Roman Chereshnev

HUMAN GAIT CONTROLLING SYSTEM USING MACHINE LEARNING METHODS SUITABLE FOR ROBOTIC PROSTHESES FOR PATIENTS SUFFERING FROM DOUBLE TRANSFEMORAL AMPUTATION

PhD Dissertation

for the purpose of obtaining Philosophy Doctor in Computer Science HSE

Academic supervisor:
PhD, Assistant Professor Attila Kertész-Farkas,

Moscow — 2019
Contents

1 Introduction 5
  1.1 The relevance of research ......................................................... 5
  1.2 Aims and objectives of research .................................................. 8
  1.3 Current approaches ................................................................. 9
  1.4 Importance of work ................................................................. 11
  1.5 Novelty and summary of the Author’s main results ............................. 12
  1.6 Publications ................................................................. 14

2 Literature review 15
  2.1 Overview of the history of HAR .................................................. 15
    2.1.1 Machine learning in HAR .................................................. 17
    2.1.2 Seeking the best classifiers for HAR ....................................... 18
    2.1.3 Ensemble methods for HAR ................................................ 20
    2.1.4 Era of Smartphones ........................................................... 22
  2.2 Review of HAR datasets ........................................................... 23
  2.3 Literature review of methods for human gait inference ......................... 27
  2.4 Review of prosthetic leg controllers .......................................... 28
    2.4.1 High-level controller ......................................................... 29
    2.4.2 Mid-level controllers .......................................................... 32
      2.4.2.1 Phase-based approach ............................................... 32
      2.4.2.2 Non phase-based approach ........................................... 36

3 Overview of the GaIn system 41
  3.1 GaIn system design principles and objectives ................................... 42
  3.2 Main results ................................................................. 44

4 HuGaDB: A human gait database for gait inference 46
  4.1 Motivation and design goals ..................................................... 46
  4.2 Sensor network topology .......................................................... 46
  4.3 Data acquisition programs ......................................................... 47
  4.4 Participants ................................................................. 49
  4.5 Data format ................................................................. 52
  4.6 HuGaDB issues ................................................................. 53
  4.7 Noise ................................................................. 53
  4.8 Conclusions ................................................................. 56
  4.9 Availability ................................................................. 56
This thesis is dedicated to my parents.
1 Introduction

1.1 The relevance of research

Machine learning (ML) methods provide a general framework to adapt algorithms to certain tasks using a large collection of data. ML-based methods excel in several tasks, including image recognition [140], speech recognition [95], and product or service recommendation [132]. This dissertation introduces novel machine learning methods for human activity recognition (HAR) problems.

Broadly speaking, HAR is a field that focuses on recognizing or analyzing the activities performed by humans [90]. Activity recognition may be useful in public surveillance for security reasons, in fall detection in elder care, as well as in gesture recognition, virtual reality, homeland security, robotics, exoskeletons, smart environments, etc. [88, 16]. Human activity analysis can be useful in healthcare, for instance, in inpatient recovery monitoring after surgery, exoskeleton control, monitoring performance improvement, analyzing athletes’ technique in sports, etc.

HAR methods are primarily based on two types of data: visual or sensory. In the first group, HAR methods are mostly based on images or videos captured by cameras. In the second group, the prediction of HAR methods is based on sensory data obtained from inertial sensors, such as accelerometers and gyroscopes of mobile phones or specifically mounted sensors on certain parts of the human body. The topic of this thesis falls into the second category and focuses on sensory data-based HAR.

There are three main areas of HAR: (1) gesture recognition, (2) recognition of activities of daily living, and (3) human gait analysis.

Gesture recognition (GR) mainly focuses on recognizing hand-drawn gestures in the air [96, 91, 45, 55]. Patterns to be recognized may include numbers, circles, boxes, or Latin alphabet letters [45]. Prediction is usually made on data obtained from smartphone sensors or special gloves equipped with inertial sensors, such as 3-axis accelerometers, 3-axis gyroscopes, and occasionally electromyography (EMG) sensors, to measure the electrical potential on the human skin during muscular activities [2].

Recognition of activities of daily living (ADL), on the other hand, aims at recognizing daily lifestyle activities [167, 164, 166]. For instance, an interesting research topic is recognizing activities in or around the kitchen, such as cooking; loading the dishwasher or washing machine; preparing brownies or salads; scrambling eggs; light cleaning; opening or closing drawers, the fridge, or doors; and so on. Often, these activities can be interrupted by, for example, answering phones [137, 70, 68, 116, 33, 24, 122]. In this topic, on-body inertial sensors are usually worn on the wrist, back, or ankle; however, additional sensors, such as temperature sensors, proximity sensors, water consumption sensors, heart rate sensors, etc., can be employed as well.

Human gait analysis (HGA), in contrast, focuses not only on the identification of activities performed by the user but also on how the activities are performed [30]. This can be useful in health-care systems for monitoring patients recovering after surgery, fall detection, and diagnosing
the state of, for example, Parkinson’s disease [125, 124], and even for increasing typing accuracy on touch screens during walking [105]. An unusual gait cycle can be evidence of disease; therefore, gait analysis is important in evaluating gait disorders, as well as neurodegenerative diseases such as multiple sclerosis, cerebellar ataxia, brain tumors, etc. Multiple sclerosis patients show alterations in step size and walking speed [44]. The severity of Parkinson’s disease and stroke shows a strong correlation with stride length [123]. Wearable sensors can be used to detect and measure gait-related disorders, to monitor patient’s recovery, and to improve athletic performance. For instance, EMG sensors can be used to evaluate muscle contraction force to improve performance [154, 152] in running [146] and other sport fields [34]. Emergency fall events can be detected with tri-axial accelerometers attached to the elderly people’s waists [21]. Accelerometers installed on the hips and legs of people with Parkinson’s disease can be used to detect freezing of gait and can prevent falling incidents [10, 125, 124, 31].

**Human gait inference (HGI),** also referred to as human gait trajectory prediction aims at predicting, what the movements of amputated or injured leg parts (thigh, shank, or foot) would be for walking-related activities [139]. HGI methods are often hierarchical and consist of three layers. The first, called high-level control, aims at recognizing the current activity performed by the patient. Once the activity or the intention of the user is recognized, then the high-level controller commands the mid-level controller to infer the appropriate gait. The need for a high-level controller is explained by the fact that the locomotion task requires its own mid-level controller in most of the cases. Mid-level controllers generate the gait trajectory patterns for robotic prosthetic legs or exoskeletons. Mid-level controllers can be categorized into two types: phase-based and non-phase-based. Phase-based mid-level controllers consist of several models that infer gait for particular gait phase. The phases of the human gait are shown in Figure 1. After recognition of the current phase, the mid-level controller performs the appropriate actions. Non-phase mid-level
Figure 2: Concept of robotic prosthetic legs for patients suffering from double trans-femoral amputation. Circles show the location of EMG sensors, and boxes show the location of accelerometers and gyroscopes.

controllers directly aim at predicting the desired gait trajectory, usually based on the physiological motion of the other leg, using linear regression models. The low-level controller carries out the physical control of the robotic legs at close to the hardware level. Even though HGI is closely related to HAR, advanced ML methods are not routinely used in this field, especially in mid-level controller design.

In this thesis, the Author introduces a new system called GaIn (standing for Gait Inference) that is suitable for controlling the robotic prosthetics of patients suffering at most double transfemoral amputation by means of machine learning techniques. The concept of this idea is illustrated in Figure 2. The Author’s idea is based on the observation that the correlation between the movements of the leg parts of healthy people — people without functional gait disorder during usual activities — is high, yet non-linear. Figure 3 shows the non-linear correlation between the thigh and shank angles (of the same leg) during several gait cycles, measured during walking-related activities. The angles of the thigh and shank are measured to the horizontal line. Consequently, it is possible to infer the movements of both lower legs based on the movements of both thighs using machine learning methods. The GaIn system could be installed on microchip- or smartphone-controlled robotic leg prostheses that could be attached to patients in a non-invasive way to infer the movements of the lower limbs, as illustrated in Figure 2. Therefore, the GaIn system could help patients suffering partial or double lower limb amputation to move and walk by themselves. The GaIn system consists of two controllers: (1) a high-level controller, based on the RapidHARe method, for activity mode and patient intention recognition and (2) the non-phase based mid-level controller for gait inference. Both controllers were developed by the Author of this thesis. The first component is based on a dynamic Bayesian model and recognizes whether the patient is sitting, standing, or moving. In a sitting position, GaIn does not allow any gait inference to be performed, so the legs remain motionless. However, when thigh muscle activity is detected by electromyography sensors, the controller performs standing up
activity. When the patient is standing and starts swinging one of his legs, then GaIn activates the gait inference procedure. When a person stands and wants to sit down, the high-level controller can predict his intention based on signals from his muscles. Because human movement is produced by neural mechanisms in the motor cortex of the human brain or spinal neural circuits [106] the Author believes that neurally inspired artificial neural networks could be suitable models for gait inference. Therefore, GaIn uses recurrent neural networks to infer human gait. In addition, GaIn was designed to be fast and computationally inexpensive, with low prediction latency. These features are necessary in order to be applied in mobile devices where energy consumption matters [28]. The Author notes that turning while walking involves rotating the torso, the hips, and the thighs at the hip joints but not the shanks [59]; therefore, our analysis does not involve examination of turning strategies. It should be noted that the methods related to the low-level controller and the actual construction of such robotic prosthetic legs are not part of this thesis.

The inference of the position of the shanks is made on the position (angle) and motion (angular velocity) of the thigh residual limbs, wherein the position and motion are determined using 3-axis accelerometers and 3-axis gyroscopes. In addition, as shown in the experiments, the GaIn system is capable of producing smooth inference for activity changes between several activity modes, including walking at various speeds, taking stairs up and down, sitting down and standing up, as well as running. It should be noted that EMG sensors are used only to recognize standing up or sitting down intention and not for gait inference.

1.2 Aims and objectives of research

The GaIn system could potentially be installed on microchip-controlled robotic leg prostheses that could be attached to patients in a non-invasive way to infer the movements of the lower limbs. In order to make GaIn efficient for use in portable real-time prediction systems, it should meet the following requirements:

A-1: **Low prediction latency.** Gain should respond quickly to sudden changes in user behavior in real-time.
A-2: **Fast and energy-efficient.** In order to be suitable for mobile and portable systems, GaIn is to be energy-efficient and computationally inexpensive.

A-3: **Smooth recognition.** GaIn should provide consistent recognition within a given activity mode and rapid transition in-between activity modes.

A-4: **Generalization.** GaIn should be accurate for new patients whose data was not seen during training.

A-5: **Accuracy.** The GaIn system should be developed using machine learning techniques because these methods have demonstrated the ability to adapt to problem-specific tasks with high accuracy.

The GaIn method carries out the gait inference using accelerometer, gyroscope, and EMG sensors mounted on both thighs. These sensors are inexpensive and widely available.

### 1.3 Current approaches

Here, most relevant methods to this dissertation are discussed briefly. The reader is kindly referred to Chapter 2 for a detailed review of all relevant methods. The first artificial neural network-based (ANN) system to aid patients with spinal cord injuries was developed in 1995 by Sepulveda and his colleagues [126], who showed that ANNs are plausible models to restore muscle signals based on the joint flexion and extension at the hip, knee, and ankle. In addition, the proposed system obeyed voice control to switch between activity modes. The main conclusions of the article are that (1) one needs two separate neural networks for swing and stance phases and that (2) the ANN model requires calibration to the patient. In this PhD thesis, the Author shows that these conclusions are incorrect. First, the Author has built a single neural network which is capable of inferring the gait in both the swing and stance phases. However, the Author emphasizes that higher natural variance was observed in the swing phase for the same person than in the stance phase; however, very accurate gait prediction is not necessarily needed for the swing phase, because the more important supporting work is done by the standing leg. Also, note that voice-based control is not necessary to switch between walking and other activity modes. In addition, sitting down and standing up can be recognized from thigh muscle activities using EMG sensors. As for the second point, the Author suspects that Sepulveda and his colleagues used data from too few patients and thus their model did not generalize well. In this PhD research work, an adequate number of people provided data, and the GaIn system obtained good generalization performance during the training.

Perhaps the best-known non-phase-based mid-level controller is the complementary limb motion estimation (CLME) developed by the group led by Martin Buss [142]. CMLE is based on the idea that the trajectory of a missing leg can be mapped from the movement and the position of the whole sound leg using linear transformations. Therefore, CLME uses information from the state of the whole sound leg and provides an inference method for patients suffering at most single-leg trans-femoral amputation. This work unfortunately has serious limitations: (1) This system
was trained and tested on the same patient; therefore, the system’s generalization performance is unknown. (2) The system was trained and tested only for walking on a treadmill and ascending stairs. (3) Sitting down and standing up were not considered or investigated. Contrary to the CMLE method, the GaIn system uses information from only the movements of the thighs, and it can be used with patients suffering from double leg amputation; GaIn was tested in natural environments in several walking-related activity modes, including the transitions between activity modes, sitting down and standing up. The GaIn system is demonstrated to have low generalization error for new users.

A reinforcement learning-based (RL) method for gait inference was published in January of 2019 by Wen et al. [151], after the Author of this thesis had completed his research project. Wen and his co-authors divided the full gait cycle into four sections and used a reinforcement learning algorithm to set up parameters of each of the four mid-level controllers. The authors achieved root-mean-square error of $3.99 \pm 0.62^\circ$ (compared to the target) for two participants (one healthy and one one-side transfemoral amputee). The authors hypothesized that the feedback from knee kinematics and optimization state was reasonable as a first step towards autonomous mid-level gait control, but questions regarding the appropriate control objective remain open. Unfortunately, the authors tested their system only in laboratory conditions on a treadmill at a steady pace on a flat surface. One of the biggest drawbacks of their approach is that real-life systems should be trained for several ambulatory tasks, such as walking on grass, going up and down stairs, stopping and starting to walk, and so on. In addition, this approach requires having a good high-level controller to transmit impedance function from one locomotion activity mode to another.

