, that algorithm assumes that the long signal consists of templates connected in a continuous manner, where there are neither idle periods nor unmatched intervals, and applies open-end DTW (see section A.3) sequentially to the long signal to detect the templates. Indeed, it divides the long signal into intervals where each interval is classified as one of the templates. Thus, it is not as flexible as the MTMM-DTW algorithm proposed here.

4.3 Experiments and Results

4.3.1 Physical Setup

Body-worn sensor units containing inertial sensors and magnetometers are used to capture the body motions. Five MTx units manufactured by Xsens Technologies [29] are fixed to different positions on the subject's body. Each unit contains a tri-axial accelerometer, a tri-axial gyroscope, and a tri-axial magnetometer. The ranges of the accelerometers in units 4 and 5 are $\pm 5g$, and in units 1–3 are $\pm 18g$, where $g = 9.807$ m/s^2 is the gravitational constant. The ranges of all the gyroscopes and magnetometers are ±1200°/s and ±75 µT, respectively.

Two different sensor configurations are used to capture leg and arm movements because the exercises considered in this thesis require only arm or only leg movements. In the first configuration [shown in Figure 4.1(a)], which is focused on leg movements, units 1 and 2 are placed on the outer sides of the left and right ankles and units 4 and 3 are placed on the outer sides of the left and right knees, respectively. Unit 5 is placed on the torso. The second configuration [shown in Figure 4.1(b)] is designed to capture right arm movements, where unit 1 is placed on the outer side of the right knee, unit 2 is placed on the wrist, unit 3 is placed on the inner side of the upper arm above the elbow, unit 4 is placed at the top of the right shoulder, and unit 5 is placed on the torso. Since these sensor configurations are designed to capture only the right arm and the right leg movements, they are not symmetrical on the human body. The system does not use the knowledge of sensor placement or sensor orientations and does not rely on rule-based algorithms; therefore, the sensor units can be placed anywhere on the body to properly capture the movements of specific exercises, provided that the configuration is the same in recording the templates and exercising. Moreover, additional units or sensors can be used, or some redundant units or sensors may be excluded if desired. For this reason, the proposed system is highly flexible and modular, especially when compared with the systems modeling the human body and evaluating the exercises based on rule-based algorithms.

All of the sensors are calibrated by using the default calibration procedure of the system and sampled at 25 Hz. Therefore, there are 9 sensors × 5 units = 45 discrete-time signals recorded in the experiments. The units of accelerometer and gyroscope signals are $\mathrm{m/s^2}$ and rad/s, respectively. The unit of the magnetometers is stated to be "arbitrary units normalized to earth field strength" by the manufacturer [29]. Hence, the magnetometer signals are expected to vary between -1 and 1 as long as there are no external magnetic field sources.

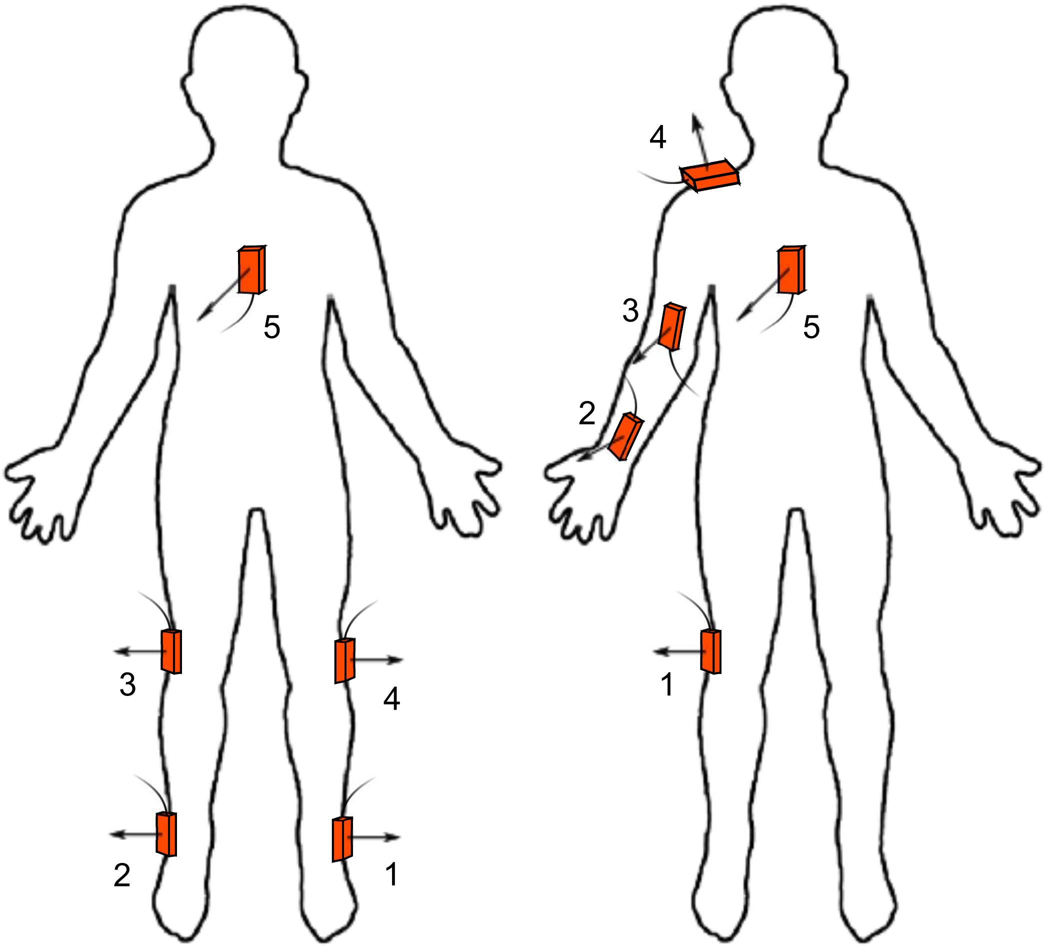
4.3.2 Exercises

The exercises considered in this thesis were suggested and approved by a medical doctor, namely physical therapy specialist Assoc. Prof. Dr. İlknur Tuğcu, at the Gülhane Military Medical Academy, Turkish Armed Forces Rehabilitation Centre [85]. They are the most commonly assigned exercises to patients, mostly for orthopedic rehabilitation. A brief description of each exercise is provided below:

1. while sitting on a high flat surface, raising the right leg, waiting for 5 s with the right knee kept straight, and returning to the initial position

2. while sitting upright on a stool and the arms hanging downwards, bending the upper body 30° to the front, waiting for 5 s, and returning to the initial position

3. while lying flat on the back on a flat surface, raising the right leg from the hip joint with the right knee and left leg kept straight, waiting for 5 s, and returning to the initial position



(a)

(b)

Figure 4.1: Sensor placement on the human body. (a) The first and (b) the second configuration that focus on the right leg and the right arm movements, respectively. The Xsens MTx sensor units are shown as boxes with the arrows and the cables being z and $-x$ direction of the sensors, respectively. The y axis can be found considering that right-handed coordinate systems are used.

4. while lying flat on the left side on a flat surface, raising the right leg from the hip joint with the right knee and left leg kept straight, waiting for 5 s, and returning to the initial position

5. while lying flat with the face downwards on a flat surface, raising the right leg from the hip joint with the right knee and the left leg kept straight, waiting for 5 s, and returning to the initial position

6. while sitting on a chair, keeping the right arm straight with the right hand being close to the right knee and the palm facing upwards, raising a 1-kg weight held in the right hand upwards from the elbow joint, waiting for 5 s, and returning to the initial position

7. while standing upright with the right arm kept straight and hanging downwards, raising a 1-kg weight held in the right hand to the right side from the shoulder joint while keeping the elbow joint straight, waiting for 5 s with the right arm being in a horizontal position, and returning to the initial position

8. while lying flat with the face downwards on a flat surface, the right arm being out of the surface in a horizontal position, with the right arm making an angle of 90° with the trunk and the right forearm hanging downwards, the elbow joint at 90°, raising the right forearm upwards to make the elbow straight, waiting for 5 s and returning to the initial position

4.3.3 Experiments

The experiments are designed to test whether the proposed MTMM-DTW algorithm is able to

1. detect the exercises within a long signal recorded in a typical exercise session, and hence determine the idle time periods and estimate the number of repetitions of the exercise(s),
2. classify each exercise type,
3. determine whether the exercise is performed properly or not, and classify the error type if there is any.

Two commonly occurring errors that patients make during exercise sessions are the following:

1. they perform the movements too fast that they do not wait for the required amount of time in a certain position in order to quickly complete the number of repetitions they need to perform in a day or during an exercise session, or
2. they perform the exercises in low amplitude; that is, they do not completely execute the movement. This may be caused by physical incapability (such as after a stroke) or by negligence, carelessness, etc.

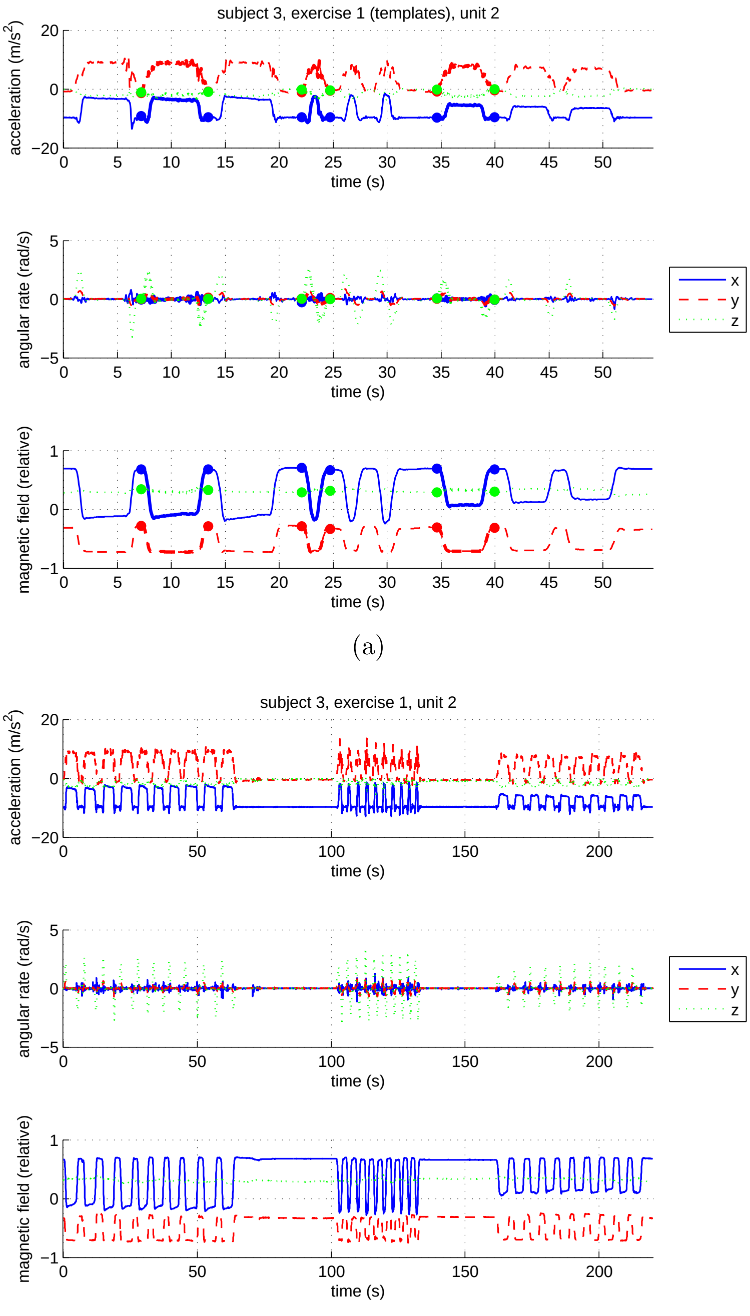
In the following, these will be referred as *type-1* and *type-2 error*s, respectively. Therefore, there are three execution types for each experiment: (1) correct, (2) with type-1 error, (3) with type-2 error.