The Author has compared the results of Wen et al. to those obtained with GaIn, and the comparison is summarized in Table 1. Note that, unfortunately, a direct comparison cannot be performed because the methods were tested on different data. On the one hand, Wen and his colleagues used one patient to calibrate and test their model in walking activity on a treadmill, and they obtained $3.99^\circ$ root mean squared error (RMSE) for one leg. On the other hand, the Author of this thesis tested the GaIn system to predict the gait trajectory of one leg and used data from several participants in various ambulatory activity modes performed in real environments. GaIn was tested in walking with a subject whose data was not seen during the training, and it achieved $4.75^\circ$ RMSE in this harder scenario. However, when GaIn was trained and tested with the same subject, it achieved an error as low as $3.58^\circ$ during walking activity on average over several subjects. It should be noted that the smallest error that GaIn achieved was $2.37^\circ$ with participant $ID = 6$. It also should be noticed that the GaIn system encapsulates gait trajectory prediction into one neural network model for several ambulatory activity modes, including starting and stopping walking as well, while the method by Wen et al. has been tested only in the walking scenario.

Based on an excellent review by Tucker et al. [139] from 2015, the main drawbacks of current gait inference methods are as follows:

B-1: Some methods assume a fully periodic gait process. It has been shown that this assumption is incorrect [60].
Table 1: Comparison of RL by Wen et al. [151] against GaIn in walking activity.

<table>
<thead>
<tr>
<th></th>
<th>RL from [151]</th>
<th>GaIn on new subject</th>
<th>GaIn on same subject</th>
</tr>
</thead>
<tbody>
<tr>
<td>Root Mean Squared Error (RMSE)</td>
<td>3.99</td>
<td>4.75</td>
<td>3.58</td>
</tr>
</tbody>
</table>

1 Wen et al. trained and tested their model with the same subject. 2 GaIn was tested on a subject whose data were not seen during training. 3 GaIn was trained and tested on the data of the same subjects. 4 The difference between the true and the predicted angles of the shanks.

B-2: Several methods have been developed for only one activity. Moreover, these methods cannot adapt to changes in terrain, and they do not provide procedures to handle starting or terminating an ambulatory activity [86].

B-3: One complication with inference methods is whether they can handle gait inference safely when activity changes between gait phases [139].

B-4: Gait inference methods often require information about the subject, such as length of limbs, the position of the center of mass, and pelvis direction [8, 29, 73, 153].

B-5: The desired impedance function depends on the locomotion task, as the dynamics and kinematics of the joints vary across different locomotion modes [41].

B-6: Current lower limb prosthesis controllers are not capable of transitioning automatically and seamlessly between locomotion modes, such as walking on level ground, stairs, and slopes [161].

1.4 Importance of work

Limb losses occur due to (a) vascular disease (54%) including diabetes and peripheral arterial disease; (b) trauma (45%); and (c) cancer (less than 2%) [168]. Up to 55% of people with a lower extremity amputation due to diabetes will require amputation of the second leg within 2-3 years [110]. In the USA, about 2 million people live with limb loss [168]. In the last 18 years, in Italy, there were 4877 arteriopathic patients who needed lower limb amputations as a consequence of their illness. Sixty-six percent of them were major amputations, of which 73% were transfemoral amputations while only 34% were partial foot or toe amputations [38].

The Author hopes the prosthesis will be a useful tool in combating disability discrimination as is called for under several human rights treaties, such as the Rights of Persons with Disabilities convention by the United Nations [61] and Equality Acts [144, 17] in jurisdictions worldwide. These also mandate access to goods, services, education, transportation, and employment. The Author expects that the GaIn tool will be effective in helping patients tackle common obstacles such as stairs and curbs in urban areas.

The Author assumes that the GaIn system can potentially be useful for exoskeleton controls. Exoskeletons can provide augmented physical power or assistance in gait rehabilitation. In the former case, exoskeletons can be used to help firefighters and rescue workers in dangerous environments, nurses to move heavy patients [78], or soldiers to carry heavy loads [79]. Rehabilitation exoskeletons can be used to provide walking support for elderly people or can be applied in the rehabilitation of stroke or spinal cord injury [147, 133]. The neuromuscular disease cerebral palsy,
which affects the symmetry and the variability of walking, represents the main pathology that requires the use of exoskeletons/prostheses to rehabilitate walking [103].

1.5 Novelty and summary of the Author’s main results

In this thesis, the Author introduces a new method called GaIn for predicting the movements of amputated leg parts for walking-related activities such as walking, taking stairs, sitting down, standing up, etc. This dissertation is supported by three articles, all of them published in international research journals as original articles. A summary of the supporting articles can be found in Table 24.

The GaIn system comprises three main parts: (1) a dataset suitable for training and testing, (2) a high-level controller to recognize the patient’s activity modes and intentions, and (3) a gait inference method to generate the trajectory for robotic prosthetic legs. Below, the novelty and the Author’s results are summarized in three thesis points, and a summary of the supporting articles can be found in Table 24.

1. HuGaDB: the dataset for training the GaIn system [26]. Unfortunately, existing datasets for HGA and HAR were not adequate for the aim of this research project, because they did not contain detailed information on the movements of the parts of the legs. This dataset is unique in the sense that HuGaDB is the first to provide human gait data in great detail, mainly from inertial sensors, and contains segmented annotations for studying the transitions between different activities. The Author constructed the HuGaDB dataset, of which the main and novel characteristics are the following:

(a) The HuGaDB dataset provides information about each part of the human leg during several walking-related activities in great detail, from inertial and EMG sensors. Six inertial sensors (each sensor consisted of one 3D-axis accelerometer and one 3D-axis gyroscope) were mounted on the left and right thigh, shin, and foot, respectively, and a pair of EMG sensors were mounted on the left and right thighs. Therefore, HuGaDB gives detailed information on how each part of the legs moves and how the parts move relative to each other.

(b) The HuGaDB dataset contains continuous recordings of combinations of activities, and the data are segmented and annotated with the label of the activity currently performed. Thus, this dataset is suitable for analyzing both human gait and transition activities.

(c) The data were collected from 18 participants in total. These participants were healthy young adults: four females and 14 males, average age of 23.67 (STD: 3.69) years, an average height of 179.06 (STD: 9.85) cm, and an average weight of 73.44 (STD: 16.67) kg. In total, they provide around 10 hours of data recording.

(d) The HuGaDB article was published in Springer’s Q2 journal Lecture Notes in Computer Science: [26] and it became quite popular among researchers. HuGaDB has been cited by [129, 80, 134, 13, 12] as of 29 March 2019.

2. RapidHARe: the Author developed a novel activity mode and intention recognition method
used in GaIn as a high-level controller called RapidHARe [28]. This method is also suitable for HAR tasks in general.

(a) RapidHARe is based on a dynamic Bayesian network. RapidHARe has low prediction latency (A-1),1 is of fast and computationally inexpensive (A-2), provides smooth recognition (A-3), and generalizes well to new users (A-4).

(b) RapidHARe outperforms all other state-of-the-art HAR methods in accuracy and speed (A-5). RapidHARe reduces the $F_1$-score error rate by 45%, 65%, and 63% and the accuracy error rate by 41%, 55%, and 62% when it is compared to artificial neural networks, recurrent neural networks, and hidden Markov models, respectively.

(c) RapidHARe is used in the high-level controller to predict the patient’s intention for stand up and sit down mainly from data obtained by EMG sensors placed on the skin over the vastus lateralis thigh muscles. The controller achieved 99% precision and recall in recognizing standing up intention and 99% precision and 68% recall in recognizing sitting down intention.

3. GaIn: a gait inference system that is suitable for controlling robotic prosthetic legs [27].

(a) The GaIn framework can be used in lower limb prostheses for patients suffering from double transfemoral amputation, in exoskeleton design, etc. In contrast, most other methods are only suitable for controlling one prosthetic leg.

(b) The GaIn system is based on the observation that the movement of the thigh and shin is highly but non-linearly correlated during regular walking-related activities. This is illustrated in Figure 3. No other method relies on this assumption; in fact, other methods usually extract more data from the sound leg as well.

(c) GaIn infers the shin position based on the position and movement of the thighs using recurrent neural networks with long-short-term memory units. GaIn achieves a prediction error as low as 4.55° on average on natural terrain and generalizes well to new users. In contrast, other methods are often calibrated and tested on the same patient on treadmills.

(d) The GaIn system does not assume a fully periodic gait (B-1)2; it can infer gait for several ambulatory activities (B-2, B-6), has small prediction error during activity transitions (B-3), and does not rely on information about the patients, such as length of limbs, weights, etc. (B-4). These are in contrast to some other methods in the scientific field.

(e) The gait inference model for several ambulatory modes is encapsulated into one single neural network. Other approaches often use different mid-level controllers for different gait phases and activity modes.

(f) The GaIn article was published in the Sensors journal, which is ranked as Q2 by Scopus.

1Cf. the list in section 1.2.
2Cf. the list of drawbacks in section 1.3.
1.6 Publications

The PhD candidate is the main author in all of these articles. All articles have been published in international research journals in English as original research papers. Ranking is based on Scopus and Web of Science. The independent citations are provided as of April 2019.

First-tier publications.
1. Chereshnev R., Kertész-Farkas A.: HuGaDB: Human gait database for activity recognition from wearable inertial sensor networks, Lecture Notes in Computer Science – Springer, 2017. – pp. 131–141. The journal is ranked by Web of Science as Q4 and by Scopus as Q2. This article has obtained five independent citations. HuGaDB article was presented at the 6th International Conference on Analysis of Images, Social networks and Texts and won the best talk award.

2. Chereshnev R., Kertész-Farkas A.: GaIn: Human gait inference for lower limbic prostheses for patients suffering from double trans-femoral amputation, Sensors, – 2018. – Vol. 18. – No. 12. The journal is ranked by Web of Science as Q2 and by Scopus as Q2. This article has been published recently and has not obtained any citations yet.

Second-tier publications.
3. Chereshnev R., Kertész-Farkas A.: RapidHARe: A computationally inexpensive method for real-time human activity recognition from wearable sensors, Journal of Ambient Intelligence and Smart Environments. – 2018. – Vol. 10. – No. 5. – pp. 377–391. The journal is ranked by Web of Science as Q4 and by Scopus as Q3. This article has obtained one independent citation.

Other publications.

Reports at conferences and seminars.

2 Literature review

2.1 Overview of the history of HAR

HAR is a relatively new field of science which can be found in several application areas ranging from security surveillance to healthcare. The first HAR methods relied on video images to recognize human activity and HAR has emerged from the field of computer vision in the early '90s. In these years, several novel applications and approaches have been shown up for human motion recognition. These new applications include human tracking, human movement recognition, and violence recognition (in public surveillance videos). This approach obtains information from video streams and human activity prediction is based on image processing algorithms [149]. In this thesis the Author does not focus on video-based HAR system; however, the reader is referred to [145] for a good review.

Sensor-based HAR has created a new area for human motion analysis, which emerged from the analysis of human dynamics in 1999. Sensor-based HAR has shortly become an alternative way of activity recognition to computer vision-based HAR, which was the mainstream approach in this field at that time. The first studies mainly used very special algorithms and they laid the foundation of sensor-based HAR and showed good accuracy. Researchers started using sensors to analyze human motion but not recognition. Maybe this was due to the facts that (i) computer vision had some encouraging results, and (ii) human activity recognition seemed more plausible on visual data than on sensor-data that time. Moreover, sensors required special equipment for data acquisition which was a problem for scientists at that time. Later, devices such as sensors, smartphones, and laptops become more widely available and cheaper. Sensor-based HAR has a potential application in healthcare especially in patient recovery after surgery or in robot-assisted therapy. These systems use wearable sensors such as accelerometers, microphones, barometers, gyroscope, electromyography, etcetera, installed on the human body in order to recognize and predict human movements, gait, and actions. The first PhD thesis on this area was defended in 2006 [94] which aimed user activity recognition based on his/her geolocation data collected via GPS.

The first work [42] which used wearable sensors for activity recognition for the first time in 1999 has built on human activity analysis reported by [37] and [138], and obtained sensor data from 1-dimensional accelerometers located on the sternum, wrists, thigh, and lower leg. The activity prediction was carried out by the nearest neighbor approach and achieved a 95.6% accuracy for ambulation activities – such as sitting, standing, lying supine, sitting and talking, sitting and operating keyboard, walking, stairs up, stairs down, and cycling – in a laboratory. However, the accuracy dropped to 66% in a natural environment, which is still an impressive result for first work in an area. After all, a random classifier for nine random activities would give us 11.11% of accuracy.

Amini et al. achieved comparable performance to video-based data in sensor-based HAR. In this work, authors mounted accelerometer sensors on the chest and on the rear thigh, and the
data was collected in the following activities: lying, sitting, standing, dynamic and “others”, and they trained a ANN for the activity classification. In the testing phase when the mean, standard deviation, and the entropy of the signals have not reached a predefined threshold, the activity was classified as “other”, otherwise, the activity as classified by the ANN. The experimental results were carried out in the laboratory, and achieve more than 87% accuracy, unfortunately, authors have not tested their system in a real-world environment with new subjects.

In the article Avici et al. [9], authors surveyed different approaches for activity recognition based on inertial sensors, with a focus on applications from health care, well-being, and sports. This survey, first, identified the main challenges and application directions, next analyzed the related work according to the main steps of the activity recognition process: preprocessing, segmentation, feature extraction, dimensionality reduction, and classification. As a general observation, they noted that in almost all cases results reported in the literature are obtained by first gathering the sensory information on a central computer and then processing the data off-line. Performing activity recognition online and in a distributed manner (i.e. with each sensor having just a partial view of the overall situation) remains, therefore, an open research question. However, distributed intelligence also creates new problems and these problems still remain unexplored. One of these problems is finding a way of reaching the best decision with minimum communication and power consumption.

One of the first reviews on human activity recognition was published by Guan et al. in 2011 [53]. This review categorized HAR methods based on the input sources: video-based activity recognition, which remotely monitors human activity using video sensors, and physical sensor activities recognition (PSAR). PSAR consists of two subcategories: wearable sensor-based HAR (WSAR) and object usage based HAR (OUAR). This review paper discussed the main techniques, characteristics, strengths, and limitations of each type of HAR and carried out a comprehensive comparative analysis for them. One of the main conclusions of the review article is that there is no best recognition technique for WSAR. Commonly used methods in the WSAR by that time included Naive Bayes, C4.5 decision trees, and nearest neighbors in different situations. In addition, in this work, they argued that the efficiency of hidden Markov models is limited because it can be difficult to train due to the large abundance of parameters. The OUAR approaches mainly use two types of sensors: radio frequency identification (RFID) tags and simple binary sensor.

In the review of Chen et al. [25], the authors presented a survey of the state of the art research on sensor-based HAR. They summarized the aim, methodology, history, and evolution of the sensor based approaches. Primary approaches and methods were described in the fields of activity monitoring, modeling, and recognition, respectively. Authors identified key characteristics for each individual field and further derived a classification structure to facilitate systematic analysis of the work they surveyed. The analysis has led to some valuable insights for activity modeling and recognition.

The first tutorial on HAR has been published by Bulling et al. [23], which provided an exhaustive introduction to this field. Authors one of the first who discussed the main challenges
that human activity recognition has common with general pattern recognition and identify challenges that are specific to HAR. Later, they described a general-purpose framework for developing and evaluating activity recognition systems. Each component of the framework was detailed and introduced the best practice methods designed by the activity recognition research community. Authors discussed an example of recognizing different hand gestures from data obtained from inertial sensors located on arm. They showed the implementation of each framework part for specific HAR problems and compare how they impact on overall recognition performance.

A comprehensive review on multi-sensor fusion of on-body sensor networks by Raffaele Gravina and his colleagues provides discussion on the motivations and the advantages of multi-sensor data fusion in HAR field by identifying distinctive properties and parameters affecting data fusion design choices [51].

2.1.1 Machine learning in HAR

In this time period scientist have started using machine learning techniques more intensively and the performance of several classic algorithms was evaluated in HAR. Machine learning programs such as waikato environment for knowledge analysis (WEKA) which was released in 1996 were also a great help for researches and this approach has become a standard. In this period not only were more complex data preprocessing tools used such as feature extraction, filtering, selection, and transformation but a much larger variety of activities were considered besides common locomotion activities such as activities of daily living. ADL includes activities such as brushing teeth, cleaning of the room, watching television, reading and so on. In this period the RFID sensors, too, have appeared to recognize things with which people interact.

In 2001, there was first work with feature extraction for HAR [100] such as principal component analysis (PCA), independent component analysis and wavelet transformation. They used multilayer ANN to recognize walking, going upstairs, downstairs based on two accelerometers located on the side of the hip. Best classification results for recognition of different human motions were 83-90% of accuracy.

Huynh et. al. [69] found that the fast Fourier transformation (FFT) of data features always yields the most accurate clustering with K-means. Generally, the highest peaks for the FFT coefficients can be found between the first and the tenth coefficient when the window length is around one to two seconds [69].

The next two studies by [156, 112] focused on activities of daily living recognition using RFID sensors. They tried to recognize activities such as boiling water in the microwave, cleaning a toilet, doing laundry, drinking water, making a snack, using the microwave, using the telephone, using the toilet and so on.

Olguin et. al. published an interesting work for HAR using hidden Markov models (HMM) [108]. Here, the authors obtained that the classification accuracy based on three accelerometers placed in three different parts of the body and evaluated whether there is a significant improvement in recognition accuracy by adding multiple accelerometers or not. They located sensors on right wrist, left hip, and chest. Three different subjects were asked to perform the following sequence
of activities: sit down, run, squat, walk, stand, crawl, lay down and hand movements. Scientists have found that using one sensor, the highest classification accuracy was obtained when one accelerometer was placed on the left thigh (65% accuracy), then the chest (63%) finally the wrist (61%). However, using two sensors, the highest accuracy can be obtained when the sensors are placed on the hip and the wrist (87%). The best accuracy was achieved by using all three sensor locations. In addition, they reported the number of hidden states using K-fold cross-validation, which worked the best in each activity in their work. The optimal number of hidden states varies from 2 to 5 for eight classes.

Junker et al. claim that conventional recognition schemes for continuous classification, such as hidden Markov models, are not directly applicable for HAR since they rely on appropriate zero-class models [72]. In the zero-class motion, events could be embedded into other events at the same time, in partly arbitrary movements. The authors presented a novel, two-stage gesture spotting method based on body-worn motion sensors. The method is specifically designed towards the needs and constraints of activity recognition in wearable and pervasive systems.

![Figure 4: Detailed structure of two-stage recognition framework. Image source: [72].](image)

The quest to find the best sensor placement on different body position has been promoted by Atallah et al. [7]. They provided a systematic framework that can answer the following questions: (1) where is the ideal sensor location for a given group of activities, and (2) which time-frequency features that can be extracted from wearable accelerometers are the most relevant for discriminating different activity types? It is important to mention that sensor placement is far from being a solved problem. Authors used the K-nearest neighbor classifier with different values to assess the effect of outlier points. In addition, they used a Bayesian classifier where Gaussian distributions were used to model the priors of classes. For activities such as vacuum cleaning, walking, cleaning the best location for sensors were chest and wrist. For such activities as running and cycling the best position was the ear-worn sensor since it measures the change in body posture while walking and running. The sensors located on the arm and knee also perform well in the previous activities. For lying down and getting up the best location for sensors is behind the ear.

### 2.1.2 Seeking the best classifiers for HAR

The previous section showed that machine learning techniques are capable of recognizing a wide range of activities at a high level of accuracy. Now, the next questions asked by several researchers were those: which machine learning technique provides the best results. In this time period there can been seen that for every activity, scientists used several machine learning techniques and they
tried to choose the best one. And as it turned out there is no best machine learning algorithm for all scenarios. This negative result is comparable to the no free lunch theorems.

Several studies provided a detailed comparison of many machine learning algorithms on HAR-related problems to see which methods can provide the lowest error. For instance, Bao et al. carried out a comparison of the performance of decision table, instance based learning, naive Bayes, and C4.5 [15], while Wang et al. compared decision tree (DT), C4.5, multiple-layer neural networks, and support vector machine (SVM) [148]. In this study, SVM performed the best when classifying the learned actions and identifying the new unknown actions with strong stability and generalization capability. Parkka and his colleagues compared two decision trees, namely a custom DT and an automatically generated DT to see which one outperforms the other [111]. In addition, an ANN was used as a reference classifier. The custom decision tree was built by using domain knowledge and visual inspection of the signals. An automatically generated decision tree was generated using a Matlab Statistics Toolbox function called “treefit”. The ANN was trained by resilient backpropagation. The custom decision tree treats the different activities more equally than the other classifiers because it optimizes the performance of one node at a time, not the overall performance as the other classifiers do. They used activities such as sitting at home, lying, eating, drinking, shopping, bicycling and so on. With this method, they obtained 86% of accuracy.

Linear discriminant analysis (LDA) is used to reduce the dimensionality of the resulting FFT coefficients in work by Minnen et al. [104]. Classification is achieved by projecting each test point into the LDA space and finding the nearest class centroid using the $L_2$-norm. Hidden Markov models were used to classify each segment according to the observed accelerometer data. First, the data is transformed into a feature vector consisting of the number of peaks within the segment considering all three axes, the mean amplitude of these peaks, and the raw x-axis data. An HMM was trained for each class using a mixture of Gaussian distributions for each observation node. The number of nodes and number of mixture elements in each model was manually specified for each class. Finally, a classification decision is made by selecting the class corresponding to the model with the highest likelihood for the test data.

Andreu et al. [3] used techniques which allows re-training the system as long as the application is running. In this study authors develop and study the application of a recently introduced autonomous machine learning technique called evolving (fuzzy) rule-based systems. The core idea of the proposed paradigm is to infer higher level (human intelligible) knowledge (in the form of fuzzy rules) by processing online in a computationally efficient recursive manner in real-time streams of raw data locally on these wearable wireless sensors and only transmit the high level knowledge instead of transmitting the huge amounts of raw data directly.

Altun and his colleagues provided a comparative study of various methods for the HAR using body-worn inertial sensors [1]. The authors compared Bayesian decision making, a decision tree, the least-squares method, the K-nearest neighbor algorithm, dynamic time warping, SVM, and ANN. The data was collected from users worn five inertial measurement unit (IMU) sensors located on both legs, arms, and chest. The authors performed a PCA method for feature extraction from
the raw signal. In this study authors compare classifiers based on correct differentiation rates, confusion matrices, and computational cost. Also, the authors take into account the pre-processing process, training, and storage requirements for each classifier. Three different cross-validation techniques are employed to validate the classifiers. The results indicate that in general, Bayesian decision making results in the highest correct classification rate with relatively small computational cost.

In one of the latest work by Machado et. al. the authors described a HAR framework using feature extraction and feature selection methods [98]. The authors extract temporal, statistical and frequency characteristic from the raw signal from the 3-dimensional accelerometer. Proposed framework mainly focused on on-body accelerometer sensors activity recognition. The feature selection methods are used for two main reasons: (1) increasing clustering accuracy and reduce computational complexity. Several clustering methods such as K-means, affinity propagation, mean shift, and spectral clustering were applied. The K-means showed best results with 99.29% accuracy for evaluation on data obtained from one participant and 88.57% of accuracy on data from all participants.

2.1.3 Ensemble methods for HAR

The quest in getting the best HAR accuracy has led to the employment of ensemble methods. The article by Ravi et. al. was the first work which used ensemble methods of decision tables, decision trees (C4.5), K-nearest neighbors, SVM, naive Bayes, from WEKA [119]. The authors found out that combined classifiers and popularity voting methods showed good results.

Lester et. al. presented a hybrid approach to recognizing activities. In this approach, the authors used boosting for features selection and ensemble of classifiers for activity prediction. The authors used HMM because this classifier uses temporal information [92]. The authors collected more 12 hours of data, using on-body sensors in natural environments in order to evaluate their hybrid approach. Overall, the authors got 95% accuracy classification for ten different human activities.

In two work by Zhu and his colleagues [165] [166], authors proposed a HAR method by fusing the data from two wearable inertial sensors attached to one foot and the waist of a human subject, respectively. Their multi-sensor fusion based method combines ANNs and HMMs and it can reduce the computational load. They conducted experiments using a prototype wearable sensor system and the obtained results prove the effectiveness and get 90% of the accuracy of their algorithm. The coarse-grained classification module classifies the activities into the following categories: zero displacement activities: standing, sitting, and sleeping; transitional activities: sitting-to-standing, standing-to-sitting, level walking-to-stair walking, stair walking-to-level walking, lying-to-sitting, and sitting-to-lying; strong displacement activities: walking level, walking upstairs, walking downstairs, and running. Two neural networks are designed for the data from the foot and the waist sensor. The output the data from the ANN were categorized into three types: stationary, transitional, and cyclic. The outputs of the neural networks are fed into the fusion module, which integrates the individual types of foot and waist activities and categorizes
the human activities according to special rules.

Figure 5: The overview of the recognition algorithm proposed by Zhu et al. Image source: [165].

In the work by Knan and his colleagues [82], the authors created two-level classification methods for recognizing human activity based on accelerometer sensors. The first level classifier recognizes in which set of states activity belongs. The activity can be a static, transition, or in the dynamic set. This recognition is based on ANN using means of statistical characteristics of the signal. The second level uses autoregressive modeling of acceleration signals. The authors combine autoregressive coefficients, signal-magnitude area and tilt angle to create a feature vector. This vector is used by linear discriminant analysis and ANNs to perform activity recognition. The authors’ methods can recognize 15th an average accuracy of 97.9% using only a single triaxial accelerometer attached to the participant’s chest.

Figure 6: Illustration of the augmented-feature extraction for activity-recognition method. Image source: [82].

Banos et al. [14] have been presented a fusion technique which combines the capabilities of simple classification entities at the class (activity) and source (sensor) levels. The model outperforms direct multiclass approaches with a considerable reduction in the dimensionality of the required feature vector. The use of a hierarchical weighting decision scheme has been demonstrated to significantly improve the scalability and robustness with respect to other traditional fusion techniques. The combination of poor decision entities leads to a decision system which, in general, behaves better or at worst as the best of the constituent entities. The benefits of the
presented methodology could be similarly envisioned in changeable scenarios. In principle, with the fusion at the classification level, the addition or removal of classes and/or sensors do not imply to modify and/or retrain the whole system. Only the corresponding entities should be added or removed with a subsequent updating of the fusion parameters, which might be performed at runtime and without disrupting the normal use of the recognition system.

The main idea in work by Liu et al. [97] is the combination of J48 decision tree, multi-layer perceptrons and logistic regression techniques with the average of probabilities combination rule. According to the experimental results, the authors model provides better performance than multi-layer perceptron based recognition approach suggested in their previous study. These results strongly suggest researchers applying ensemble of classifiers approach for activity recognition problems.

2.1.4 Era of Smartphones

Several HAR methods have been developed for smartphones. Smartphones have become popular and they are already equipped with gyroscope and accelerometer sensors, and users tend to carry their phone all day. The main challenge with smartphones is that they can obtain signals from only one location of the body, often in front or back pockets, in bags, or in hands, moreover, this location is not fixed, it may vary over time. Another challenge associated with the use of telephones is that the computational power and the battery of phones are limited hence computationally exhaustive methods are not recommended. That imposes the limitation on online classification. Furthermore, accelerometers and gyroscopes differ in various types of phones. Therefore, developers may experience problems with calibration on different phone models.

However, the characteristics of using the phone is a very promising area, because sensors, the instrument for data collection and storage and a computing device in one. It is much more convenient than to have a separate device for each part. In addition, smartphones can be combined with smartwatches. Perhaps, further work will focus on a combination of these devices.

To the best of the Author, the first HAR method for mobile systems was introduced by Györbíró and his colleagues [56]. They proposed C4.5 decision trees due to their simplicity, but later they experimented with neural networks. The learning phase was done offline on a desktop workstation using Matlab. Their system recognized and recorded the motional activities of a person using a mobile phone. It should be noticed that the sensors were attached to some body parts of the users, and the data was collected by the mobile phone application that was trained to recognize the activities in real-time.

In work, by Lara et al. [89] were proposed a mobile platform to provide real-time human activity recognition. This system features an efficient library for the mobile evaluation of classification algorithms; and a mobile application for real-time human activity recognition. This platform provided a learning API for Android systems because that time machine learning API’s such as WEKA was not supported by Android.

Lara et al. also wrote a survey on the state of the art in HAR based on wearable sensors [90]. A general architecture is first presented along with a description of the main components of any HAR
system. Authors also propose a two-level taxonomy in accordance with the learning approach and
the response time (either offline or online). Then, the principal issues and challenges are discussed,
as well as the main solutions to each one of them. Twenty-eight systems are qualitatively evaluated
in terms of recognition performance, energy consumption, obtrusiveness, and flexibility, among
others. Finally, authors present some open problems and ideas that, due to their high relevance,
should be addressed in future research.

2.2 Review of HAR datasets

Several databases have been released for benchmarking HAR methods; however, due to a wide
variety of sensor types and the complexity of activities, these databases are rather distinct. Now,
there will be review these areas and the corresponding databases in a taxonomic manner.