Since the proposed system works by matching the subsequences in a long signal that are similar to the provided templates, all execution types of each exercise must be recorded (by the same subject to increase the accuracy) beforehand, which is called the *training phase*. To this end, each patient first performs all three execution types of each exercise he is assigned under the supervision of a specialist and the templates are recorded. Then, the system uses these templates to evaluate the exercises he performs without any supervision. If the patient is unable to execute the exercises in the training phase because his/her muscles have not sufficiently developed yet or if s/he feels too much pain, the specialist is able to apply external force to make the patient execute the exercises [85].

To evaluate the performance of the system completely, all of the occurrences of the exercise(s) in an experiment simulating an exercise session must be known including the time interval, the exercise type and the execution type (correct, type-1 error, type-2 error). However, noting down all the information in each experiment is cumbersome because then either this detailed information needs to be recorded in MATLAB to evaluate the accuracy, or the whole system needs to be evaluated manually. For this reason, the experiments are conducted in a more systematic fashion.

In an experiment, simulating a typical real-world exercise session, the subject repeats one of the assigned exercises *correctly* for 10 times, and then waits until the 100th second of the experiment. During the waiting time, the subject neither performs an exercise nor moves too much, hence is considered to be in the "idle" state. Starting at the 100th second, he repeats the same exercise for 10 times with type-1 error, and again waits idly, this time until the 160th second. Then, the subject executes the exercise 10 times with type-2 error, and the experiment ends without any more idle time periods. Therefore, an experiment in the test set consists of 10 executions of an exercise for each execution type in addition to the varying durations of the idle time intervals in between. In Figure 4.2, typical training and test recordings are illustrated. In Figure 4.2(a), the outputs of the sensors in unit 2 belonging to the templates for the three execution types are shown. Manually selected templates in a separate recording are highlighted in Figure 4.2(a). In Figure 4.2(b), the experiment for exercise 1 performed by the third subject is depicted. Two idle time periods are observed in the experiment shown in Figure 4.2(b).

The reason for the need of the subject being stationary in this time period is that the system searches a known movement (i.e., all three execution types of



(a)

(b)

Figure 4.2: Recording of the templates and the experiment for exercise 1 performed by subject 3. (a) The three templates (highlighted with thick lines) for correct, type-1 error, and type-2 error execution types of exercise 1, (b) the experiment consisting of 10 repetitions of exercise 1 for the three execution types and idle time periods in between. Only the sensor outputs of unit 2, which is the most important one in this exercise, are shown.

subject	gender	age	weight (kg)	height (cm)
1	female	55	73	169
2	male	61	85	180
3	male	23	95	180
4	female	48	55	158
5	male	53	98	175
average	60% male	48.0	81.2	172.4

Table 4.1: Physical properties of the subjects who performed the experiments.

every exercise) throughout the experiment since it also performs exercise classification.

For the experiments, each of the 5 subjects, whose physical properties are given in Table 4.1, performed the 8 experiments each corresponding to a different physical therapy exercise. Since there are totally $5 \times 8 = 40$ experiments, each containing 30 executions, the dataset consists of $5 \times 8 \times 30 = 1{,}200$ exercise executions as well as some idle time periods in between.

Note that, the dataset is not perfect in the sense that all the executions may not strictly belong to a particular execution type. For example, an execution of an exercise with type-1 error may not be sufficiently fast to be classified as type-1 error, but at the same time it may not be executed completely correctly to be classified in the "correct" execution type; that is, some samples in the dataset may belong to more than one class. Because the execution types may be subjective, different physiotherapists may label them differently. Another problem in acquiring the dataset is the following: In the experiments, the subjects are asked to perform a particular execution type of an experiment, instead of making a physiotherapist label each execution as correct, type-1 error, or type-2 error. However, the subjects may not perform all the repetitions very well due to tiredness or just by lack of concentration or interest. For instance, all of the 10 correct executions of an exercise may not be perfect, as some executions may be close to type-1 error because of their speeds and some other may be similar to type-2 error because of their amplitudes. Although we tried to reduce this problem as much as possible by repeating the whole experiment if a clear execution error is noticed, all of the executions are still not perfect.

4.3.4 Movement Detection and Classification

The MTMM-DTW algorithm described in Section 4.2.2 is applied to the dataset for the detection and evaluation of physical therapy exercises. For each subject, the template recordings for the three execution types of each exercise are used as the template signals $\mathbf{x}^{(1)}$, $\mathbf{x}^{(2)}$, …, $\mathbf{x}^{(K)}$ with the number of templates K being 24 (= 3 execution types × 8 exercises). The test signal $\mathbf{y}$ is the recording of the same subject's experiment of the exercise, consisting of 30 executions and two idle time intervals. Then, the MTMM-DTW algorithm detects the executions in the recorded signal and classifies them.

There are 45 signals (9 sensors × 5 units) for each recording in the dataset. However, they cannot simply be viewed as a multi-dimensional signal because their units are different (see Section 4.3.1). Therefore, the signals are normalized before being used in the MTMM-DTW algorithm. The normalization is done such that all of the axes of accelerometer, gyroscope, and magnetometer signals have unit variance on the average in the whole dataset, including all the 8 exercises performed by the 5 subjects. To this end, the accelerometer signals are divided by the average standard deviation of the accelerometer signals in the dataset; hence, the x, y, and z-axes of the accelerometers in all 5 sensor units are multiplied by the same coefficient. This is repeated for gyroscope and magnetometer signals. Note that the signals are normalized *before* (not *in*) the MTMM-DTW algorithm; thus, the normalization coefficients of the 45 signals are the same for the template and the test signals; i.e., the template and test signals are jointly normalized and are on the same scale.

In applying MTMM-DTW, uniform local weights are used (i.e., $w_d = w_h = w_v = 1$). The threshold factor is selected as $\tau = 2$ to allow each matched subsequence have at least half of the duration of the matching template. The σ parameter (described on p. 73) is selected to be 0.95 to allow the matched subsequences overlap up to 5% of their durations in the beginning and at the end. In addition, the matched intervals having a DTW distance (per sample of the matching template) less than 10 are omitted, because they are not sufficiently similar to the matching template.

4.3.5 Experimental Results

The success of the proposed MTMM-DTW algorithm is measured in different aspects: the number of executions the algorithm detects, misdetection (MD) and false alarm (FA) rates, specificity and sensitivity values, and accuracy in classifying the exercises and/or the execution types, all of which are summarized for the 5 subjects in Table 4.2 and for the 8 exercises in Table 4.3.

1,125 executions are detected in the whole dataset containing 1,200 executions. This shows that the system makes −6.25% error in counting the exercises. As observed in Table 4.2, the number of detected executions for each subject vary between 194 and 255, where the correct number is 240. Table 4.3 shows that the number of detected executions vary between 115 and 160 for each of the 8 exercises. The variation in the exercises is due to the fact that some exercises inherently contain movements of lower amplitude compared to the others. For example, in exercises 4 and 5, the leg movements are small due to the difficulty of the exercise, and the system can only recognize 85% and 77% of the exercises, respectively. Considering that the algorithm tries to detect both the correct and erroneous movements of two types (executed fast or in low amplitude), it is more difficult to recognize the executions executed in low amplitude. This increases not only the number of MDs, but also the number of FAs because the templates belonging to the low-amplitude executions of the exercises are more similar to the signals in the idle time intervals.

The number of false negatives (idle intervals that are incorrectly recognized as an exercise execution) are simply the number of FAs; that is, the number of exercise executions incorrectly detected in the idle intervals. However, the number of true negatives are also needed to calculate the FA rate, specificity and

	subject						
	1	2	3	4	5	average	total
number of detected executions	194	255	216	216	244		1125
number of actual executions	240	240	240	240	240		1200
accuracy of exercise classification (%)	88.42	93.14	93.17	93.86	97.64	93.46	
accuracy of exercise and execution type classification (%)	80.69	89.29	88.42	88.48	94.62	88.65	
misdetection rate (%)	20.00	0.42	11.25	10.83	0.42	8.58	
false alarm rate (%)	2.65	11.19	3.14	2.14	3.67	4.91	
sensitivity (%)	80.00	99.58	88.75	89.17	99.58	91.42	
specificity (%)	97.35	88.81	96.86	97.86	96.33	95.09	

Table 4.2: The results summarized for all of the 5 subjects.

	exercise									
	1	2	3	4	5	6	7	8	average	total
number of detected executions	160	142	135	127	115	149	151	146		1125
number of actual executions	150	150	150	150	150	150	150	150		1200
accuracy of exercise classification (%)	97.01	90.71	91.92	91.35	83.07	96.36	99.26	94.84	93.46	
accuracy of exercise and execution type classification (%)	93.54	84.29	87.93	85.67	77.01	92.74	94.73	89.57	88.65	
misdetection rate (%)	0.00	7.33	11.33	16.00	26.67	2.67	0.00	4.67	8.58	
false alarm rate (%)	10.85	4.35	3.44	1.71	5.56	4.35	1.17	5.03	4.91	
sensitivity (%)	100.00	92.67	88.67	84.00	73.33	97.33	100.00	95.33	91.42	
specificity (%)	89.15	95.65	96.56	98.29	94.44	95.65	98.83	94.97	95.09	

Table 4.3: The results summarized for all of the 8 exercises.

accuracy values. In order to calculate the number of true negatives (idle intervals that are not recognized as an exercise execution), first the number of samples in the idle intervals are estimated by dividing the interval's duration by the duration of the correctly executed template of the exercise in each experiment, obtaining the number of negative (idle) samples. Then, the number of true negatives is calculated by subtracting the number of FAs from the number of negatives. In this way, the idle time intervals can be counted as negative samples.

The average MD and FA rates[1] are 8.58% and 4.91%, respectively, for the whole dataset (see Tables 4.2 and 4.3). The overall sensitivity and specificity rates are 91.42% and 95.09%, respectively, in the whole dataset.

The recognized executions and the correctness of their evaluation by the system are shown in Figure 4.3. In the figure, the detected executions in the 8 experiments, each containing actually 30 executions of an exercise as well as idle time intervals, are shown as bars along the time axis. The widths of the bars indicate the durations of the executions and the heights show the DTW distance between the executions and the matching templates. That is, the shorter is the bar, the more similar is the matched subsequence to its template. The idle time intervals in between the different execution types of the exercises are clearly observed in the figure, where four FAs occur: two in the first experiment, one in the fifth and one in the seventh experiment. (The figure does not show whether the matched subsequences are FAs. The actual executions and the MDs are also not shown.)

The cumulative confusion matrix that contains the three execution types A–C of exercises 1–8 is shown in Table 4.4. It is obtained by summing up the confusion matrices of the 5 subjects. The last column and the last row indicate the number

[1] The MD and FA rates are calculated as

$$\text{MD rate} = \frac{\text{number of MDs}}{\text{number of positives}} = \frac{\text{number of false negatives}}{\text{number of positives}}$$

and

$$\text{FA rate} = \frac{\text{number of FAs}}{\text{number of negatives}} = \frac{\text{number of false positives}}{\text{number of negatives}}.$$

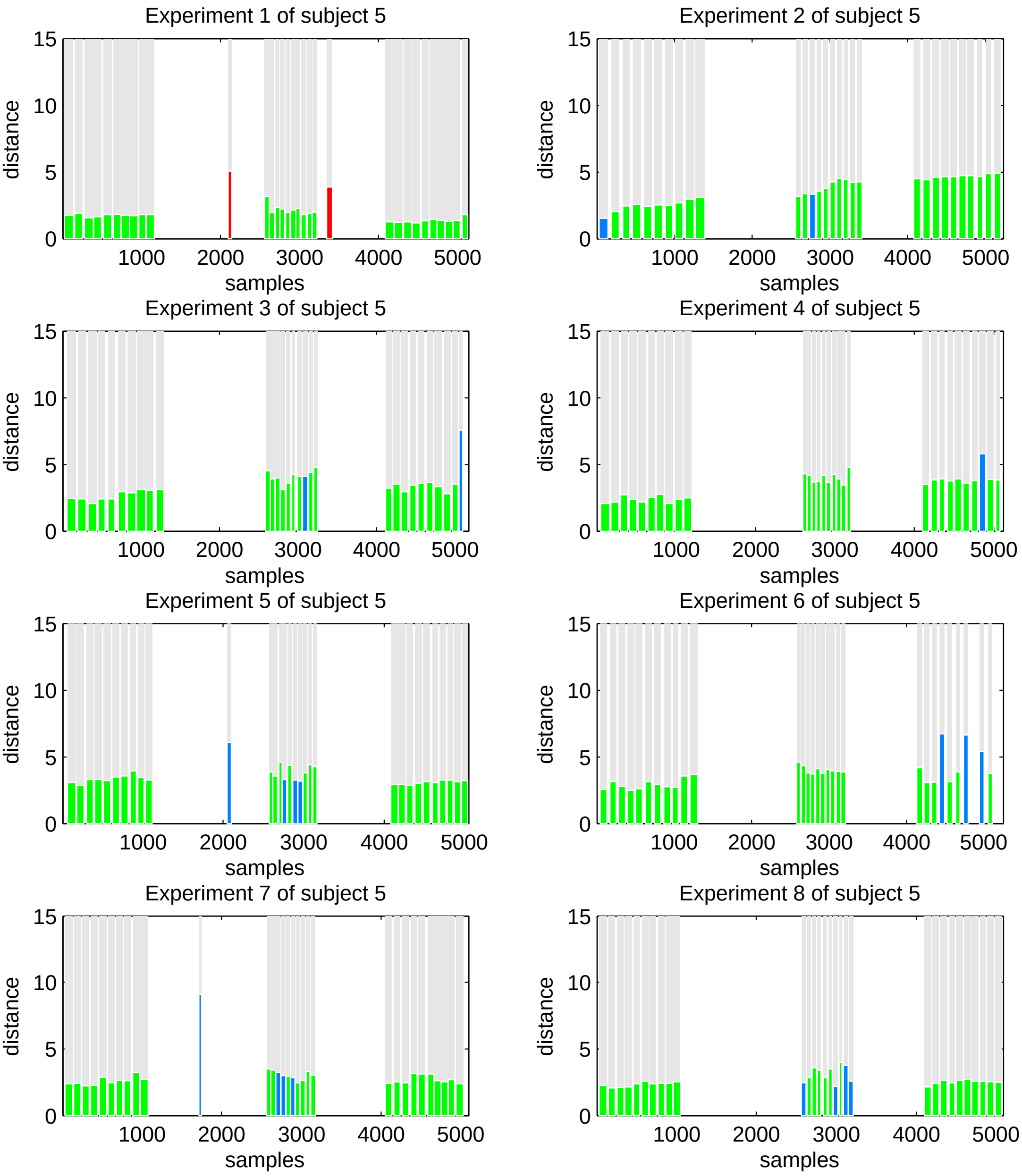


Figure 4.3: Detection and classification of exercise executions in all of the 8 experiments performed by subject 5 corresponding to the 8 exercises. Each detected execution is shown as a bar whose width is the execution's duration and height is the DTW distance between the detected subsequence and the matching template. The bar is red if the execution is classified as an incorrect exercise, blue if the execution is classified as the correct exercise but incorrect execution type, and green if the the execution is classified as correct execution type of the correct exercise. Note that FAs and MDs are *not* shown.

of MDs and FAs in each class, respectively. The total number of MDs and FAs are 103 and 74, respectively.

Although the proposed system does not have the knowledge of exercises and execution types; that is, it only recognizes the 24 classes (1A, 1B, 1C, 2A, ..., 8C), the exercise and execution type classification can be considered separately. The matrix elements corresponding to the correct *exercise* classifications are emboxed in the confusion matrix. These elements contain both correct and incorrect execution type classifications.

Combining the three execution types A, B, and C of each exercise, the 8 × 8 confusion matrix of the 8 exercises are obtained and shown in Table 4.5. It is observed that the system *never* incorrectly classifies the exercise the executions belong to, but it misses some executions or detects some additional executions, yielding MDs and FAs. This shows that the proposed system can also be used in activity recognition provided that the activities are periodic. In case that the periods of the same activity vary too much, multiple templates can be used for one activity to overcome this problem, as is done in this thesis to classify the execution types of the exercises.

The overall accuracy of the system in exercise classification only[2] is 93.46%, whereas the overall accuracy in both exercise and execution type classification[3] is 88.65%. The two accuracy values are summarized for the subjects and exercises in Tables 4.2 and 4.3, respectively. From these tables, it is observed that the performance of the system varies considerably between the subjects and the exercises. For example, the accuracy in exercise and execution type classification

[2] The accuracy of exercise classification is calculated as

$$\frac{\text{number of correct exercise classifications} + \text{number of true negatives}}{\text{number of positives} + \text{number of negatives}}.$$

[3] The accuracy of exercise and execution type classification is calculated as

$$\frac{\text{number of correct exercise and execution type classifications} + \text{number of true negatives}}{\text{number of positives} + \text{number of negatives}}.$$

		estimated classes																									
		1A	1B	1C	2A	2B	2C	3A	3B	3C	4A	4B	4C	5A	5B	5C	6A	6B	6C	7A	7B	7C	8A	8B	8C	MD	total
true classes	1A	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	50
	1B	2	39	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	50
	1C	0	0	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	50
	2A	0	0	0	44	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	50
	2B	0	0	0	3	45	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	50
	2C	0	0	0	7	4	30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9	50
	3A	0	0	0	0	0	0	43	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	50
	3B	0	0	0	0	0	0	3	43	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	50
	3C	0	0	0	0	0	0	0	6	34	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	50
	4A	0	0	0	0	0	0	0	0	0	41	7	0	0	0	0	0	0	0	0	0	0	0	0	0	2	50
	4B	0	0	0	0	0	0	0	0	0	1	49	0	0	0	0	0	0	0	0	0	0	0	0	0	0	50
	4C	0	0	0	0	0	0	0	0	0	5	6	17	0	0	0	0	0	0	0	0	0	0	0	0	22	50
	5A	0	0	0	0	0	0	0	0	0	0	0	0	42	4	1	0	0	0	0	0	0	0	0	0	3	50
	5B	0	0	0	0	0	0	0	0	0	0	0	0	5	21	0	0	0	0	0	0	0	0	0	0	24	50
	5C	0	0	0	0	0	0	0	0	0	0	0	0	5	7	25	0	0	0	0	0	0	0	0	0	13	50
	6A	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	48	2	0	0	0	0	0	0	0	0	50
	6B	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	45	0	0	0	0	0	0	0	2	50
	6C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7	1	40	0	0	0	0	0	0	2	50
	7A	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	49	1	0	0	0	0	0	50
	7B	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	47	0	0	0	0	0	50
	7C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	1	39	0	0	0	0	50
	8A	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	50	0	0	0	50
	8B	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	37	0	7	50
	8C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9	2	39	0	50
	FA	4	6	0	0	16	3	1	4	1	1	2	0	9	2	4	5	4	0	1	2	0	6	0	3		74
	total	56	45	59	54	70	34	47	55	37	48	64	17	61	34	30	63	52	40	63	51	39	71	39	42	103	1274

Table 4.4: Cumulative confusion matrix that contains the three execution types (A: correct, B: type-1 error, C: type-2 error) of all of the 8 exercises (1–8) summed up for all of the 5 subjects. The number of MDs and FAs are shown in an additional column and row, respectively. The elements in the 3×3 blocks on the diagonal correspond to correct exercise type classifications.

		estimated classes									
		1	2	3	4	5	6	7	8	MD	total
true classes	1	150	0	0	0	0	0	0	0	0	150
	2	0	139	0	0	0	0	0	0	11	150
	3	0	0	133	0	0	0	0	0	17	150
	4	0	0	0	126	0	0	0	0	24	150
	5	0	0	0	0	110	0	0	0	40	150
	6	0	0	0	0	0	146	0	0	4	150
	7	0	0	0	0	0	0	150	0	0	150
	8	0	0	0	0	0	0	0	143	7	150
	FA	10	19	6	3	15	9	3	9		74
	total	160	158	139	129	125	155	153	152	103	1274

Table 4.5: Cumulative confusion matrix of all of the 8 exercises (1–8) summed up for all of the 5 subjects. The number of MDs and FAs are shown in an additional column and row, respectively.

varies between 80.69% (subject 1) and 94.62% (subject 5) for the 5 subjects, and 77.01% (exercise 5) and 94.73% (exercise 6) for the 8 exercises. This shows that the proposed methodology depends on the types of exercises as well as the performing subjects. Therefore, the accuracy of the system can be increased by adjusting the parameters (such as the threshold factor and the overlap ratio in the MTMM-DTW algorithm and the local weights in the DTW algorithm) for specific exercises of specific subjects. However, this is not done in this thesis in order to make the system easy to use—the template signals can be recorded by a physiotherapist and then the patient can immediately use the system without any additional tuning.