**Gesture recognition** mainly focuses on recognizing hand-drawn gestures in the air. Patterns
to be recognized may include numbers, circles, boxes, or Latin alphabet letters. Prediction is
usually made on data obtained from smartphone sensors or some special gloves equipped with
kinematic sensors, such as 3-axis accelerometers, 3-axis gyroscopes, and occasionally EMG sensors,
to measure the electrical potential on the human skin during muscular activities [2]. A database
for gesture recognition is available in [45].

**Recognizing activities of daily living**, on the other hand, aims at recognizing daily lifestyle
activities. For instance, an interesting research topic is recognizing activities in or around the
kitchen, such as cooking; loading the dishwasher or washing machine; preparing brownies or
salads; scrambling eggs; light cleaning; opening or closing drawers, the fridge, or doors; and so on.
Often these activities can be interrupted by, for example, answering phones. Databases on this
topic include the Massachusetts Institute of Technology Place dataset [137, 70], Darmstadt Daily
Routine dataset [68], Ambient Kitchen [116], Carnegie Mellon University Multi-Modal Activity
Database (CMU-MMAC) [33], and Opportunity dataset [24, 122]. In this topic, on-body inertial
sensors are usually worn on the wrist, back, or ankle; however, additional sensors are used, such
as temperature sensor, proximity sensor, water consumption sensor, heart rate and so on. For
instance, CMU-MMAC includes videos, audios, RFID tags, motion capture system based on on-
body markers, and physiological sensors such as galvanic skin response and skin temperature,
which are all located on both forearms and upper arms, left and right calves and thighs, abdomen,
and wrists.

**Gait analysis** focuses not only on the recognition of activities observed but also on how
activities are performed. This can be useful in health-care systems for monitoring patients recov-
ering after surgery or fall detection or in diagnosing the state of, for example, Parkinson’s disease
[125, 124]. For instance, the Daphnet Gait dataset [11] consists of recordings of 10 participants
affected with Parkinson’s disease instructed to carry out activities that are likely to be difficult
to perform, such as walking. The objective is to detect these incidents from accelerometer data
recorded from above the ankle, above the knee, and on the trunk. On the other hand, Bovi et al.
provide a gait dataset collected from 40 healthy people with various ages as a reference dataset
[22].
Datasets for walking related activities. HAR often focus on walking-related activities, such as walking, jogging, turning left or right, jumping, laying down, going up or down the stairs, and so on. Here, the review of HAR datasets for walking-related activities.

WARD: a Wearable Action Recognition Database [158]. This database consists of continuous sequences of human actions measured by a network of wearable motion sensors. There are 20 human subjects in the database: 13 male and 7 female. The current version of WARD includes a rich set of 13 action categories that cover some of the most common actions in a human’s daily activities. WARD includes the following categories: stand, sit, lie down, walk forward, walk left-circle, walk right-circle, turn left, turn right, go upstairs, go downstairs, jog, jump, push the wheelchair. Each subject is asked to perform five trials for one action category. The wireless sensors are instrumented at five body locations: two wrists, the waist, and two ankles. Each custom-built sensor carries a triaxial accelerometer and a biaxial gyroscope. Sensors location for outdoor collection is depicted in Figure 7. In WARD, each type of activities is collected separately in separate Matlab files.

![Figure 7: Location of sensors in WARD. Image source: http://people.eecs.berkeley.edu/~yang/software/WAR/index.html.](http://people.eecs.berkeley.edu/~yang/software/WAR/index.html)

PAMAP2: Physical Activity Monitoring for Aging People dataset [121, 120]. PAMAP2 is a dataset recorded from 18 activities performed by 9 subjects, wearing 3 IMUs (contained 3D-acceleration, 3D-gyroscope, 3D-magnetometer), HR-monitor and thermometer. Dataset contained information from 1 female, 8 males; aged 27.22 ± 3.31 years, body mass index 25.11 ± 2.62 kg m⁻². The authors of PAMAP2 considered activities such as lying, sitting, standing, walking, running, cycling, Nordic walking, watching television, computer work, car driving, ascending stairs, descending stairs, vacuum cleaning, ironing, folding laundry, house cleaning, playing soccer, rope jumping, other (transient activities). Sensors were located on the chest, wrist, and ankle. Although
dataset contains “transitional activities”, the authors emphasize that these activities should be discarded in any kind of analysis. This data mainly covers transient activities between performing different activities, e.g. going from one location to the next activity’s location, or waiting for the preparation of some equipment. There are around 30% of activities labeled as “other”.

**HASC: Human Activity Sensing Consortium challenge** [74, 75, 76]. HASC challenge is created in purpose to collect a large scale human activity corpus. By the end of 2010, by the collaboration of 20 teams, more than 6700 accelerometer data with 540 subjects have been collected through this project. They also developed a tool named “HASC Tool” for management, evaluation, and collection of a large number of activity sensor data. HASC Challenge is not a contest. It is a “technology challenge”. As a result of the long discussion, authors have decided to start gathering a single accelerometer sensor data of simple activities with various kinds of sensors, positions, and sampling rates. By publishing the data with various kinds of sensors, they believed that researchers can find a better configuration of activity recordings. Challenge rules were the following.

1. Each participant should gather at least five subjects with the following activities.
2. 5 set of 6 activities(20 seconds): stay, walk, jogging, skip, going up and going down stairs.
3. 120 seconds of labeled activity sequence which includes all of the above 6 activities. (Each activity should be longer than 5 sec).
4. Each participant can use any kind of sensor but it should be available in the market.
5. Activity data must be described in the HASC data format.
6. Each participant will get all the activity data without label data of the sequence data. They can submit the result of the recognition in the label data format or submit the recognition algorithm.
7. HASC steering member will evaluate the recognition rate.

The major types of sensors are the iPhone and iPod Touch. Every activity file has its own meta-data information. Authors defined a meta-data file format to record the subject’s gender, weight, height, sensor’s terminal type, sampling rate, and position. The number of subjects: males - 89, females - 12, unknown 439.

**University of Southern California human activity dataset (USC-HAD)** [163, 164] consists of well-defined low-level daily activities intended as a benchmark for algorithm comparison, particularly for healthcare scenarios. Authors used high-precision well-calibrated sensing hardware such that the collected data is accurate, reliable, and easy to interpret. Their goal was to make the dataset and research based on it repeatable and extendible by others. The authors used IMU with 6 degrees of freedom, specially designed for human movement collection. Each sensor itself is a multi-modal sensor that integrates a 3-axis accelerometer, 3-axis gyroscope, and a 3-axis magnetometer. Authors have collected data obtained from 14 participants (7 male, 7 female).
USC-HAD contained following activities: walking forward, walking left, walking right, walking upstairs, walking downstairs, running forward, jumping up, sitting, standing, sleeping, elevator up, elevator down. To collect data, authors packed a sensor into a standard-sized mobile phone pouched on the front right hip/waist (Figure 8). Data for each activity is separately stored in files.

Figure 8: Location of IMU sensor in USC-HAD. Image source: [163].

**MAREA: Movement Analysis in Real-world Environments using Accelerometers** [83]. The MAREA is the gait database with data collected in different environments. 20 healthy participants participated in data collection. Overall 12 males and 8 females participated in data collection. Their characteristics: average age: 33.4 ± 7 years, average mass: 73.2 ± 10.9 kg, average height: 172.6 ± 9.5 cm). 3-axes accelerometers were located on participants waist, left wrist and left and right ankles using. Figure 9 shows the position and orientation of each accelerometer at the beginning of each experiment. The participants also wore special shoes with piezo-electric force sensitive resistors. These shoes are collecting information about heel strike and toe off.

11 subjects participated in the indoor experiments and 9 subjects participated in the outdoor experiments. In order to test the performance of gait event detection algorithms in different environmental settings, five different scenarios were defined using the MAREA database, namely indoor walk, indoor walk and run, treadmill all, outdoor walk, an outdoor walk and run.

In some databases, exceptional efforts are taken to provide a reliable benchmark. The body sensor network conference (http://bsncontest.org) [46], for instance, has carried out a contest where organizers provided three different datasets from different research groups. Databases differ in sensor types used and activities recorded. Another team called the Evaluating Ambient Assisted Living Systems Through Competitive Benchmarking – activity recognition, provides a service to evaluate HAR systems live on the same activity scenarios performed by an actor [47]. In this contest, each team brings its own activity recognition system, and the evaluation criteria attempt to capture the practical usability: recognition accuracy, user acceptance, recognition delay, installation complexity, and interoperability with ambient-assisted living systems.

Databases of smartphone sensor data are available in [5, 128, 102].
2.3 Literature review of methods for human gait inference

The first machine learning-based method to emulate rhythmic patterns, to the best of the Author’s knowledge, was introduced in 1992 and relied on a Jordan recurrent neural network [131]. This work showed that recurrent neural networks (RNN) are capable of producing cyclic, sinusoidal signals in time. This is, in fact, interesting because the pattern generator model functions as a finite state machine since the structural organization of the network takes care of the trajectory generation instead of using pace-making cells or a system clock. Moreover, the structure of the RNN was inspired by the biological neural system, which also must be responsible to generate human movements.

Sepulveda and his colleagues, however, showed in 1993 that 16 EMG signals obtained from leg muscles can be mapped to joint angles and joint movements [127]. Here, it should be noticed, that this data is obtained from whole legs (hip, knees, and ankles).

This first ANN system to aid spinal cord injured patients was developed in 1995 by Sepulveda and his colleague [126]. In this work, authors showed that ANNs are plausible model to restore muscles signals based on the joint flexion and extension at hip, knee, and ankle. In addition, the proposed system obeyed to voice control to switch between activity modes. The article is similar to the current PhD thesis to some extent; however, there are striking differences. The main
conclusions of the article are (1) one needs two separate neural networks for swing and stance phases, respectively, and (2) the ANN model requires calibration to the patient. In this PhD thesis, the Author shows that these conclusions are incorrect. First, the Author has built a single neural network which is capable of inferring the gait in both, the swing and stance phases. Note that higher and natural variance was observed in swing phase for the same person; however, very accurate gait prediction is not necessarily needed for the swing phase, because the more important supporting work is done by the standing leg. Also, note that voice-based control is not necessary to switch between walking and other walking-related activity modes such as going up or down the stairs because appropriate gait can be inferred for the variety of walking-related activities from only thigh movements. In addition, sitting down and standing up activities can be recognized from thigh muscle activities using EMG sensors. As for the second point, probably, Sepulveda and his colleagues used data from too few patients and thus their model did not generalize well. In this PhD research work, an adequate amount of people provided data and the GaIn system obtained good generalization performance during the training.

Similar work has been published by Graupel in 1995 [50], in which the authors created an ANN-based control system for paralyzed patients with spinal cord injuries for inferring EMG signals. Additionally, this proposed system is accompanied by a manual control pad to allow effective safe man-machine interaction.

A different approach was introduced by Benbrahim et al. in 1997 [19] for biped robot control. In this work, the authors developed a reinforcement-based learning method to simulate the central pattern generator which is a group of neurons in the human neural system which is directly responsible to control muscles for human walking. A biped robot control faced some plausible restriction such as keeping the body height and posture within a predefined range, keeping the free leg above a certain height, have periodic movements and etc.

The first work to control prosthetic hands using neural networks based on the input EMG signals obtained from biceps and triceps muscles was published by Khoshaba et al. in 1990 [84]. Note that controlling methods for prosthetic hands is not related to this PhD thesis.

### 2.4 Review of prosthetic leg controllers

The modern methods for controlling and generating gait trajectory patterns for robotic prosthetic legs and exoskeletons consist of three levels: (1) a high-level controller which is mostly responsible to recognize the current or the intended activity to be performed by the patient, (2) mid-level controller to infer the appropriate gait trajectory, and (3) low-level controller which is in fact responsible to control the physical device at the architecture level. Now, there will be a review of the high- and mid-level controllers in the following sections. It should be noticed, that low-level controller is not a subject of this thesis work, therefore their review is omitted here. For a good review, the reader is referred to [139].
2.4.1 High-level controller

This subsection is focused on high-level design papers. There are two main classifier approaches: heuristic rule-based and activity recognition based on machine learning.

Early high-level controllers were operated by heuristic rule-based classifiers such as finite state machines [136, 118, 114] or decision trees [93, 107, 77], in which the rules were manually hand-crafted based on observations or prior knowledge to predict or change activities. Each of these methods operates using the same principle: given the set of all possible activities, the researcher identifies a fixed set of rules that indicate the transition from activity to another. However, the number of rules required increases nearly combinatorially with the number of activities [139]. In the following wave, machine learning-based pattern recognition methods have been proposed. These methods employ a variety of classifiers for activity mode recognition.

Young and his colleagues utilized information obtained from the load cell and 3-axis gyroscope and accelerometer from six transfemoral amputees during five activities: walking, ascending/descending stairs and ascending/descending ramps [161]. After data acquisition windowing was employed and the mean, standard deviation, maximum, and minimum values of the window were extracted as features. The windowing technique incorporates historical data. The authors compared 3 classifiers: Gaussian mixture model (GMM) classifier, a majority vote of GMM of 5 previous time windows, and an HMM. A 5-count majority vote used five windows of length 300 ms before heel contact and toe-off events with increments of 20 ms between each window. HMM used 8 GMMs. The authors split steps into two phases (swing and stance) and split each phase into 4 parts. The priors for each step were calculated as the previous steps posterior probabilities multiplied by the states transition probability. A simple GMM performed 95.2% accuracy for activity mode prediction and around 80% accuracy for activity mode transition recognition. Majority vote classifier achieved 92.1% accuracy on locomotion mode recognition and 78% of accuracy on the transition between activity modes. Finally, HMM achieved 98% accuracy on activity mode recognition and around 80% of accuracy on activity mode transitions.

The EMG sensors are popular equipment in activity recognition.

Huang et al. proposed a high-level controller based on EMG signals [66]. In this work, authors tried to solve the problem of big variation of EMG signals patterns within the same activity. The authors proposed to recognize activities based on data obtained in small time windows. They hope that in this small window variation of the signal will be small. The authors designed a special activity recognition algorithm based on information obtained from one phase of gait. They used four phase windows of one stride: (1) immediately after heel contact; (2) prior to toe off; (3) immediately after toe-off, and (4) prior to the next heel-contact. One classifier was trained for each phase window. The authors used an LDA classifier and a small two layers ANN with 10 hidden units and they observed lower classification error with LDA than with ANN. The data was collected from eight participants using sixteen channels of surface EMG sensors located on the muscles of one lower limb. The authors collected data during seven activities: level-ground walking, stepping over an obstacle, ascending stairs, descending stairs, ipsilateral turning, contralateral turning, and standing still from eight able-bodied subjects. The average classification
errors in the four phases were 12.4%±5.0%, 6.0%±4.7%, 7.5%±5.1%, and 5.2%±3.7%, respectively. The authors concluded that their results are promising, however, these accuracies are not enough for a prosthetic leg control system.

In a recent work by Huang et al. [67], the authors used SVM classifier with radial basis function kernel and with sliding window technique based on data obtained from EMG and inertial sensors. However, unlike the previous studies, the authors used continuous gait phases with durations ranging from 200 to over 800 milliseconds with a special majority voting algorithm. In this algorithm, the number of voting decisions increased each time when a rare transition was identified. In this study, the regular number of voting points was 5; the voting length increased up to 15 when a rare transition was classified. The authors achieved over 99% accuracy in the stance phase and 95% accuracy in the swing phase for recognizing activity modes.

Varol et al. used GMM classification for recognizing standing, sitting, and walking activities [143]. As the input signal for classifiers was used joint angles and angular velocities of the prosthesis joints obtained from sensors. The authors extracted mean and standard deviation from data frames and put them into LDA and PCA feature extraction methods, respectively. Authors created special voting scheme consisting of overlapping frames that are classified at each 10 ms interval. In this voting algorithm, the last \(N\) classifier results were saved in a voting vector and activity switching occurs if there were more than 80% of the classification results are in agreement. The main results can be seen in Figure 10. One of the main disadvantages of this work was the high classification time delay. Due to collecting data frame procedure and time requiring for voting, the delay constituted close to 500 milliseconds (half a second).

![Figure 10](image-source)

Figure 10: Classification performance (AUC of ROC) of different number of GMM components with PCA and LDA dimension reduction to (A) one, (B) two, and (C) three dimensions for frame length. Note the y-axis scaling for each dimension. Image source: [143].

Another approach was proposed by Young et al. [160]. Authors used EMG signals from 7 muscles and signals from 13 embedded mechanical sensors of the powered knee and ankle prosthesis to predict the current activity of amputee participant. One LDA classifier was constructed to classify locomotion modes at heel contact, and another LDA classifier was constructed at toe off. Authors tried to create seamlessly changing between locomotion activities – such as transitioning from level walking to stair ascent. Data from three subjects were used to train the classifiers and data from one subject were used to test the classifiers. Four-fold cross-validation was used to
test each of the four subjects as the novel test subject. Accuracy on novel participant was 48% for EMG and mechanical based classifier and 62% for a classifier based on mechanical sensors. However, accuracy was improved by including a few walking trials of the novel subject in the training data, the overall accuracy became 86% with EMG and 83% without. The high error rates observed is a sign of poor generalization to new users and it may be partially due to the small sample size, i.e. the training pool consisted of 3 subjects only. Authors empathized that a possible solution to these issues is the development of a single user-independent intent recognition system that can be applied across multiple users.

Another interesting approach is using information obtained from the brain through electroencephalography (EEG). Mindwalker proposed by Gancet et al. [43] aims at the development of novel brain neural computer interfaces and robotics technologies based on EEG, with the goal of obtaining a crutch-less assistive lower limbs exoskeleton. The main goal of the authors is using motor cortex EEG signals for generating online legs kinematics angles corresponding to the walking pattern and pace as imagined by a participant. Brain signals were processed with a special dynamic recurrent neural network. Complementary brain-computer interfaces control approaches such as arm electromyograms were also investigated. This research also instigated the idea of inference angles of the thigh, shank, and foot by means of only two EMG signals of the arm (rectified, filtered, and smoothed) from sitting participant. The authors noted several challenges to working with EEG signal. The main challenge with the recording of EEG is artifacts: mechanical artifacts, due to relative movement of EEG cap electrodes during the walk, produce random noise which is difficult to filter; and physiological artifacts, due to muscles activity in the vicinity of the cap. Moreover, the authors were interested in developing a concept of dry EEG cap, that could allow, contrary to more traditional wet caps, to be convenient for daily use (i.e. put-on/take-off in a few seconds, and easily cleaned and maintained). However, the authors empathized that the feasibility of their approach could not be confirmed in the current study, although experimental protocols carried out so far do not close the door to this approach.

Other work that used signal obtained from EEG sensors was presented by Duvinage et al. [36] The authors used P300 based command in order to predict 5 activity: low-speed, medium-speed, high-speed walkings, stop and other activity. The P300 evoked potential is a potential elicited 300 ms after a rare and relevant stimulus, visual or auditory, which appears, for example, when the traffic lights are turning from the red to the green. Participant chose these activities by looking and focusing on a special letter on special screen shown in Figure 11. When the user is not looking at the screen, a non-control state is detected leading to no modification of the current speed. Authors used LDA for classifiers for predicting activity. However, authors empathized that there are several problems with their approach. First, the decision time is quite slow for real-time applications. Second, the screen is used for rehabilitation purposes on a treadmill but not for walking in a street.

Kilicarslan et al. [85] introduced an EEG-based classification approach for two tasks: walking, turning right, turning left activities (task 1); sit, rest, stand activities (task 2). Data were obtained from the paraplegic participant and tested with the same participant with NeuroRex exoskeleton
In Task 1, participant walked by a path while exoskeleton was controlled by an operator remotely. The subject was focused only on the given walking, turning right, turning left task. In Task 2 exoskeleton also was controlled by experimenter during performing sitting, standing and rest cycles for 5 minutes. The authors obtained 1280 dimensional feature vector from the EEG sensor. The authors used local Fisher’s discriminant analysis for feature dimension reductions. For classification, the authors used GMM-based classifiers. The mean accuracy was $97.74 \pm 1.2\%$ for task 1 and $99.31 \pm 0.54\%$ for task 2. The experiments showed that there were unique EEG extracted features for each task. The authors got a mean accuracy of $18.78\%$ for task 1 model on test 2 data and $8.9\%$ accuracy for the case when they test the task 2 model with the data of task 1.

2.4.2 Mid-level controllers

The mid-level controllers can be further categorized, depending if the method involves phase-states of the gait cycle or not. In this chapter, the phase-based methods are reviewed first and then the non-phase based approaches are discussed.

2.4.2.1 Phase-based approach

Phase-based controllers divide the full gait cycle into several phases as shown in Figure 1 and infer the leg positions piece-wise by a series of impedance functions [63]. It should be noticed, instead of predicting the leg position, the mid-level controllers aim at predicting the torque to put the robotic joint into the proper position. For example, the behavior of the knee during both stance and swing can be approximated as a linear spring and damper. During stance, the knee behavior is dominated by a relatively high stiffness, whereas in swing, damping is the dominant dynamic in the knee.

In a work by Sup et al. [135], the authors designed finite state machine-based impedance control prosthesis with four states (see in Figure 12). The control strategy was implemented on the prosthesis prototype on a healthy subject using an able-bodied testing adaptor. The parameters were tuned to the user based on the combination of joint sensor data, video data, and feedback from
the user. For example, a user can feel that prosthesis does not create necessary torques during support or push-off. Then experimenters increased or decreased controller parameters. After changing parameters participant gave new feedback. With this iterative process, the impedance functions have been configured.

Figure 12: A finite-state model of gait. Each box represents a state and the transition conditions between states are specified. Image source: [135].

In general, a linear impedance model for each joint within a given activity mode is represented using a linear spring and damper:

\[ \tau = -k(\theta - \theta_k) - b\dot{\theta}, \]

where torque \( \tau \) is written as a function of linear stiffness \( k \) and the damping coefficient \( b \). The \( \theta \) denotes the joint angle, \( \dot{\theta} \) denotes the joint angular speed, and \( \theta_k \) is an equilibrium point.

A linear regression analysis of gait data conducted by David Winter indicated that the joint torque can be sufficiently characterized by a linear combination of joint angle and velocity [155]. Fite et al. [39] designed an active-knee prototype equipped with ground contact at the heel by force sensing resistors. Figure 13 shows the inference results of their controller compared with actual knee torque.

Figure 13: Biomechanical knee torque data (target, blue) and estimate (prediction, red) within a full gait cycle. Image source: [39].
In a work by Gregg et al. [52], the authors used the idea that participant control of prosthetic leg with is was performed by forces created by residual limb at the socket. With this simple idea, the authors built a mid-level controller using virtual constraints approach from robotic leg field. Virtual constraints define desired joint patterns as functions of a phasing variable. However, the output dynamics of a prosthetic leg generally depend on the human interaction forces, which must be measured and canceled by the feedback linearizing control law.

This feedback requires expensive multiaxial strain gauges, and actively eliminating the interaction forces can minimize the human influence on the prosthesis. To eliminate these limitations, Gregg et al. presented a method for projecting virtual constraints into the empty space of the components of human interaction in the output dynamics. Unfortunately, the authors compared their results between the original virtual constraints and the proposed implementation of the virtual constraints when modeling the patient’s prosthetic leg. The authors come to the conclusion that the projection did not significantly change the magnitude of the torque moments in the joints, with the exception of immediately after the impact events. However, the projected gait had a slower stride speed and a longer stride period compared to the original gait, whereas the stride length was basically unchanged.

EMG sensors installed on the residuum have also been tested and used in prosthesis control systems. However, despite advancements in upper extremity EMG-based controllers, little progress has been made in EMG-based active-knee prosthesis control. Using EMG sensors in control systems seem to be a plausible idea; however, EMG sensors can pose challenges for gait control, due partly to difficulty in obtaining reliable EMG measurement “due to noise pick up and movement artifact” [35].

Volitional impedance control framework allows a transfemoral amputee to control the motion of a powered knee prosthesis during none weight-bearing activity [57]. The approach was tested using three transfemoral amputees, and their ability to control knee movement was characterized by a set of knee joint trajectory tracking tasks. The average root mean square trajectory tracking errors of the prosthetic knee employing the EMG-based volitional control were 6.2°. However, this work focused on none walking movement. Participants movements while sitting, it was not tested in walking related activity modes.

An exoskeleton for knee supporting with extra force was developed by Fleischer et al. in 2008 [40]. They created a control system based on signals obtained from 6 EMG sensors located on one leg to predict the movement of the leg and posed extra force in order to help a person during locomotion. Authors used analytic models for force inference; however they did not use any information about the trajectory of movement, they only tried to use the correlation between muscles activity and limb rotations around joints. This system was tested and calibrated only on one person and authors empathized that their controlling model can only support real leg and cannot be suitable for devices gait trajectory prediction. Moreover, their system required prior information about the length of the muscle fibers and length of the tendon. In addition, their model turned out to be very sensitive to these quantities that vary between subjects. As for results, the movement generated was not smooth. This originates from the waviness of the EMG...
signals. Since the support is directly depending on the prediction of the operator’s contribution, the resulting support is very unsteady.

Pfeifer and his colleagues aimed at finding body point (predictors) with the biggest correlation with the EMG signal of the vastus medialis muscle [115]. They carried out supervised cross-validation estimation, namely, in each combination, one of the subjects serves as validation subject and the data from the other four subjects is used to compute the best predictors. The authors found that the biggest correlation coefficient already reaches a local maximum for three predictors, but these predictors differed in different validation folds.

Impedance methods based on direct EMG signals were developed [64, 65] as well in the following form

\[
\tau = K_f u_f (\theta_{\text{max}} - \theta) - K_e u_e (\theta - \theta_{\text{min}}),
\]

where subscripts \(f\) and \(e\) stands for flexion and extension parameters. \(u_f\) and \(u_e\) represent myoelectric commands from flexor and extensor muscles obtained from EMG sensors located on the residual limb and normalized to maximum voluntary contraction. Limb impedance is characterized by spring stiffness constants \(K_f\) and \(K_e\) and spring set-point constants \(\theta_{\text{min}}\) and \(\theta_{\text{max}}\), where \(\theta\) represents knee angle and \(\tau\) represent desired knee torque.

Unfortunately, it turned out that adequate limb-control performance can be achieved upon extensive training of the amputee patient with the EMG-controlled limb. While the patient is able to exploit the myoelectric impedance control structure to walk with the prosthesis, the means by which locomotive function is achieved deviates from that seen in the biomechanics of an intact knee during walking. Specifically, while the participant demonstrated the ability to directly modulate knee impedance during level walking, the myoelectric commands during late support and late swing resulted in desired impedance terms that deviate significantly from the knee impedances exhibited by able-bodied subjects. However, it should be noted that the participant was able to achieve effective volitional knee control in the absence of any significant proprioceptive and haptic feedback. The participant can detect ground interaction indirectly through the socket interface and visually track the limb position.

It has been pointed out by most of the researchers that working with EMG signals is very problematic. A drawback to the effectiveness of an EMG-based control architecture is the sensitivity of surface EMG measurements to movement artifact and perspiration. Movement artifact resulting from relative motion between the EMG electrodes and underlying soft tissue can cause tribologic potentials that are unrelated to the contractile activity of the underlying muscle.

Finally, as was mention earlier a very recent method for a mid-level controller, which was published in January of 2019 by Wen et al. [151], after the Author of this thesis had completed his research project.

Wen and his co-authors divided the full gait cycle into four sections and they used a reinforcement learning algorithm to set up parameters of a mid-level controller given in the form of Equation 1. They developed a special algorithm to configure 12 impedance parameters for each locomotion phase. The authors achieved root-mean-square error 3.99 ± 0.62° (compared to the target) error for two participants (one healthy and one side trans-femoral amputee). The authors
hypothesize that feedback from knee kinematics and optimization state was reasonable as a first step towards autonomous mid-level gait control, but questions regarding to the appropriate control objective remain open. Uninformatively, the authors tested their system only in laboratory condition on a treadmill at a steady pace on a flat surface. One of the biggest drawbacks of their approach is that real-life systems should be trained for several ambulatory tasks, such as walking on grass, going up and down on stairs, stopping and starting walking, and so on. In addition, this approach requires to have a good high-level controller to transit impedance function from one locomotion activity mode to another.

The Author have compared their results to the ones obtained with GaIn and the comparison is summarized in Table 2; however, a direct comparison cannot be performed. The methods were tested on different data. On one hand, Wen and his colleagues used patient to calibrate their model in walking activity on a treadmill and they obtained 3.99° or RMSE (root mean squared error). On the other hand, the Author of this thesis used data from several participants in various ambulatory activity modes performed in the real environment and GaIn was tested in walking with a subject whose data was not seen during the training. The GaIn achieved 4.75° RMSE in this harder scenario. However, when GaIn was trained an tested with the same subject, it achieved an error as low as 3.58° during walking activity on average over several subjects. It should be noticed that the smallest error that GaIn achieved was 2.37° with participant $ID = 6$. It also should be noticed that the GaIn system encapsulates gait trajectory prediction into one neural network model for several ambulatory activity modes including starting and stopping walking as well, while the method by Wen et al. has been tested only in the walking scenario. It also should be noticed that GaIn is a non-phase based approach (discussed below), while Wan’s method relies on a precise phase-state estimation.