Note that the computational complexity of the multi-template algorithm is directly proportional to the number of templates used (see Section 4.3.6), so one might use only the three templates of the performed exercise to increase efficiency, in which case the system is assumed to have the information of the exercise the patient is trying to perform, and hence the patient is required to select the exercise in the beginning. In addition, this simplification has a chance to increase the accuracy of the system.

The system proposed here is flexible and adaptable because it does not require previously defined rules, it does not limit the number of templates, and it does not impose constraints on the sensor signals. Therefore, one might remove the templates of the incorrect executions. Then, the system would match the incorrect executions to the correctly executed template signals. In this case, the correctness of the matched executions can be determined based on a threshold applied on their DTW distance. If the distance is large, which means that the execution is not sufficiently similar to the template, the execution is classified as incorrect. Another approach may be using both the incorrect templates and a threshold: if an execution is matched to the correct template with a large DTW distance, the feedback would be incorrect execution. This method would be more robust against unknown errors or movements.

4.3.6 Computational Complexity

The computational complexity of the DTW and the subsequence DTW algorithms is directly proportional to the product of the lengths of the two sequences. The same is true for the STMM-DTW and MTMM-DTW algorithms since they both use the subsequence DTW algorithm. The computational complexity of the MTMM-DTW algorithm is also directly proportional to the number of templates. However, the algorithm repeats a particular process until a condition is reached; thus, its efficiency differs for different signals of the same length. Consider, for instance, two test signals with the first one containing only one occurrence of a long template signal and the second containing 10 occurrences of a short template signal. The computational complexity of the MTMM-DTW algorithm is expected to be higher when the latter signal is used compared to the case with the former signal because there are at least 10 executions of the subsequence DTW algorithm in the latter case, whereas there may be a few subsequence DTW executions in the former case. Therefore, the proposed method does not have low computational complexity. Nevertheless, the algorithm is efficient enough to be used in real time on a laptop with quad-core processor at 2 GHz (Core i7 2630QM) and 8 GB of RAM even when it runs in 32-bit MATLAB. The efficiency increases about 200 times if the subsequence DTW algorithm is programmed in C. If a graphical processing unit (GPU) is also used in addition to the central processing unit (CPU), the computational complexity increases *further* up to 29 times [88]. When a field programmable gate array (FPGA) is used instead of a PC, the subsequence DTW algorithm runs up to 4,500 times faster than its version programmed in C [88], which makes the real time implementation possible even on a low-cost PC or a portable device. To increase the efficiency further, the number of templates at a physical therapy session can be decreased by requiring the patient to select the exercise s/he is going to perform in the beginning of the session and using the templates of (all the execution types of) that exercise.

4.4 Conclusion

In this chapter, a novel algorithm, MTMM-DTW, is proposed to detect all of the occurrences of multiple templates in a signal. The proposed algorithm is flexible because (1) there is no limit for the number of occurrences, number of templates, and durations of signals, (2) the occurrences may be allowed to overlap with each other at a desired level, (3) each of the template signals may be of any length since the DTW distance is normalized with respect to the template durations in the algorithm, (4) the signals may be single- or multi-dimensional, (5) the local weights can be adjusted to favor specific directions in warping the signals, (6) the length of each matching subsequence can be restricted relatively to the template durations to avoid short matches, and (7) any suitable distance function can be used as a dissimilarity measure.

The proposed algorithm is applied to an important problem in physiotherapy: automatically detecting the individual executions of given exercise movements in a physiotherapy session and evaluate them to provide feedback to the patient and the doctor. For this purpose, the patient wears five sensor units containing inertial sensors and magnetometers. The system is trained by the patient performing the assigned exercises in both correct an incorrect ways under the supervision of a specialist. No adjustments specific to patient are needed. Then, the patient can execute the exercises on his/her own, wearing the sensor units in the same way as s/he did in the training phase, and the system provides feedback about the correctness of each execution as well as some statistical information such as how many executions s/he has performed and how many of them were correct for each exercise type. In other words, the system automatically detects the executions in the exercise session, classifies each of them as one of the exercise types, and further classifies them as correctly or incorrectly executed, indicating the error type if there is any.

The success of the system is evaluated with experiments covering eight types of exercise performed by five subjects. Each subject performed each of the three execution types of each exercise once in the training phase and 10 times in the test phase, yielding a dataset consisting of 120 (in the training phase) + 1,200 (in the test phase) = 1,320 executions in addition to idle time intervals. With the proposed methodology, 1,125 movements are detected. 93.46% of the 1,200 executions in the test set are classified as the correct exercise and 88.65% of which are classified correctly in both the execution type and the exercise. The algorithm misses 8.58% of the performed executions and detects 4.91% executions in excess. Considering these outcomes, the performance of the proposed method is acceptable, especially in counting the exercises, which is an important problem in addition to the evaluation of the exercises because counting the executions of the patient can be cumbersome. In addition, the automatic evaluation of executions is very important to obtain an objective result because even direct observation of the exercise session by an expert may lead to a subjective result [62, 63].

The main advantages of the system are that (1) the patient does not need to push a button to indicate the beginning and the end of the exercise executions, wait for a signal to start an execution, nor select the exercise s/he is going to perform, (2) there is no need for the system to be configured for different exercise movements or different patients; the only need is the recording of the template executions of each exercise performed by each patient, (3) the system works independent of the sensor types and placements; hence, any sensor configuration reflecting the movements properly may be directly used without making any adjustments, provided that the configuration is the same in recording the template signals and using the system.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

The scope of theis thesis includes the detection and classification of activities of daily living (ADLs) using two different types of wearable sensors. First, ADLs are classified using tag-based RF localization (in Chapter 2). Secondly, variations in the inertial sensor and magnetometer data of human activities are investigated (in Chapter 3) with respect to the subjects, activities, units, and sensors. Finally, Physical therapy exercises are detected, classified and evaluated using inertial sensors and magnetometers (in Chapter 4). All the measurements are acquired directly on the human body in 3-D because we believe that "activity can best be measured where it occurs" [9][1]. Positions of the RF tags placed on the body are estimated in an RF localization system, whereas the measurements of accelerometers, gyroscopes, and magnetometers reflect the movements of certain body parts in systems utilizing body-worn inertial sensors. The former requires antennas placed in the environment, but the latter can be used anywhere provided that the processing of the data is performed on the body or the data is

[1] This is still valid for RF localization because the positions of the RF tags mounted on the body are estimated, although it requires external antennas placed on the environment and external processing.

transmitted wirelessly to an external unit to be processed. This is not very difficult or costly when inexpensive, computationally powerful, and power-efficient portable systems such as mobile phones are considered.

Although both RF localization and inertial sensor-based systems are wearable, the nature of theses systems and their way of data acquisition are very different. In the former system, the tag positions are estimated and there is often a high-frequency noise that can be filtered out without losing much information. In addition, the performance of the system does not change as time passes. However, accelerometers and gyroscopes provide rate of velocity and angle, respectively, instead of the position or angle at that time. Theoretically, the rate information the inertial sensors provide can be converted to the actual values by integrating them (twice for accelerometers to obtain the position information and once for gyroscopes to obtain the orientation information). However, the measurement errors grow with time unboundedly despite how small they are and the output tends to drift [39, 40]. Furthermore, the bias error at the output of these sensors is not constant but changes with the operating temperature of the unit. It is observed that a very high classification accuracy is obtained in activity recognition with inertial sensors despite that they provide rate information [35] because it may be thought that daily human activities can also be distinguished by the *changes* in the body positions instead of directly the positions of the body parts or the body posture. Acceptable performance is obtained with position measurements, too, in classifying ADLs (see Chapter 2).

The main problem in activity recognition using RF localization is the asynchronous position measurements of the RF tags and missing samples that cause the sampling rate to become non-uniform. This problem is resolved by fitting a curve to the samples and re-sampling the curves uniformly and synchronously. Then, their features can be extracted ordinarily. Although this curve-fitting approach is not mathematically optimal in any sense, acceptable results in classification are obtained by using it.

Classifying and evaluating physical therapy exercises is quite different from activity recognition. In this case, current positions and orientations of body parts seem to be more useful than the velocity rate and angular rate information measured by inertial sensors. For instance, consider a physical therapy exercise consisting of raising the right leg, waiting, and returning to the initial position while lying on a flat surface. The exercise should be evaluated as correct both if the subject completely releases his/her leg so that his/her leg hits strongly to the surface (case 1) or if he/she slowly lowers his leg so that it touches slowly the surface (case 2). The only difference in the position (in particular, height in this movement) of the leg is that the sharp corner in the former case is more rounded in the latter case. This difference would not be significant if the positions are estimated with an RF localization system, as desired. However, since the accelerometer measures the second derivative of the position, this sharp corner in the position curve causes two high peaks with opposite amplitudes in the acceleration data in case 1, whereas there are no such peaks at all in case 2. A similar situation occurs in gyroscope signals. This behavior may be observed in Figure 4.2 as peaks in the acceleration and gyroscope data. Note that, this does not happen in the magnetometer signals because they reflect the exact values of the magnetic field at that instant. Therefore, using an RF localization system can also be considered for the evaluation of physical therapy exercises.