<table>
<thead>
<tr>
<th></th>
<th>RL from [151]</th>
<th>GaIn on new subject</th>
<th>GaIn on same subject</th>
</tr>
</thead>
<tbody>
<tr>
<td>Root Mean Squared Error (RMSE)</td>
<td>3.99</td>
<td>4.75</td>
<td>3.58</td>
</tr>
</tbody>
</table>

1 The GaIn was tested on a subject whose data was not seen during training. 2 GaIn was trained and tested on the data of the same subjects.

### 2.4.2.2 Non phase-based approach

Non-phase based methods do not utilize information about the current state of the gait cycle. Relatively few methods have been developed in this category. Perhaps the most known method in this category is the complementary limb motion estimation developed by the group of Martin Buss introduced in 2006 [142]. This method lies in the idea that the trajectory of the missing leg can be predicted from the movement of the sound leg using linear transformations.

A typical covariance matrix $\Sigma$ includes information about the variance of each variable, as well as their correlations. Let $d$ denote the dimension of the data vector. It can be shown that the $d$ eigenvectors of $\Sigma$ form the orthonormal basis of the optimal coordinate transformation. The matrix $\Gamma$ formed by the eigenvectors, which are sorted in descending order of the corresponding eigenvalue, maps $\mathbf{x}$ on the new coordinates $\mathbf{y}$:
\[ y = \Gamma^T x. \] (3)

Due to the fact that for orthonormal matrices, the inverse is equal to the transpose, \( x \) can be reconstructed from \( y \) by

\[ x = \Gamma y. \] (4)

The equation 4 is called overdetermined, if the number \( d-q \) of known components of \( x \) is at least equal to the dimensionality \( p \) of \( y \). Thus, it can be solved for unknown \( y_i \) and partly unknown \( x_i \). Assuming that the first components \( x_1 \in \mathbb{R}^{(d-q)} \) of \( x \) are known, and the remaining part \( x_2 \in \mathbb{R}^q \) is unknown, 4 can be written as:

\[ x_1 = \Gamma_1 y, \] (5)

\[ x_2 = \Gamma_2 y, \] (6)

where \( \Gamma_1 \in \mathbb{R}^{(d-q) \times p} \) and \( \Gamma_2 \in \mathbb{R}^{q \times p} \) are the corresponding submatrices of \( \Gamma \). Thus, \( x_2 \) can be reconstructed from \( x_1 \) by

\[ x_2 = \Gamma_2 \hat{\Gamma}_1 x_1, \] (7)

where \( \hat{\Gamma}_1 \) is the left pseudo-inverse of \( \Gamma_1 \):

\[ \hat{\Gamma}_1 = (\Gamma_1^T \Gamma_1)^{(-1)} \Gamma_1^T. \] (8)

The authors used data related to 10 healthy male subjects from the Carnegie Mellon video based database [33]. In these videos, participant walk-in straight line for several seconds. The authors evaluate three experiments:

1. Inference of hip and knee angles based on information from another leg for the same participant (Figure. 14),

2. Inference of hip and knee angles based on the \( \Gamma \) build from other leg based on averaged reference data (9 participants) and tested on one excluded participant (Figure. 15),

3. Inference of knee angles and angular speed based on information from other leg and same leg thigh based on averaged reference data (Figure 16))

The results show that PCA filtering yields good estimation for hip angels and a quite good estimation for knee angles. Unfortunately, the authors did not provide any metric numbers in their articles. Results are shown in 15(A) in black dotted line shows that, however, the performance of the system drops significantly for unseen participants. To circumvent this drawback, the authors applied the Kalman filter and the results are shown in 15(B) by the red line.
Figure 14: Estimates of the hip ($\phi_h$) and knee angles ($\phi_k$) based data obtained from the same participant. PCA reconstruction is based on angle data (A), angles and angular velocities (B), angles, angular velocities and angular accelerations (C). The number of principal components was 2, 4, and 5. The dashed blue line is the original trajectory, the black dots indicate estimates. Image source: [142].

Figure 15: Estimates based on averaged reference data. (A) Estimation of the hip ($\phi_h$) and knee angles ($\phi_k$). (B) Kalman filtering of the knee angle ($\phi_k$) and the angular velocity ($\dot{\phi}_k$). The dashed blue line is the original trajectory, the black dots the estimation, and the solid red line shows the estimation obtained with Kalman-filter. Image source: [142].

To further improve the performance of the CLME method, PCA and the best linear unbiased estimation (BLUE) method were employed in a sequel work by the same group [141]. The BLUE method can be formulated as follows:

$$
\begin{bmatrix}
\phi_p \\
\dot{\phi}_p
\end{bmatrix}
= K
\begin{bmatrix}
\phi_h \\
\dot{\phi}_h
\end{bmatrix}
+ k,
$$

(9)

where $\phi_h$ and $\dot{\phi}_h$ are angles and angular speeds of human body joints, respectively, and $\phi_p$ and $\dot{\phi}_p$ denote angles and angular speeds of the prosthetic leg, respectively. The coefficients of $K$ and $k$ were estimated from the joints ($\phi$ and $\dot{\phi}$) of healthy people. The mean values ($\bar{\phi}$ and $\bar{\dot{\phi}}$) and STD (standard deviation), subsumed in the diagonal matrix $S$, are extracted for all joints. Using this information, the normalized state vector $x_h$ was defined, containing only the data of human joints that is available in the amputee subject:

$$
x_h = S_h^{-1}
\begin{bmatrix}
\phi_h \\
\dot{\phi}_h
\end{bmatrix}
- 
\begin{bmatrix}
\bar{\phi}_h \\
\bar{\dot{\phi}}_h
\end{bmatrix}.
$$

(10)

The same was used for the states of the prosthetic joint(s):
Figure 16: Estimation of knee motion only. Depicted are estimates of the knee angle and angular velocity based on the data obtained from the same participant (A), and on averaged gait data (B). The number of principal components used is 7 in both cases. The dashed blue line is the original trajectory and the solid red line is estimation obtained with Kalman filter. The source of images: [142].

\[
x_p = S_p^{-1} \left[ \begin{bmatrix} \dot{\varphi}_p \\ \ddot{\varphi}_p \end{bmatrix} - \begin{bmatrix} \bar{\dot{\varphi}}_p \\ \bar{\ddot{\varphi}}_p \end{bmatrix} \right].
\]  

The estimate of \( x_p \) as a function of \( x_h \) is then found by minimizing the expected error, formulated as

\[
E(||x_p - Cx_h||^2) \rightarrow \text{min}.
\]

Using the covariance matrices \( M_{hh} \) and \( M_{hp} \) of the respective data vectors in recorded physiological motion, the solution is given by:

\[
C = (M_{hh}^{-1}M_{hp}^{-1}),
\]

\[
\hat{x}_p = Cx_h.
\]

The outputs are augmented with mean and standard deviation of the physiological motion, which gives reference angle and velocity for the prosthetic joint(s). In summary, the coefficients in \( K \) and \( k \) in 9 are obtained by:

\[
K = S_pCS_h^{-1},
\]

\[
k = -K \begin{bmatrix} \bar{\varphi}_h \\ \bar{\ddot{\varphi}}_h \end{bmatrix} + \begin{bmatrix} \bar{\dot{\varphi}}_p \\ \bar{\ddot{\varphi}}_p \end{bmatrix}.
\]

There could be a discrepancy between estimated velocity and the derivative of the estimated angle. Therefore, the Kalman filter was used to merge the two pieces of information for each joint. The left hip angle \( \varphi_{\text{hip},l} \) and left knee angle \( \varphi_{\text{kn},l} \) was used as observed; however, the right prosthetic knee angle \( \varphi_{\text{kn},r} \) was estimated.

During the data acquisition for the estimation of the coefficients of \( K \) and \( k \) was carried out with the 23-year-old, healthy female participant. She walked on a treadmill at a speed of 3 km/h,
as well as up and down stairs, equipped with the goniometer-gyroscope units to measure knee and hip flexion angles and velocities on both legs.

In the test phase, the CLME method controlled a C-leg prosthetic robotic leg for a 4-year-old female participant with transfemoral amputation during walking on a treadmill, as well as up and down stairs. The participant was allowed to hold on to the bars during treadmill walking and to the handrail of the stairs, respectively. Furthermore, an assisting person secured her on the stairs. The result was compared with data obtained from the amputee participant using the C-Leg controller.

The patient was able to walk smoothly after a few minutes of practice. The author reported that the participant noticed how the left and right legs were coupled, and she also managed to alter her gait voluntarily. She was able to walk at varying velocities with balance, and correct placement of the prosthetic foot on the next step. However, in stair descent, the performance of the CLME controlled prosthesis was less satisfactory, as it did not match the participant’s smooth stair descent with her C-Leg. The results are shown in Figure 18.

![Figure 17: Human gait data estimation of healthy participants. Image source: [141].](image)

![Figure 18: Knee angle trajectories during treadmill walking with the C-Leg (A) and with a CLME-controlled active knee joint (B). Sound and prosthetic knee joint angles are normalized. Image source: [142].](image)
3 Overview of the GaIn system

This chapter provides an overall overview of the GaIn control system. The system’s specification, aims, objectives, and main results are discussed there. The GaIn system provides a method for controlling robotic prosthetic legs for patients suffering at most double transfemoral amputation. The concept of this idea is illustrated in Figure 2. The Author’s idea is based on the observation that the correlation between the movements of the leg parts of healthy people — people without functional gait disorder during usual activities — is high. Consequently, it is possible to infer the movements of both lower legs based on the movements of both thighs using machine learning methods. The GaIn system employs a recurrent neural network with long-short-memory (LSTM) cells to infer the shank movement during walking-related activities. The Author believes that a recurrent neural network is a suitable mathematical model to simulate the motor cortex of the human nervous system.

Sitting down and standing up intentions cannot be recognized from thigh movements. In fact, these sitting-related activities drive thigh movements. Therefore, GaIn applies EMG sensors placed on the skin over the vastus lateralis the thigh muscles; therefore, the patient can signal her/his sitting down or standing up intentions by increasing thigh muscle activity.

The GaIn control system takes the input data from triaxial accelerometer and gyroscope sensors and EMG sensors located on both thighs. The GaIn control system consists of two major parts: (i) a high-level controller, which recognizes the user’s current or the intended activity, and (ii) a mid-level controller, which performs the appropriate actions: sitting down, standing up, or gait inference.

The high-level controller orchestrates the mid-level controller via the following rules:

- When the user is sitting, then the high-level controller does not allow the mid-level controller to infer gait, and both legs remain motionless. If an adequate amount of electrical activity from both thigh muscles is recognized by the high-level controller based on data from the EMG sensors, then the mid-level controller system performs the standing up procedure.

- When the user is standing, then the high-level controller can (i) keep the user in a standing position, (ii) get the mid-level controller to start gait inference if one leg starts swinging, or (iii) get the mid-level controller to perform a sitting down procedure if the electrical activity of both thigh muscles is suddenly high and both thighs have a similar position.

- When the user is walking, running, or taking the stairs, then the high-level controller keeps the mid-level controller performing gait inference using a recurrent neural network or perform stopping and standing position.

- When the user suddenly stops, then the high-level controller commands the mid-level controller to stop the procedure and keeps the patient in a standing position.

Figure 19 shows the possible transitions between different activities. For instance, if the user is walking, then the system cannot perform a sitting down activity without first stopping and
standing. When the user is sitting, then GaIn cannot infer walking-related activities without first standing up and standing.

![Activity transition graph of the GaIn control system.](image)

Incorrect activity mode or intention recognition may pose safety issues. When the GaIn system incorrectly recognizes a standing up activity while the user is sitting, then the system simply stretches the robotic prosthetic leg, resulting in no harm to the patient. However, when a sitting down intention is predicted while the user is simply standing, then the patient will fall and may suffer serious injury. Thus, it is more important to achieve low false alarm (high precision) than low missed alarm (high recall) rates for the sitting down activity. Therefore, the decision threshold was calibrated so that the activity recognition module achieved a low false alarm rate (precision) at the expense of high missed alarm (recall). This is a trade-off between precision and recall. As a consequence, users may need to produce clearer and longer signals to the system for sitting down, but the Author expects that this will result in causing fewer injuries from falling.

### 3.1 GaIn system design principles and objectives

The GaIn system could potentially be installed on microchip-controlled robotic leg prostheses that could be attached to patients in a non-invasive way to infer the movements of the lower limbs, as illustrated in Figure 2. Therefore, the GaIn system could help patients suffering single or double lower limb amputation to move and walk by themselves. However, in order to make GaIn efficient in portable real-time prediction systems, it should meet the following requirements:

1. **Low prediction latency.** GaIn should respond quickly to sudden changes in user behavior in real-time.

2. **Fast and energy-efficient.** In order to be suitable for mobile and portable systems, GaIn is to be energy-efficient and CPU-friendly.

3. **Smooth recognition.** GaIn should provide consistent recognition within a given activity mode and rapid transition in-between activity modes.

4. **Generalization.** GaIn should be accurate for new patients whose data was not seen during training.

**Low prediction latency.** The first requirement ensures that the model has low latency; therefore, activity prediction can be made instantly based on the latest observed data. Therefore, bidirectional models, such as bidirectional LSTM RNN [91] or dynamic time warping [96] methods, are not appropriate for this project aims for two main reasons. First, these bidirectional
methods require a whole observed sequence before making any predictions, which would increase their latency. Second, the prediction they make on a frame is based on subsequent data. Standard hidden Markov models have become the *de facto* approach for human activity recognition [108, 99, 92, 72], and they yield good performance in general. However, they do so at the expense of increased latency in prediction, because Viterbi algorithms use the whole sequence, or at least some part of it, to estimate a series of activities (i.e., hidden states), and their time complexity is polynomial. Therefore, HMMs are not adequate for on-the-fly prediction, because the latency of these methods can be considered rather high.

**Fast and energy efficient.** Continuous sensing and evaluating by CPU-intensive prediction methods rapidly deplete a mobile system’s energy. Therefore, the second point requires the system to be energy-efficient enough for mobile-pervasive technologies. Several approaches have been introduced for this problem. Some methods aim to keep the number of necessary sensors low by adaptive selection [162] or based on the activity performed [49, 157, 87] for accurate activity prediction. Other approaches aim to reduce the computational cost by feature selection [4], feature learning [117], or proposing computationally inexpensive prediction models such as C4.5, random forest [101], or decision trees [130].