Another factor that reduces the accuracy of the system that is used in physical therapy is the unsupervised classification compared with the activity recognition system. In the latter, most of the classifiers are trained to obtain the best accuracy in the test set, and the feature reduction process weights and combines feature values for the best representation (PCA) or separability (LDA). On the other hand, in the former system, the MTMM-DTW algorithm matches each exercise execution to the most similar template in terms of DTW distance, as in the 1-NN classifier. In addition, feature reduction is not applied (indeed, the features are the signals themselves) to map the samples to a new space for better classification. This unsupervised manner makes the classification process uncontrollable except for specifying some parameters.

Each physical therapy exercise is assumed to have three execution types (one correct and two incorrect) in this study. Also, the two errors are assumed to be the most common errors made by the patients, namely movement executed too fast and movement executed too low in amplitude, referring to a specialist [85]. Sometimes, it is difficult to classify which execution type the executions belong to because some executions may resemble two classes simultaneously and the evaluation may be subjective. For instance, an execution may have a little shorter than the correct one, which may be evaluated as correct or incorrect (fast) even by two specialists. Furthermore, other types of errors may also be involved. These factors also make it difficult to label the true classes of the executions in the experiments, reducing the classification accuracy. For this reason, even if the system does not make any errors, the accuracy cannot be 100% due to the fact that the true classes of some samples are ambiguous.

The main contributions of this thesis are as follows: In Chapter 2, a new approach based on curve-fitting is proposed to solve the asynchronous and non-uniform sampling of the tag positions. Despite that this method is not optimal in any sense, the results demonstrate that high classification accuracy can be obtained with this method. In Chapter 3, the variations in the activity data for the subjects and activities are investigated through the use of distance measures, which has not been studied for inertial sensors before. In addition, the effects of different types of normalization and distance measures are demonstrated. Finally, in Chapter 4, a system that detects and evaluates the executions of the physical therapy exercises is developed based on one of the similarity criteria proposed in Chapter 3. There is no such comprehensive and accurate system that only requires recording three template executions for each exercise in the training phase and then automatically detects the executions without any external help using inexpensive and practical inertial sensors. For this purpose, a novel algorithm is developed as an extension to DTW to detect the occurrences of one or multiple signals in a long signal. This algorithm is quite flexible and adjustable that it may be potentially applied to other areas. The most important advantage of our system compared to the other systems in the literature is that our algorithm does not use the information of physical properties of the subjects, a set of rules that defines the exercise movements, sensor types, sensor orientations and positions on the body, and the number of axes, sensors, or units. The only requirement is that these need to remain the same in recording the templates and using the system. This simplicity also makes it easy to add a new exercise type, a new sensor, or to change some or all of the sensors by a non-expert without re-programming the system.

5.2 Future Work

In the study involving activity recognition using RF localization presented in Chapter 2, activity recognition through tracking of the body parts can be explored using the asynchronously and non-uniformly acquired RFID data in its raw form. Features can be directly calculated from the non-uniformly acquired samples with special techniques and then classification can be performed. HMMs can be used for activity spotting and detecting the transition instants accurately. Variable segment durations that are truncated at the activity transition points can then be considered. The set of activities can be broadened and activity and location information can be combined to provide more accurate results. In addition, the samples themselves may also be used together with the features extracted from them, since the sampling rate (of the fitted curves) is not high, because the samples may provide information about the activity that is not covered by the features. Then, the feature reduction techniques will deal with the dimensionality problem and high computational complexity by selecting some (linear combinations of) samples and features in the reduced space. In addition, the 10 classes other than the “falling” class can be combined to obtain a binary fall detection problem, which may be solved by using HMMs [89].

In investigating the variations in the activity dataset (Chapter 3), the effects of physical properties of the subjects may be compensated by developing a specific method in order to decrease the inter-subject variations of the sensor data. In addition, the DTW distance may be used in a smarter way: To compare two long signals, several random periods in the time-domain data of the periodic activities may be selected and the subsequence DTW algorithm (see p. A.3), which finds the minimum DTW distance between the short signal and a subsequence of the long signal, may be applied to each period and the whole part of the other signal to assess the similarity in a more accurate way.

For the physical therapy part (Chapter 4), the proposed system may be implemented to run in real-time to provide feedback immediately after each execution. The parameters can be optimized differently for the individual exercise types and subjects. The experiments may be performed in a more robust way to minimize intra-class variations. Different sensor technologies may be used, such as RF localization, to directly get the position information instead of accelerometers and gyroscopes that provide the second derivative of the linear position and the first derivative of the angular position (angle), respectively. In this case, the relative positions of the RF tags worn on the body will be estimated in 3-D space without any drift errors that exist in inertial sensors. The main drawback of such a system compared to inertial sensing would be that it would radiate radio waves to the environment. Moreover, the system may be used with only one template of the correct execution of each exercise so that the executions are classified in one of the exercises according to the most similar template and then evaluated by applying a threshold to the corresponding DTW distance—if the distance is below the threshold, meaning that the execution is sufficiently similar to the template, the execution is classified as correct. However, in this case, it would probably be needed to determine the threshold separately for each exercise or each subject, which does not seem to be easy to be done automatically by the system.

Appendix A

Dynamic Time Warping

The standard DTW matches two discrete-time signals [represented as vectors $\mathbf{x} = (x_1 \quad x_2 \quad \dots \quad x_N)^\top$ and $\mathbf{y} = (y_1 \quad y_2 \quad \cdots \quad y_M)^\top$] by "elastically" transforming their time (or sample) axes such that they are most similar to each other. See Figure 3.1(b) for an illustration. To quantize similarity, a *local distance (cost) measure* must be defined between two samples x and y as

$$c(x, y) : \mathcal{F} \times \mathcal{F} \to \mathbb{R}_{\geq 0} \tag{A.1}$$

with $\mathcal{F}$ being the feature space such that $x_n, y_m \in \mathcal{F} \ \forall n, m$ [86]. Since $c(x, y)$ determines the cost of assigning x and y to each other, the more similar $\mathbf{x}$ and $\mathbf{y}$ are, the smaller is the cost (distance). In this thesis, the local cost is selected to be the square of the distance between $\mathbf{x}$ and $\mathbf{y}$: $c(x, y) = (x - y)^2$, as is done usually.

To find the optimal match between $\mathbf{x}$ and $\mathbf{y}$, one can calculate the *cost matrix* $\mathbf{C}$ of size $N \times M$ between each pair of elements of $\mathbf{x}$ and $\mathbf{y}$ as

$$\mathbf{C} = [\ C_{n,m}\ ] = [\ c(x_n, y_m)\ ] \tag{A.2}$$

and find the *optimal warping path* in the cost matrix $\mathbf{C}$ with the smallest cumulative cost.

A warping path can be represented with the sequence $p = (p_1, p_2, \ldots p_L)$ where

$$p_l = (n_l, m_l) \in [1 : N] \times [1 : M], \quad 1 \leq l \leq L. \tag{A.3}$$

There are three basic conditions for the warping path [86]:

1. *Boundary condition*: The path starts from the very first element of the cost matrix and ends at the very last element; i.e., $p_1 = (1, 1)$ and $p_L = (N, M)$.

2. *Monotonicity condition*: The path can proceed to the right, to the bottom, or to any direction in between (such as bottom-right), but it cannot return back; i.e., $n_1 \leq n_2 \leq \ldots \leq n_L$ and $m_1 \leq m_2 \leq \ldots \leq m_L$.

3. *Step-size condition (continuity)*[1]: The path can proceed to the neighbor element at the right, at the bottom or at the bottom-right; i.e.,

$$p_l \in \{p_{l-1} + (0, 1), \quad p_{l-1} + (1, 0), \quad p_{l-1} + (1, 1)\}, \quad 2 \leq l \leq L. \tag{A.4}$$

The *total (cumulative) cost* of a warping path p between the signals $\mathbf{x}$ and $\mathbf{y}$ is defined simply as the sum of the local costs of the matched elements of $\mathbf{x}$ and $\mathbf{y}$:

$$C_p(\mathbf{x}, \mathbf{y}) = \sum_{l=1}^{L} c\,(x_{n_l}, y_{m_l}). \tag{A.5}$$

Then, the *optimal warping path* p^* is the path having minimum total cost among all warping paths between $\mathbf{x}$ and $\mathbf{y}$ satisfying the path conditions:

$$p^* = \arg\min_p C_p(\mathbf{x}, \mathbf{y}) \tag{A.6}$$

The *DTW distance* between $\mathbf{x}$ and $\mathbf{y}$ is then defined as the total distance of the optimal warping path:

$$\mathrm{DTW}(\mathbf{x}, \mathbf{y}) = C_{p^*}(\mathbf{x}, \mathbf{y}) = \min_p C_p(\mathbf{x}, \mathbf{y}) \tag{A.7}$$

[1] There are several choices for the step-size condition. See reference [86] for different step-size conditions.

Unlike its name, DTW distance does *not* satisfy the triangle inequality even if c is a metric and hence DTW distance is not a metric [86].

Instead of an exhaustive search in all possible warping paths between $\mathbf{x}$ and $\mathbf{y}$, which would be extremely inefficient, there is an algorithm with computational complexity $\mathcal{O}(NM)$ based on dynamic programming. For this purpose, *accumulated cost matrix* of size $N \times M$ is defined as

$$\mathbf{D} = [\ D_{n,m}\ ] = [\ \mathrm{DTW}\left(\mathbf{x}_{1:n}, \mathbf{y}_{1:m}\right)\ ] \tag{A.8}$$

where $\mathbf{x}_{1:n} = (x_1 \quad x_2 \quad \cdots \quad x_n)$ and $\mathbf{y}_{1:m} = (y_1 \quad y_2 \quad \cdots \quad y_m)$ are the prefixes of the signals $\mathbf{x}$ and $\mathbf{y}$ with lengths n and m, respectively, for $1 \leq n \leq N$ and $1 \leq m \leq M$. Obviously, $D_{N,M}$ is the desired DTW distance; i.e., $\mathrm{DTW}(\mathbf{x}, \mathbf{y}) = D_{N,M}$. The accumulated cost matrix $\mathbf{D}$ can be computed relatively efficiently with the following equations [86]:

$$\begin{aligned}
D_{n,1} &= \sum_{i=1}^{n} c(x_i, y_1) \\
D_{1,m} &= \sum_{i=1}^{m} c(x_1, y_i) \\
D_{n,m} &= \min\left\{D_{n-1,m-1}, D_{n-1,m}, D_{n,m-1}\right\} + c(x_n, y_m) \\
&\qquad\qquad\qquad\qquad 1 \leq n \leq N, \quad 1 \leq m \leq M
\end{aligned} \tag{A.9}$$

Note that, all the elements of $\mathbf{D}$ must be calculated to obtain the very last element $D_{N,M}$, which is the DTW distance. By using this method, the DTW distance is calculated *without* explicitly finding the optimal warping path p^*. Using $\mathbf{D}$, p^* can be calculated by initializing $p^*_L = (N, M)$ and progressing in reverse order: If p^*_l is computed, p^*_{l-1} is calculated as

$$p^*_{l-1} = \begin{cases} (1, m-1) & \text{if } n = 1 \\ (n-1, 1) & \text{if } m = 1 \\ \arg\min\left\{D_{n-1,m-1}, D_{n-1,m}, D_{n,m-1}\right\} & \text{otherwise.} \end{cases} \tag{A.10}$$

This approach will finally end up with $p^*_1 = (1, 1)$. In other words, starting at the bottom-right element $D_{N,M}$, the optimal warping path p^* steps into the next smallest neighbor element in $\mathbf{D}$ (proceeding only to the left, top, or top-left) and finally ends up with the top-left element $D_{1,1}$.