**Smooth recognition.** This point is to ensure that the activity recognition method provides consistent prediction within the same activity mode, but changes rapidly when the activity mode has changed. Lester et al. [92] pointed out that single-frame prediction methods such as decision stumps or support vector machines are prone to yielding scattered predictions. However, human activity data are time series data in nature, and subsequent data frames are highly correlated. This tremendous amount of information can be exploited simply by sequential models such as HMM and RNN, or by incorporating the sliding-window technique in single-frame methods (e.g., nearest-neighbor). In fact, Mannini and his co-authors [99] pointed out that the continuous-emissions HMM-based sequential classifier performs systematically better than its simple single-frame GMM counterpart (99.1% vs. 92.2% accuracy). Actually, the proposed sequential classifier beats its tested single-frame competitors overall (the best single-frame classifier is the nearest mean classifier, which achieves up to 98.5% accuracy). This highlights the relevance of exploiting the statistical correlation from human dynamics.

**Generalization to new users.** People walk in different ways, and thus, human gait cycles vary. Figure 20 shows the variance of various gaits for the same person (A) and for various people (B). Therefore, it is essential that machine learning systems be robust against this natural variance and their performance be comprehensively evaluated with new patients whose data have not been seen during the training. Unfortunately, the Author observed that many HAR methods in the scientific literature have been validated with mixed data, i.e., training and testing datasets are distinct but contains data from the same users. Thus, those evaluations are not adequate to assess how their systems would perform with new users. The performance of the GaIn control system was evaluated using a supervised cross-validation approach [81]. In this approach, data from a
designated participant were held out for tests, and the rest of the data from the other participants were used for training. Thus, this approach gives a reliable estimation of how well the GaIn system would perform for a new patient whose data have not been seen before. The GaIn control system was optimized so that it performed the best on the new patient’s data.

Finally, note that, when the HuGaDB data were recorded a few years ago, some variance in sensors, installation was allowed. The location and the orientation of the sensors were not precisely regulated on purpose. This provided some variance in the data, and this makes the GaIn system more robust to sensor installation in practice and provides better generalization for the machine learning algorithms. This also will give the patients more freedom in putting the sensors on.

![Figure 20: Gait cycle variance during walking.](image)

(A) Single user  
(B) Various users

Figure 20: Gait cycle variance during walking. (A) Gait cycles produced by the same user multiple times. (B) Gait cycles produced by different users. Legend indicates the source of the data. Data are scaled to the range $[-1, +1]$. Data are taken from the HuGaDB dataset [26].

### 3.2 Main results

The Author’s overall results of gait inference using high- and mid-level controllers can be seen in a video at [https://youtu.be/aTeYPGxncnA](https://youtu.be/aTeYPGxncnA), from which two screenshots are shown in Figure 21. Many videos have been generated based on different data recorded from different participants, but the Author did not see visually notable differences in the videos. The reason this particular example was chosen is that these data contain a variety of activities during a relatively short time.

![Figure 21: Screenshot of GaIn during gait inference.](image)

Figure 21: Screenshot of GaIn during gait inference. Around 56 data frames add up to 1 second. See the full video at: [https://youtu.be/aTeYPGxncnA](https://youtu.be/aTeYPGxncnA).

Figure 22 shows the inference for a continuous series of standing up, sitting down, and a few walking-related activities. This figure consists of two parts: the first part (from 4.2 seconds to
20 seconds) shows sitting-related predictions by the high-level controller, while the second part (from 21 seconds) shows shank movement inference by the mid-level controller for walking-related activities. In the first part, the high-level controller is supposed to predict sitting-related activity modes. The activity modes are indicated by the color of the background. The true modes are shown at the bottom, while the predicted activity modes are shown at the top. Here, the activity modes were recognized correctly albeit with around a 1-second lag. The activity modes are recognized mainly based on the variance in the EMG signals (solid dark and light green lines) and the position of the shanks are irrelevant here. It should be noted that the length of the sitting down and standing up activities in the figures is irrelevant here because the length would depend on how the robotic prosthetic legs performed these movements once the patient’s intention was recognized.

![Figure 22: Gait inference and activity recognition using GaIn.](image_url)
4 HuGaDB: A human gait database for gait inference

The Author created his own human gait database, called HuGaDB, for training and evaluating machine learning methods for GaIn because current datasets were found to be unsuitable for GaIn. The reader can find a review of the current HAR datasets in section 2. In general, current datasets for HAR collected data from sensors mounted on one leg and/or on the hips, arms, etc. Therefore, they are not adequate for studying how the parts of the legs move relatively to each other within and in-between different activity modes. The database is available free of charge at https://github.com/romanchereshnev/HuGaDB.

4.1 Motivation and design goals

The main purpose of this dataset is to provide detailed gait data to study how the parts of the legs move individually and relative to each other during activities such as walking, running, standing up, etc. A summary of the activities can be found in Table 3. These activity modes comprise the most common walking related activities. The Author did not include specific leg movements such as kicks or dance moves. This dataset contains continuous recordings of combinations of activities, and the data are segmented and annotated with the label of the activity currently performed. Thus, this dataset is also suitable for analyzing human gait and activities between transitions.

Mainly, inertial sensors were used for data acquisition. In total, six inertial sensors were placed on the right and left thighs, shins, and feet, and data were collected from 18 healthy participants, providing a total of 10 hours of recording. This allowed the Author to investigate how the parts of the legs move individually and relative to each other within and in-between activities. This dataset could be used as control data, for instance, in healthcare-related studies, such as walking rehabilitation or Parkinson’s disease recognition, as well. In virtual reality or gaming, this dataset could also be used to model virtual human movements by reproducing the leg movements from the accelerometer data by simply taking the integrals. In fact, it is not limited to a virtual environment and could be used to train to walk and move humanoid robots to make them more human-like and escape the uncanny valley.

This dataset is unique in the sense that it is the first to provide human gait data in great detail mainly from inertial sensors and contains segmented annotations for studying the transition between different activities.

4.2 Sensor network topology

In data collection, MPU9250 inertial sensors and electromyography sensors created in the Laboratory of Applied Cybernetic Systems at BiTronics Lab (www.bitronicslab.com), Moscow Institute of Physics and Technology, were used. Each EMG sensor has a voltage gain is about 5000 and band-pass filter with bandwidth corresponding to power spectrum of EMG (10-500 Hz). A sample rate of each EMG-channel is 1.0 kHz, analog-to-digital resolution is 8 bits, input voltages: 0 - 5
Table 3: Characteristics of HuGaDB.

<table>
<thead>
<tr>
<th>ID</th>
<th>Activity</th>
<th>Time sec (min)</th>
<th>Percent</th>
<th>Samples</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Walking</td>
<td>11544 (192)</td>
<td>32.15</td>
<td>679073</td>
<td>Walking and turning at various speeds on a flat surface</td>
</tr>
<tr>
<td>2</td>
<td>Running</td>
<td>1218 (20)</td>
<td>3.39</td>
<td>71653</td>
<td>Running at various paces</td>
</tr>
<tr>
<td>3</td>
<td>Going up</td>
<td>2237 (37)</td>
<td>6.23</td>
<td>131604</td>
<td>Taking stairs up at various speeds</td>
</tr>
<tr>
<td>4</td>
<td>Going down</td>
<td>1982 (33)</td>
<td>5.52</td>
<td>116637</td>
<td>Taking the stairs down at various speeds and steps</td>
</tr>
<tr>
<td>5</td>
<td>Sitting</td>
<td>4111 (68)</td>
<td>11.45</td>
<td>241849</td>
<td>Sitting on a chair; sitting on the floor not included</td>
</tr>
<tr>
<td>6</td>
<td>Sitting down</td>
<td>409 (6)</td>
<td>1.14</td>
<td>24112</td>
<td>Sitting on a chair; sitting down on the floor not included</td>
</tr>
<tr>
<td>7</td>
<td>Standing up</td>
<td>380 (6)</td>
<td>1.06</td>
<td>22373</td>
<td>Standing up from a chair</td>
</tr>
<tr>
<td>8</td>
<td>Standing</td>
<td>5587 (93)</td>
<td>15.56</td>
<td>328655</td>
<td>Static standing on a solid surface</td>
</tr>
<tr>
<td>9</td>
<td>Bicycling</td>
<td>2661 (44)</td>
<td>7.14</td>
<td>156560</td>
<td>Typical bicycling</td>
</tr>
<tr>
<td>10</td>
<td>Up by elevator</td>
<td>1515 (25)</td>
<td>4.22</td>
<td>89144</td>
<td>Standing in an elevator while moving up</td>
</tr>
<tr>
<td>11</td>
<td>Down by elevator</td>
<td>1185 (19)</td>
<td>3.30</td>
<td>69729</td>
<td>Standing in an elevator while moving down</td>
</tr>
<tr>
<td>12</td>
<td>Sitting in car</td>
<td>3069 (51)</td>
<td>8.55</td>
<td>180573</td>
<td>Sitting while an travelling by car as a passenger</td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td>35903</td>
<td>598</td>
<td>2111962</td>
<td></td>
</tr>
</tbody>
</table>

Volts. The inertial sensors consisted of a 3-axis accelerometer and a 3-axis gyroscope integrated into a single chip. Data were collected with accelerometer’s range equal to ±2g with sensitivity 16.384 least significant bit (LSB)/g and gyroscope’s range equal to ±200°/s with sensitivity 16.4 LSB /°/s.

Accelerometer and gyroscope signals were stored in int16 format. EMG signals are stored in uint8. Therefore, accelerometer data can be converted to m/s² by dividing raw data 32768 and multiplying it by 2g. Raw gyroscope data can be converted to °/s by multiplying it by 2000/32768. Raw EMG data can be converted to Volts by multiplying it 0.001/255. The raw data were kept in collection in case one prefers other normalization techniques.

In total, three pairs of inertial sensors and one pair of EMG sensors were installed symmetrically on the right and left legs with elastic bands. A pair of inertial sensors were installed on the rectus femoris muscle 5 centimetres above the knee, a pair of sensors around the middle of the shinbone at the level where the calf ends, and a pair on the feet on the metatarsal bones. Two EMG sensors were placed on vastus lateralis and connected to the skin with three electrodes. The locations of the sensors are shown in Figure 23. In total, 38 signals were collected, 36 from the inertial sensors and 2 from the EMG sensors. A summary of the input raw features can be found in Table 4.

The sensors were connected through wires with each other and to a microcontroller box, which contained an Arduino Nano electronics platform with a Bluetooth module. The microcontroller collected 56.3500 samples per second on average with standard deviation (STD) 3.2057 and then transmitted them to a laptop through a Bluetooth connection.

4.3 Data acquisition programs

The Author implemented four program modules in C# programming language to collect data from the Arduino micro-controller: LowerLimbActivityDriver, CollectionCountinuesGUI, LowerLimbActivityTesting, and Editor. These programs are available for free at https://github.com/romanchereshnev/HuGaDBCollection.

The LowerLimbActivityDriver module is an interface to communicate with Arduino Bluetooth at low level. This library reads the sensors data from the Arduino device via Bluetooth connection.
Table 4: HuGaDB raw signals.