A.1 Multi-Dimensional Signals

Note that, one can compute the DTW distance and the optimal warping path of two signals $\mathbf{x}$ and $\mathbf{y}$ by knowing only the cost matrix $\mathbf{C}$, which can be computed by using the local cost function $c(x, y)$ defined on every sample x and y of $\mathbf{x}$ and $\mathbf{y}$. In the case of *multi-dimensional signals* $\overline{\mathbf{x}}$ and $\overline{\mathbf{y}}$, the local cost $c\left(\overline{x}, \overline{y}\right)$ can be defined to handle this case such that its range is scalar as before. Then, the DTW and the optimal warping path can be calculated in exactly the same way as done for scalar signals. In this case, the same "warping" is applied to all the dimensions of $\overline{\mathbf{x}}$ and all the dimensions of $\overline{\mathbf{y}}$, considering the overall similarity between $\overline{\mathbf{x}}$ and $\overline{\mathbf{y}}$.

A.2 Local Weights

Local weights w_d, w_h, and w_v may be added to the DTW algorithm in order to favor the diagonal, horizontal, or the vertical direction, respectively, in warping the signals. The weights are used only when calculating the accumulated cost matrix $\mathbf{D}$ in the following way:

$$\begin{aligned}
D_{n,1} &= \sum_{i=1}^{n} c(x_i, y_1) \\
D_{1,m} &= \sum_{i=1}^{m} c(x_1, y_i) && 1 \leq n \leq N, \quad 1 \leq m \leq M \qquad \text{(A.11)} \\
D_{n,m} &= \min \begin{cases} D_{n-1,m-1} + w_d c(x_n, y_m) \\ D_{n-1,m} + w_h c(x_n, y_m) \\ D_{n,m-1} + w_v c(x_n, y_m) \end{cases}
\end{aligned}$$

A.3 Free Endpoints

For some applications, the signals to be matched are cropped manually and hence exhibit idle parts in the beginning and/or at the end. Sometimes, the signals naturally contain a prefix or a suffix that contains no valuable information. Then, the signals should be matched to each other with some unmatched parts in the beginning or at the end of *one* of the signals. From now on, the signals $\mathbf{x}$ and $\mathbf{y}$ are called the *template* and the *test* signals, respectively, assuming the test signal may contain idle prefix and/or suffix parts, whereas the template signal does not. This assumption is valid in pattern classification and pattern search problems, where there are few template signals that are obtained for this purpose and hence do not contain undesired idle parts, but there are many test signals that may contain additional parts in the beginning and/or at the end [86].

Using the standard DTW algorithm, the idle parts in the signals will cause an additional undesired cost, increasing the DTW distance. A better approach may be to ignore the prefix and/or the suffix of the test signal. To this end, the standard DTW algorithm can be modified to allow *free endpoints*, ignoring prefix or suffix parts of the test signal $\mathbf{y}$. The length of the ignored parts are selected optimally in the sense that the DTW distance between $\mathbf{x}$ and the matched subsequence of $\mathbf{y}$ is minimized [41, 86]. If both endpoints are free, the algorithm is called *subsequence DTW* [86] or *open-begin open-end DTW (OBE-DTW)* [41] and the resulting distance is

$$\mathrm{DTW}_{\mathrm{subsequence}} = \min_{m_1, m_2} \mathrm{DTW}\left[\mathbf{x}, \mathbf{y}(m_1 : m_2)\right] \tag{A.12}$$

where $\mathbf{y}(m_1 : m_2) = (y_{m_1} \quad y_{m_1+1} \quad \cdots \quad y_{m_2})$ is the subsequence of $\mathbf{y}$ with $1 \leq m_1 \leq m_2 \leq M$. The m_1 and m_2 values minimizing the DTW distance, namely m_1^* and m_2^*, determine the (optimal) matched subsequence of $\mathbf{y}$ to $\mathbf{x}$. If m_1 is set to 1, only the suffix of $\mathbf{y}$ is excluded. Similarly, if $m_2 = M$, only the prefix of $\mathbf{y}$ is excluded. Obviously, the standard DTW algorithm is obtained if $m_1 = 1$ and $m_2 = M$.

To allow free endpoints, the standard DTW algorithm needs to be modified as follows:

- To exclude the prefix of $\mathbf{y}$, the first boundary condition of the warping path is extended so that $p_1 = (1, m_1)$ with $1 \leq m_1 \leq M$, allowing the beginning point of the warping path to reside anywhere in the first row of the accumulated cost matrix $\mathbf{D}$. Thus, the first $m_1 - 1$ samples of $\mathbf{y}$ are ignored. To this end, the first modification in the standard DTW algorithm is in the calculation of the first row of the accumulated cost matrix $\mathbf{D}$:

$$D_{1,m} = c(x_1, y_m) \tag{A.13}$$

  In this way, the first row of $\mathbf{D}$ consists of the costs of matching x_1 to each element of $\mathbf{y}$ instead of accumulated costs. Since the rest of $\mathbf{D}$ depends on this row, the matrix will be different than the one in the standard DTW. The DTW distance obtained from the newly accumulated cost matrix (DTW distance is simply the last element of $\mathbf{D}$) will simply give the desired distance possibly excluding the suffix of $\mathbf{y}$. The second modification is in the calculation of the optimal warping path. The optimal path p^* is calculated in reverse order as before; however, this time the process ends when the first row of $\mathbf{D}$ is reached; i.e., when $p_1^* = (1, m')$. Here, the column where the process ends is the first sample of the matched part of $\mathbf{y}$: $m' = m_1^*$.

- To exclude the suffix of $\mathbf{y}$, the second boundary condition of the warping path is extended so that $p_L = (N, m_2)$ with $1 \leq m_2 \leq M$, allowing the end point of the warping path to reside anywhere in the last row of the accumulated cost matrix $\mathbf{D}$. Thus, the last $M - m_2$ samples of $\mathbf{y}$ are ignored. For this purpose, the standard DTW algorithm is executed, but this time the DTW distance (allowing exclusion of the suffix of $\mathbf{y}$) is the minimum element in the last row of the accumulated cost matrix $\mathbf{D}$ instead of the last element $D_{N,M}$:

$$\mathrm{DTW}(\mathbf{x}, \mathbf{y}) = \min_{m_2} D_{N,m_2} \tag{A.14}$$

Here, the optimal value m_2^* of m_2 determines the exclusion of the last $M - m_2$ elements of $\mathbf{y}$, simply ignoring the rest of $\mathbf{D}$ on the right side. Obviously, the optimal warping path is now initialized as $p_L^* = (N, m_2^*)$ instead of $p_L^* = (N, M)$ from the end. The suffix of $\mathbf{y}$ is possibly ignored with these two modifications in the standard DTW algorithm.

- To allow the exclusion of both the prefix and the suffix of $\mathbf{y}$, the two modifications to exclude the prefix (in calculating the first row of $\mathbf{D}$ and the new ending criterion in calculating p^*) and the two modifications to exclude the suffix (in obtaining the DTW distance from $\mathbf{D}$ and in the starting point of p^*) explained above must be done in the standard DTW algorithm. Then, the *subsequence DTW* algorithm is obtained.

Note that, as in the standard DTW, the subsequence DTW algorithm is able to handle multi-dimensional signals and local weights. Surprisingly, the computational complexity of the subsequence DTW algorithm is $\mathcal{O}(NM)$, which is exactly the same as the standard DTW, although there are two additional parameters m_1 and m_2 that are jointly optimized to minimize the DTW distance.

Bibliography

[1] B. Logan, J. Healey, M. Philipose, E. M. Tapia, and S. Intille, "A long-term evaluation of sensing modalities for activity recognition," *Proceedings of the 9th International Conference on Ubiquitous Computing (UbiComp 2007), Lecture Notes in Computer Science*, J. Krumm et al.(eds.), vol. 4717, pp. 483–500, Springer: Berlin, Heidelberg, Germany, 2007.

[2] T. B. Moeslund and E. Granum, "A survey of computer vision-based human motion capture," *Computer Vision and Image Understanding*, 81(3):231–268, March 2001.

[3] T. B. Moeslund, A. Hilton, and V. Krüger, "A survey of advances in vision-based human motion capture and analysis," *Computer Vision and Image Understanding*, 104(2–3):90–126, November–December 2006.

[4] L. Wang, W. Hu, and T. Tan, "Recent developments in human motion analysis," *Pattern Recognition*, 36(3):585–601, March 2003.

[5] J. K. Aggarwal and Q. Cai, "Human motion analysis: a review," *Computer Vision and Image Understanding*, 73(3):428–440, March 1999.

[6] G. Sukthankar and K. Sycara, "A cost minimization approach to human behavior recognition," *Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'05)*, pp. 1067–1074, Utrecht, The Netherlands, 25–29 July 2005.

[7] M. Luštrek and B. Kaluža, "Fall detection and activity recognition with machine learning," *Informatica*, 33(2):197–204, 2009.

[8] M. Luštrek, B. Kaluža, E. Dovgan, B. Pogorelc, and M. Gams, "Behavior analysis based on coordinates of body tags," *Ambient Intelligence, Lecture Notes in Computer Science*, M. Tscheligi et al.(eds.), vol. 5859/2009, pp. 14–23, Springer: Berlin, Heidelberg, Germany, 2009.

[9] N. Kern, B. Schiele, and A. Schmidt, "Multi-sensor activity context detection for wearable computing," *Proceedings of the European Symposium on Ambient Intelligence (EUSAI 2003), Lecture Notes in Computer Science*, E. Aarts et al.(eds.), vol. 2875, pp. 220–232, Springer: Berlin, Heidelberg, Germany, 2003.

[10] M. Bouet and A. L. dos Santos, "RFID tags: Positioning principles and localization techniques," *Proceedings of the 1st IFIP Wireless Days Conference*, pp. 1–5, United Arab Emirates, 24–27 November 2008.

[11] Y. Zhang, M. G. Amin, and S. Kaushik, "Localization and tracking of passive RFID tags based on direction estimation," *International Journal of Antennas and Propagation*, vol. 2007, article no: 17426, 2007.

[12] S. A. Weis, "RFID (Radio Frequency Identification)," In the *Handbook of Computer Networks*, vol. 3(3), ch. 198, Wiley Publishing: New York, 2008.

[13] H. Liu, H. Darabi, P. Banerjee, and J. Liu, "Survey of wireless indoor positioning techniques and systems," *IEEE Transactions on Systems, Man and Cybernetics, Part C: Applications and Reviews*, 37(6):1067–1080, November 2007.

[14] P. Steggles and S. Gschwind, "The Ubisense smart space platform," Technical report, Ubisense, St. Andrews House, 90 St. Andrews Road, Chesterton, Cambridge, CB4 1DL, U.K., 2005, `http://www.ubisense.net/`.

[15] R. Want, "An introduction to RFID technology," *IEEE Pervasive Computing*, 5(1):25–33, January–March 2006.

[16] L. M. Ni, Y. Liu, Y. C. Lau, and A. P. Patil, "LANDMARC: Indoor location sensing using active RFID," *Wireless Networks*, 10(6):701–710, 2004, (from the special issue entitled "Pervasive Computing and Communications").

[17] D. Hahnel, W. Burgard, D. Fox, K. Fishkin, and M. Philipose, “Mapping and localization with RFID technology,” *Proceedings of the International Conference on Robotics and Automation*, vol. 1, pp. 1015–1020, New Orleans, LA, U.S.A., 26 April–1 May 2004.

[18] B.-S. Choi, J.-W. Lee, and J.-J. Lee, “Localization and map-building of mobile robot based on RFID sensor fusion system,” *Proceedings of the 6th IEEE International Conference on Industrial Informatics*, pp. 412–417, Daejeon, Korea, 13–16 July 2008.

[19] M. Feldhofer, S. Dominikus, and J. Wolkerstorfer, “Strong authentication for RFID systems using the AES algorithm,” *Cryptographic Hardware and Embedded Systems (CHES 2004), Lecture Notes in Computer Science*, M. Joye and J.-J. Quisquater (eds.), vol. 3156/2004, pp. 357–370, International Association for Cryptologic Research, 2004.

[20] M. Philipose, K. P. Fishkin, M. Perkowitz, D. J. Patterson, D. Fox, H. Kautz, and D. Hahnel, “Inferring activities from interactions with objects,” *IEEE Pervasive Computing*, 3(4):50–57, October–December 2004.

[21] J. R. Smith, K. P. Fishkin, B. Jiang, A. Mamishev, M. Philipose, A. D. Rea, S. Roy, and K. Sundara-Rajan, “RFID-based techniques for human-activity detection,” *Communications of the ACM (special issue: RFID)*, 48(9):39–44, September 2005.

[22] P.-C. Huang, S.-S. Lee, Y.-H. Kuo, and K.-R. Lee, “A flexible sequence alignment approach on pattern mining and matching for human activity recognition,” *Expert Systems with Applications*, 37(1):298–306, January 2010.

[23] L. Wang, T. Gu, X. Tao, and J. Lu, “Sensor-based human activity recognition in a multi-user scenario,” *Ambient Intelligence, Lecture Notes in Computer Science*, M. Tscheligi et al.(eds.), vol. 5859/2009, pp. 78–87, Springer: Berlin, Heidelberg, Germany, 2009.

[24] E. M. Tapia, S. S. Intille, and K. Larson, “Activity recognition in the home using simple and ubiquitous sensors,” *Pervasive Computing (PERVASIVE*

2004), Lecture Notes in Computer Science, A. Ferscha and F. Mattern (eds.), vol. 3001/2004, pp. 158–175, Springer: Berlin, Heidelberg, Germany, 2004.

[25] M. Buettner, R. Prasad, M. Philipose, and D. Wetherall, "Recognizing daily activities with RFID-based sensors," *Proceedings of the 11th International Conference on Ubiquitous Computing (Ubicomp'09)*, pp. 51–60, Orlando, FL, U.S.A., 30 September–3 October 2009.

[26] J. Wu, A. Osuntogun, T. Choudhury, M. Philipose, and J. M. Rehg, "A scalable approach to activity recognition based on object use," *Proceedings of the IEEE 11th International Conference on Computer Vision*, Rio de Janeiro, Brazil, 14–20 October 2007.

[27] M. Stikic, T. Huynh, K. van Laerhoven, and B. Schiele, "ADL recognition based on the combination of RFID and accelerometer sensing," *Proceedings of the 2nd International Conference on Pervasive Computing Technologies for Healthcare*, pp. 258–263, Tampere, Finland, 30 January–1 February 2008.

[28] B. Kaluža, V. Mirchevska, E. Dovgan, M. Luštrek, and M. Gams, "An agent-based approach to care in independent living," *Ambient Intelligence, Lecture Notes in Computer Science*, B. de Ruyter et al.(eds.), vol.6439/2010, pp. 177–186, Springer: Berlin, Heidelberg, Germany, 2010.

[29] Xsens Technologies B.V., Enschede, The Netherlands, *MTi and MTx User Manual and Technical Documentation*, 2009, `http://www.xsens.com`.

[30] *MicroStrain Inclinometers and Orientation Sensors*, MicroStrain, Williston, VT 05495 U.S.A., September 2010, `http://www.microstrain.com/3dm-gx2.aspx`.

[31] A. M. Sabatini, "Inertial sensing in biomechanics: a survey of computational techniques bridging motion analysis and personal navigation," in *Computational Intelligence for Movement Sciences: Neural Networks and Other Emerging Techniques*, R. K. Begg and M. Palaniswami (eds.), pp. 70–100, Hershey, PA, U.S.A.: Idea Group Publishing, 2006.

[32] W. Zijlstra and K. Aminian, “Mobility assessment in older people: new possibilities and challenges,” *European Journal of Ageing*, 4(1):3–12, March 2007.

[33] M. J. Mathie, A. C. F. Coster, N. H. Lovell, and B. G. Celler, “Accelerometry: providing an integrated, practical method for long-term, ambulatory monitoring of human movement,” *Physiological Measurement*, 25(2):R1–R20, April 2004.

[34] W. Y. Wong, M. S. Wong, and K. H. Lo, “Clinical applications of sensors for human posture and movement analysis: A review,” *Prosthetics and Orthotics International*, 31(1):62–75, March 2007.

[35] K. Altun, B. Barshan, and O. Tunçel, “Comparative study on classifying human activities with miniature inertial and magnetic sensors,” *Pattern Recognition*, 43(10):3605–3620, October 2010.

[36] K. Altun and B. Barshan, “Human activity recognition using inertial/magnetic sensor units,” *Proceedings of Human Behavior Understanding (HBU 2010)*, *Lecture Notes in Computer Science*, A. A. Salah et al., eds., vol. 6219, pp. 38–51, Springer: Berlin, Heidelberg, Germany, August 2010.

[37] B. Ayrulu-Erdem and B. Barshan, “Leg motion classification with artificial neural networks using wavelet-based features of gyroscope signals,” *Sensors*, 11(2):1721–1743, January 2011.

[38] O. Tunçel, K. Altun, and B. Barshan, “Classifying human leg motions with uniaxial piezoelectric gyroscopes,” *Sensors*, 9(11):8508–8546, October 2009.

[39] K. Altun and B. Barshan, “Pedestrian dead reckoning employing simultaneous activity recognition cues,” *Measurement Science and Technology*, 23(2):025103, February 2012.

[40] G. Welch and E. Foxlin, “Motion tracking: no silver bullet, but a respectable arsenal,” *IEEE Computer Graphics and Applications*, 22(6):24–38, November–December 2002.

[41] P. Tormene, T. Giorgino, S. Quaglini, and M. Stefanelli, “Matching incomplete time series with dynamic time warping: an algorithm and an application to post-stroke rehabilitation,” *Artificial Intelligence in Medicine*, 45(1):11–34, January 2009.

[42] P. Fergus, K. Kafiyat, M. Merabti, A. Taleb-bendiab, and A. El Rhalibi, “Remote physiotherapy treatments using wireless body sensor networks,” *Proceedings of the International Conference on Wireless Communications and Mobile Computing: Connecting the World Wirelessly*, pp. 1191–1197, New York, U.S.A., 21–24 June 2009.

[43] J. M. Winters, Y. Wang, and J. M. Winters, “Wearable sensors and telerehabilitation,” *IEEE Engineering in Medicine and Biology Magazine*, 22(3):56–65, May–June 2003.

[44] S. L. Wolf, “Electromyographic biofeedback applications to stroke patients,” *Physical Therapy*, 63(9):1448–1459, September 1983.

[45] T. Giorgino, P. Tormene, G. Maggioni, C. Pistarini, and S. Quaglini, “Wireless support to poststroke rehabilitation: Myheart’s neurological rehabilitation concept,” *IEEE Transactions on Information Technology in Biomedicine*, 13(6):1012–1018, November 2009.

[46] A. Yurtman and B. Barshan, “Human activity recognition using tag-based localization (Etiket-tabanlı konumlama ile insan aktivitelerinin tanınması),” *Proceedings of the IEEE 20th Conference on Signal Processing, Communications, and Applications*, 18–20 April 2012, Fethiye, Muğla, Turkey.

[47] “Localization data for person activity data set,” University of California, Irvine Machine Learning Repository, November 2010, `http://archive.ics.uci.edu/ml/datasets/Localization+Data+for+Person+Activity`.

[48] H. Junker, O. Amft, P. Lukowicz, and G. Tröster, “Gesture spotting with body-worn inertial sensors to detect user activities,” *Pattern Recognition*, 41(6):2010–2024, 2008.

[49] E. Guenterberg, H. Ghasemzadeh, V. Loseu, and R. Jafari, “Distributed continuous action recognition using a hidden Markov model in body sensor networks,” *Distributed Computing in Sensor Systems, Lecture Notes in Computer Science*, B. Krishnamachari et al.(eds.), vol. 5516/2009, pp. 145–158, Springer: Berlin, Heidelberg, Germany, 2009.

[50] N. Bicocchi, M. Mamei, and F. Zambonelli, “Detecting activities from body-worn accelerometers via instance-based algorithms,” *Pervasive and Mobile Computing*, 6(4):482–495, August 2010.

[51] A. Mannini and A. M. Sabatini, “Machine learning methods for classifying human physical activity from on-body accelerometers,” *Sensors*, 10(2):1154–1175, February 2010.

[52] T. V. Duong, H. H. Bui, D. Q. Phung, and S. Venkatesh, “Activity recognition and abnormality detection with the switching hidden semi-Markov model,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005)*, vol. 1, pp. 838–845, San Diego, CA, U.S.A., 20–25 June 2005.

[53] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, Wiley: New York, 2nd edition, 2001.

[54] R. P. W. Duin, P. Juszczak, P. Paclik, E. Pekalska, D. de Ridder, D. M. J. Tax, and S. Verzakov, *PRTools 4.1, A MATLAB Toolbox for Pattern Recognition*, Delft University of Technology, The Netherlands, version 4.1, August 2007, `http://www.prtools.org/`.

[55] J. K. Aggarwal and M. S. Ryoo, “Human activity analysis: A review,” *ACM Computing Surveys*, 43(3):16.1–16.43, April 2011.

[56] Y. Sheikh, M. Sheikh, and M. Shah, “Exploring the space of a human action,” *Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV 2005)*, vol. 1, pp. 144–149, Beijing, China, 17–21 October 2005.

[57] A. Veeraraghavan, R. Chellappa, and A. Roy-Chowdhury, “The function space of an activity,” *Proceedings of the IEEE Computer Society Conference*

on Computer Vision and Pattern Recognition, vol. 1, pp. 959–968, New York, U.S.A., 17–22 June 2006.

[58] M. L. Moy, K. Matthess, K. Stolzmann, J. Reilly, and E. Garshick, "Free-living physical activity in COPD: assessment with accelerometer and activity checklist," *Journal of Rehabilitation Research and Development*, 46(2):277–286, 2009.

[59] A. Yurtman and B. Barshan, "Investigation of personal variations in activity recognition using miniature inertial sensors and magnetometers (Minyatür eylemsizlik duyucuları ve manyetometrelerle aktivite tanımada kişiler arası farklılıkların incelenmesi)," *Proceedings of the IEEE 20th Conference on Signal Processing, Communications, and Applications*, 18–20 April 2012, Fethiye, Muğla, Turkey.

[60] A. Yurtman and B. Barshan, "Inter- and intra-subject variations in activity recognition using inertial sensors and magnetometers," *The 5th International Conference on Cognitive Systems, Collection of Posters*, p. 8, Vienna, Austria, 22–23 February 2012.

[61] M. Milenkovic, E. Jovanov, J. Chapman, D. Raskovic, and J. Price, "An accelerometer-based physical rehabilitation system," *Proceedings of the Thirty-Fourth IEEE Southeastern Symposium on System Theory*, pp. 57–60, Huntsville, AL, U.S.A., 18–19 March 2002.

[62] C. Stéphane, H. Mathieu, and B. Patrick, "Accelerometer-based wireless body area network to estimate intensity of therapy in post-acute rehabilitation," *Journal of NeuroEngineering and Rehabilitation*, 5(1):20–31, September 2008.

[63] F. Pitta, T. Troosters, V. S. Probst, M. A. Spruit, M. Decramer, and R. Gosselink, "Quantifying physical activity in daily life with questionnaires and motion sensors in COPD," *European Respiratory Journal*, 27(5):1040–1055, May 2006.

[64] I. Raso, R. Hervás, and J. Bravo, "m-Physio: Personalized accelerometer-based physical rehabilitation platform," *Proceedings of The Fourth International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies*, pp. 416–421, Florence, Italy, 25–30 October 2010.

[65] Y. Tao, H. Hu, and H. Zhou, "Integration of vision and inertial sensors for home-based rehabilitation," *Proceedings of the 2nd IEEE Workshop on Integration of Vision and Inertial Sensors, IEEE International Conference on Robotics and Automation*, Barcelona, Spain, 18 April 2005.

[66] H. Zhou and H. Hu, "Inertial motion tracking of human arm movements in stroke rehabilitation," *Proceedings of the 2005 IEEE Conference on Mechatronics and Automation*, vol. 3, pp. 1306–1311, Ontario, Canada, April 2005.

[67] K. Kifayat, P. Fergus, S. Cooper, and M. Merabti, "Body area networks for movement analysis in physiotherapy treatments," in *Proceedings of the 24th IEEE International Conference on Advanced Information Networking and Applications Workshops (WAINA)*, pp. 866–872, Perth, Australia, 20–23 April 2010.

[68] N. U. Ahamed, K. Sundaraj, R. B. Ahmad, and SAM M. Rahman, "Biosensors assisted automated rehabilitation systems: A systematic review," *International Journal of the Physical Sciences*, 7(1):5–17, 2 January 2012.

[69] E. Jovanov, A. Milenkovic, C. Otto, and P. C. de Groen, "A wireless body area network of intelligent motion sensors for computer assisted physical rehabilitation," *Journal of Neuroengineering and Rehabilitation*, 2(1):6, March 2005.

[70] Y. Higashi, M. Sekimoto, F. Horiuchi, T. Kodama, T. Yuji, T. Fujimoto, M. Sekine, and T. Tamura, "Monitoring rehabilitation training for hemiplegic patients by using a tri-axial accelerometer," *Proceedings of the 23rd IEEE Annual International Conference on Engineering in Medicine and Biology Society*, vol. 2, pp. 1472–1474, Istanbul, Turkey, 25–28 October 2001.

[71] M. Hamel, R. Fontaine, and P. Boissy, "In-home telerehabilitation for geriatric patients," *IEEE Engineering in Medicine and Biology Magazine*, 27(4):29–37, July–August 2008.

[72] B. G. Steele, B. Belza, K. Cain, C. Warms, J. Coppersmith, and J. Howard, "Bodies in motion: monitoring daily activity and exercise with motion sensors in people with chronic pulmonary disease," *Journal of Rehabilitation Research and Development*, 40(5):45–58, September–October 2003.

[73] P. E. Taylor, G. J. M. Almeida, T. Kanade, and J. K. Hodgins, "Classifying human motion quality for knee osteoarthritis using accelerometers," *Proceedings of the IEEE Annual International Conference on Engineering in Medicine and Biology Society*, pp. 339–343, Buenos Aires, Argentina, 31 August–4 September 2010.

[74] J. Brutovsky and D. Novak, "Low-cost motivated rehabilitation system for post-operation exercises," *Proceedings of the 28th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBS'06)*, pp. 6663–6666, New York, U.S.A., 30 August–3 September 2006.

[75] G. Uswatte, C. Giuliani, C. Winstein, A. Zeringue, L. Hobbs, and S. L. Wolf, "Validity of accelerometry for monitoring real-world arm activity in patients with subacute stroke: evidence from the extremity constraint-induced therapy evaluation trial," *Archives of Physical Medicine and Rehabilitation*, 87(10):1340–1345, October 2006.

[76] N. W. Tierney, J. Crouch, H. Garcia, M. Walker, B. V. Lunen, G. DeLeo, G. Maihafer, and S. Ringleb, "Virtual reality in gait rehabilitation," *Proceedings of MODSIM World*, Richmond, VA, U.S.A., 10–13 September 2007.

[77] S. Wilson, R. J. Davies, T. Stone, J. Hammerton, P. Ware, S. Mawson, N. Harris, C. Eccleston, H. Zheng, N. D. Black, and G. Mountain, "Developing a telemonitoring system for stroke rehabilitation," *Contemporary Ergonomics*, 2007:505–512, 2007.

[78] M. Kirilly, *Sensor-Based Skill Assessment for Health and Fitness Applications*, Master's thesis, Department of Media Technology, Technische Universität München, Munich, March 2011.

[79] R. D. Willmann, G. Lanfermann, P. Saini, A. Timmermans, J. te Vrugt, and S. Winter, "Home stroke rehabilitation for the upper limbs," *Proceedings of the 29th IEEE Annual International Conference on Engineering in Medicine and Biology Society*, pp. 4015–4018, Lyon, France, 22–26 August 2007.

[80] H. Zheng, R. J. Davies, and N. D. Black, "Web-based monitoring system for home-based rehabilitation with stroke patients," *Proceedings of the 18th IEEE Symposium on Computer-Based Medical Systems*, pp. 419–424, Dublin, Ireland, 23–24 June 2005.

[81] Y. Tao and H. Hu, "A novel sensing and data fusion system for 3-D arm motion tracking in telerehabilitation," *IEEE Transactions on Instrumentation and Measurement*, 57(5):1029–1040, May 2008.

[82] Ö. Yoleri, M. D., Physical Therapy Specialist, Physical Medicine and Rehabilitation Department, İzmir Katip Çelebi University, Atatürk Training and Research Hospital, personal communication, July 2012.

[83] D. Moss, "Psychophysiological psychotherapy: The use of biofeedback, biological monitoring, and stress management principles in psychotherapy," *Psychophysiology Today: the Magazine for Mind-Body Medicine*, 2(1):14–18, 2005.

[84] S. Anwer, N. Quddus, M. Miraj, and A. Equebal, "Effectiveness of electromyographic biofeedback training on quadriceps muscle strength in osteoarthritis of knee," *Hong Kong Physiotherapy Journal*, 29(2):86–93, December 2011.

[85] İ. Tuğcu, M. D., Physical Therapy Specialist, Department of Physical Therapy and Rehabilitation, Gülhane Military Medical Academy, Turkish Armed Forces Rehabilitation Centre, personal communication, June 2012.

[86] M. Müller, *Information Retrieval for Music and Motion*, vol. 6, Springer: Berlin, Heidelberg, Germany, 2007.

[87] L. Deng, H. Leung, N. Gu, and Y. Yang, "Automated recognition of sequential patterns in captured motion streams," in *Web-Age Information Management* (L. Chen, C. Tang, J. Yang, and Y. Gao, eds.), *Lecture Notes in Computer Science*, vol. 6184, pp. 250–261, Springer: Berlin, Heidelberg, Germany, 2010.

[88] D. Sart, A. Mueen, W. Najjar, E. Keogh, V. Niennattrakul, "Accelerating Dynamic Time Warping Subsequence Search with GPUs and FPGAs," *Proceedings of the 10th IEEE International Conference on Data Mining (ICDM)*, pp. 1001–1006, 13–17 December 2010.

[89] B. U. Toreyin, E. B. Soyer, I. Onaran, and A. E. Çetin, "Falling person detection using multisensor signal processing," *EURASIP Journal on Advances in Signal Processing*, 2008(29):1–7, January 2008.