<table>
<thead>
<tr>
<th>Feature number</th>
<th>Sensor type</th>
<th>Position</th>
<th>Axis</th>
<th>Units</th>
<th>Symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Accelerometer</td>
<td>Right foot</td>
<td>X</td>
<td>$\frac{m}{s^2}$</td>
<td>$a_{(r,f,x)}$</td>
</tr>
<tr>
<td>1</td>
<td>Accelerometer</td>
<td>Right foot</td>
<td>Y</td>
<td>$m$</td>
<td>$a_{(r,f,y)}$</td>
</tr>
<tr>
<td>2</td>
<td>Accelerometer</td>
<td>Right foot</td>
<td>Z</td>
<td>$\frac{degree}{s}$</td>
<td>$a_{(r,f,z)}$</td>
</tr>
<tr>
<td>3</td>
<td>Gyroscope</td>
<td>Right foot</td>
<td>X</td>
<td>$\frac{degree}{s}$</td>
<td>$\omega_{(r,f,x)}$</td>
</tr>
<tr>
<td>4</td>
<td>Gyroscope</td>
<td>Right foot</td>
<td>Y</td>
<td>$\frac{degree}{s}$</td>
<td>$\omega_{(r,f,y)}$</td>
</tr>
<tr>
<td>5</td>
<td>Gyroscope</td>
<td>Right foot</td>
<td>Z</td>
<td>$\frac{degree}{s}$</td>
<td>$\omega_{(r,f,z)}$</td>
</tr>
<tr>
<td>6</td>
<td>Accelerometer</td>
<td>Right shank</td>
<td>X</td>
<td>$\frac{m}{s^2}$</td>
<td>$a_{(r,s,x)}$</td>
</tr>
<tr>
<td>7</td>
<td>Accelerometer</td>
<td>Right shank</td>
<td>Y</td>
<td>$\frac{m}{s^2}$</td>
<td>$a_{(r,s,y)}$</td>
</tr>
<tr>
<td>8</td>
<td>Accelerometer</td>
<td>Right shank</td>
<td>Z</td>
<td>$\frac{m}{s^2}$</td>
<td>$a_{(r,s,z)}$</td>
</tr>
<tr>
<td>9</td>
<td>Gyroscope</td>
<td>Right shank</td>
<td>X</td>
<td>$\frac{degree}{s}$</td>
<td>$\omega_{(r,s,x)}$</td>
</tr>
<tr>
<td>10</td>
<td>Gyroscope</td>
<td>Right shank</td>
<td>Y</td>
<td>$\frac{degree}{s}$</td>
<td>$\omega_{(r,s,y)}$</td>
</tr>
<tr>
<td>11</td>
<td>Gyroscope</td>
<td>Right shank</td>
<td>Z</td>
<td>$\frac{degree}{s}$</td>
<td>$\omega_{(r,s,z)}$</td>
</tr>
<tr>
<td>12</td>
<td>Accelerometer</td>
<td>Right thigh</td>
<td>X</td>
<td>$\frac{m}{s^2}$</td>
<td>$a_{(r,t,x)}$</td>
</tr>
<tr>
<td>13</td>
<td>Accelerometer</td>
<td>Right thigh</td>
<td>Y</td>
<td>$\frac{m}{s^2}$</td>
<td>$a_{(r,t,y)}$</td>
</tr>
<tr>
<td>14</td>
<td>Accelerometer</td>
<td>Right thigh</td>
<td>Z</td>
<td>$\frac{m}{s^2}$</td>
<td>$a_{(r,t,z)}$</td>
</tr>
<tr>
<td>15</td>
<td>Gyroscope</td>
<td>Right thigh</td>
<td>X</td>
<td>$\frac{degree}{s}$</td>
<td>$\omega_{(r,t,x)}$</td>
</tr>
<tr>
<td>16</td>
<td>Gyroscope</td>
<td>Right thigh</td>
<td>Y</td>
<td>$\frac{degree}{s}$</td>
<td>$\omega_{(r,t,y)}$</td>
</tr>
<tr>
<td>17</td>
<td>Gyroscope</td>
<td>Right thigh</td>
<td>Z</td>
<td>$\frac{degree}{s}$</td>
<td>$\omega_{(r,t,z)}$</td>
</tr>
<tr>
<td>18</td>
<td>Accelerometer</td>
<td>Left foot</td>
<td>X</td>
<td>$\frac{m}{s^2}$</td>
<td>$a_{(l,f,x)}$</td>
</tr>
<tr>
<td>19</td>
<td>Accelerometer</td>
<td>Left foot</td>
<td>Y</td>
<td>$\frac{m}{s^2}$</td>
<td>$a_{(l,f,y)}$</td>
</tr>
<tr>
<td>20</td>
<td>Accelerometer</td>
<td>Left foot</td>
<td>Z</td>
<td>$\frac{m}{s^2}$</td>
<td>$a_{(l,f,z)}$</td>
</tr>
<tr>
<td>21</td>
<td>Gyroscope</td>
<td>Left foot</td>
<td>X</td>
<td>$\frac{degree}{s}$</td>
<td>$\omega_{(l,f,x)}$</td>
</tr>
<tr>
<td>22</td>
<td>Gyroscope</td>
<td>Left foot</td>
<td>Y</td>
<td>$\frac{degree}{s}$</td>
<td>$\omega_{(l,f,y)}$</td>
</tr>
<tr>
<td>23</td>
<td>Gyroscope</td>
<td>Left foot</td>
<td>Z</td>
<td>$\frac{degree}{s}$</td>
<td>$\omega_{(l,f,z)}$</td>
</tr>
<tr>
<td>24</td>
<td>Accelerometer</td>
<td>Left shank</td>
<td>X</td>
<td>$\frac{m}{s^2}$</td>
<td>$a_{(l,s,x)}$</td>
</tr>
<tr>
<td>25</td>
<td>Accelerometer</td>
<td>Left shank</td>
<td>Y</td>
<td>$\frac{m}{s^2}$</td>
<td>$a_{(l,s,y)}$</td>
</tr>
<tr>
<td>26</td>
<td>Accelerometer</td>
<td>Left shank</td>
<td>Z</td>
<td>$\frac{m}{s^2}$</td>
<td>$a_{(l,s,z)}$</td>
</tr>
<tr>
<td>27</td>
<td>Gyroscope</td>
<td>Left shank</td>
<td>X</td>
<td>$\frac{degree}{s}$</td>
<td>$\omega_{(l,s,x)}$</td>
</tr>
<tr>
<td>28</td>
<td>Gyroscope</td>
<td>Left shank</td>
<td>Y</td>
<td>$\frac{degree}{s}$</td>
<td>$\omega_{(l,s,y)}$</td>
</tr>
<tr>
<td>29</td>
<td>Gyroscope</td>
<td>Left shank</td>
<td>Z</td>
<td>$\frac{degree}{s}$</td>
<td>$\omega_{(l,s,z)}$</td>
</tr>
<tr>
<td>30</td>
<td>Accelerometer</td>
<td>Left thigh</td>
<td>X</td>
<td>$\frac{m}{s^2}$</td>
<td>$a_{(l,t,x)}$</td>
</tr>
<tr>
<td>31</td>
<td>Accelerometer</td>
<td>Left thigh</td>
<td>Y</td>
<td>$\frac{m}{s^2}$</td>
<td>$a_{(l,t,y)}$</td>
</tr>
<tr>
<td>32</td>
<td>Accelerometer</td>
<td>Left thigh</td>
<td>Z</td>
<td>$\frac{m}{s^2}$</td>
<td>$a_{(l,t,z)}$</td>
</tr>
<tr>
<td>33</td>
<td>Gyroscope</td>
<td>Left thigh</td>
<td>X</td>
<td>$\frac{degree}{s}$</td>
<td>$\omega_{(l,t,x)}$</td>
</tr>
<tr>
<td>34</td>
<td>Gyroscope</td>
<td>Left thigh</td>
<td>Y</td>
<td>$\frac{degree}{s}$</td>
<td>$\omega_{(l,t,y)}$</td>
</tr>
<tr>
<td>35</td>
<td>Gyroscope</td>
<td>Left thigh</td>
<td>Z</td>
<td>$\frac{degree}{s}$</td>
<td>$\omega_{(l,t,z)}$</td>
</tr>
<tr>
<td>36</td>
<td>EMG</td>
<td>Right thigh</td>
<td></td>
<td></td>
<td>$\epsilon_r$</td>
</tr>
<tr>
<td>37</td>
<td>EMG</td>
<td>Left thigh</td>
<td></td>
<td></td>
<td>$\epsilon_l$</td>
</tr>
</tbody>
</table>

This driver is used in the LowerLimbActivityTesting and CollectionCoutninesGUI programs.

CollectionCoutninesGUI is the main annotating tool. It contains a graphical user interface, where the experimenter can annotate the data being collected. Then experimenter provides some meta-information, like accelerometer resolution and participant ID number. After that, the experimenter selects the activity mode and may start to collect data. During data collection, experimenter presses the button indicating the current recorded activity.
Figure 23: Location of sensors. (A) EMG sensors are shown as circles, while boxes represent inertial sensors. (B) Mounted sensors. The black waist bag contains the Arduino driver.

When the experimenter pushes the “Stop” button, the program stops collecting data and checks the data quality. This is needed because wires can detach from sensors during data collection. In this situation, the Arduino device sends zeros instead of useful warning information. After recording stops, CollectionCountinuesGUI automatically checks if the data collection was interrupted during collection and informs the experimenter. Finally, the CollectionCountinuesGUI stores the data in raw format. The Editor program can read these raw files and converts them into HuGaDB format, which will be described in section 4.5.

The LowerLimbActivityTesting program was created to visualize the data recorded from the sensors in order to check for errors. LowerLimbActivityTesting helps to understand if there were any problems with connection to the Arduino device or with the wire connections between sensors.

4.4 Participants

The data were collected from 18 participants. These participants were healthy young adults: 4 females and 14 males, average age of 23.67 (STD: 3.69) years, an average height of 179.06 (STD: 9.85) cm, and an average weight of 73.44 (STD: 16.67) kg. Characteristics of participants are shown in Table 5.

The participants performed a combination of activities at different speeds but casual way, and there were no obstacles placed on their way. The experimenter recorded the data continually using a laptop and annotated the data with the activities performed. When the subject was asked to perform a trial of one specific activity, the experimenter be nearby and marked the activity performed. This provided us a long, continuous sequence of segmented data annotated
Figure 24: Screenshot of the CollectionCountinuesGUI program.

Figure 25: LowerLimbActivityTesting screenshot while data collection.

with activities. In total, 2,111,962 samples were collected from all the 18 participants, and they provided a total of 10 hours of data.

Data acquisition was carried out mainly inside a building. However, activities such as running, bicycling, and sitting in a car were performed outside. Going up and down were recorded on different stairs (in a building and outside) with different step heights. Moreover, the data were also collected in a moving elevator and a vehicle. In these scenarios, the activities performed were simply standing or sitting. However, a force impacts the accelerometer sensors, and in certain applications, it may be important to consider these facts.

During data acquisition, there were four typical scenarios performed by almost all subjects:
1. Person started from sitting position. The experimenter started recording and commanded the participant to stand up. After standing up, the participant stood around 10-30 seconds to calm the muscles. Then, the experimenter commanded the participant to sit down, and the participant sat for 10-30 seconds. This procedure was usually repeated four times with various speeds: sometimes faster or slower than usual. After performing, the subject rested for 1-2 minutes.

2. Participant started in standing position. The experimenter started recording and commanded the participant to walk. However, recording sometimes was started during walking. Walking was performed at different speeds by the participant. The experimenter commanded the participant to start and stop running. This cycle was repeated several times with different speeds. After performing, the participants rested for 1-2 minutes.

3. Participant started in standing position by the stairs. The experimenter ordered the participant to go up the stairs. The participant was asked to do this at different speeds and with different patterns, for example, the participant was asked to take two stairs at each step. Usually, the participant walked up four floors. After performing, the participant rested for 1-2 minutes.

4. Same as before, but going down the stairs.

However, data collection scenarios were not limited to those described above. For instance, a one participant was instructed to perform the following activities: starting from a sitting position, sitting – standing up – walking – going up the stairs – walking – sitting down.
Table 5: Characteristics of the HuGaDB participants.

<table>
<thead>
<tr>
<th>id</th>
<th>weight (kg)</th>
<th>height (cm)</th>
<th>age</th>
<th>sex (M=Male, F=Female)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>75</td>
<td>177</td>
<td>24</td>
<td>M</td>
</tr>
<tr>
<td>2</td>
<td>80</td>
<td>183</td>
<td>22</td>
<td>M</td>
</tr>
<tr>
<td>3</td>
<td>65</td>
<td>183</td>
<td>23</td>
<td>M</td>
</tr>
<tr>
<td>4</td>
<td>93</td>
<td>189</td>
<td>24</td>
<td>M</td>
</tr>
<tr>
<td>5</td>
<td>63</td>
<td>183</td>
<td>35</td>
<td>M</td>
</tr>
<tr>
<td>6</td>
<td>54</td>
<td>168</td>
<td>25</td>
<td>F</td>
</tr>
<tr>
<td>7</td>
<td>52</td>
<td>161</td>
<td>22</td>
<td>F</td>
</tr>
<tr>
<td>8</td>
<td>80</td>
<td>176</td>
<td>23</td>
<td>M</td>
</tr>
<tr>
<td>9</td>
<td>65</td>
<td>175</td>
<td>24</td>
<td>F</td>
</tr>
<tr>
<td>10</td>
<td>118</td>
<td>183</td>
<td>27</td>
<td>M</td>
</tr>
<tr>
<td>11</td>
<td>85</td>
<td>203</td>
<td>24</td>
<td>M</td>
</tr>
<tr>
<td>12</td>
<td>85</td>
<td>192</td>
<td>23</td>
<td>M</td>
</tr>
<tr>
<td>13</td>
<td>64</td>
<td>174</td>
<td>18</td>
<td>M</td>
</tr>
<tr>
<td>14</td>
<td>68</td>
<td>175</td>
<td>19</td>
<td>M</td>
</tr>
<tr>
<td>15</td>
<td>72</td>
<td>178</td>
<td>23</td>
<td>M</td>
</tr>
<tr>
<td>16</td>
<td>48</td>
<td>164</td>
<td>26</td>
<td>F</td>
</tr>
<tr>
<td>17</td>
<td>85</td>
<td>179</td>
<td>25</td>
<td>M</td>
</tr>
<tr>
<td>18</td>
<td>70</td>
<td>180</td>
<td>19</td>
<td>M</td>
</tr>
</tbody>
</table>

4.5 Data format

Data obtained from the sensors were stored in flat text files. The data is stored in flat text files because they are one of the most universal formats, and they can be easily preprocessed in all programming languages on every system. One data file contains one recording, which is either a single activity (e.g., walking) or a series of activities. Every file name was created according to the template HuGaDB vX ACT PR CNT.txt. HuGaDB is a prefix that means human gait data and vX means the version of the data files, currently v1. ACT is a variable, and it denotes the activity ID that was performed. If a file contains a series of different types of activities, then it is indicated as VARIOUS. PR indicates the ID of the person who performed the activity. Data recording was repeated a few times, and CNT is a counter for this. For example, a file named HuGaDB v1 walking 17 02.txt contains data from participant 17 while he was walking for the second time. The file naming convention is summarized in Table 6.

Table 6: Description of the file naming convention.

<table>
<thead>
<tr>
<th>TAG</th>
<th>Description</th>
<th>Type</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>HuGaDB</td>
<td>Prefix</td>
<td>fixed</td>
<td>Data files start with this prefix</td>
</tr>
<tr>
<td>vX</td>
<td>Version number</td>
<td>integer</td>
<td>Indicates the version of the data format</td>
</tr>
<tr>
<td>ACT</td>
<td>Activity</td>
<td>string</td>
<td>Indicates the type of activity</td>
</tr>
<tr>
<td>PR</td>
<td>Participant ID</td>
<td>integer</td>
<td>Indicates the subject who recorded the data</td>
</tr>
<tr>
<td>CNT</td>
<td>Counter</td>
<td>integer</td>
<td>Counter for repeated experiments</td>
</tr>
</tbody>
</table>

The main body of the data files contains tab-delimited raw, unnormalized data obtained from the sensors directly. Each data file starts with a header, which contains metainformation. It
summarizes the list of activities, the IDs of the activities recorded, and the time and date of the recording. This is summarized in Table 7.

The main data body of every file has 39 columns. Each column corresponds to a sensor, and one row corresponds to a sample. The order of the columns is fixed. The first 36 columns correspond to the inertial sensors, the next 2 columns correspond to the EMG sensors, and the last column contains the activity ID. The activities are coded as shown in Table 3. The inertial sensors are listed in the following order: right foot (rf), right shin (rs), right thigh (rt), left foot (lf), left shin (ls), and left thigh (lt), followed by right EMG (r) and left EMG (l). Each inertial sensor produces three acceleration data on x, y, z axes and three gyroscope data on x, y, z axes. For instance, the column named “acc_rtz” contains data obtained from the z-axis of accelerometer located on the right thigh.

Sample data with respect to the activities are visualized through a heat map representation in Figure 27. A screenshot of some part of data file can be seen in Figure 28.

### 4.6 HuGaDB issues

There are a few concerns with the HuGaDB dataset which are worth mentioning. First, the annotation was carried out by the experimenter during data collection, and the annotation boundary might be inaccurate. In some cases, the boundary can be off by around 0.3–0.4 seconds, which is roughly equal to 20-30 data frames. Moreover, it might even be impossible to determine the boundary exactly between two dynamic activity modes such as walking and going up the stairs.

However, note that the inaccuracy in annotation may serve as some variance in the data that could be considered as regularization and essentially could result in better generalization of the machine learning methods.

### 4.7 Noise

When the data were recorded, some variance in sensors, installation was allowed. The Author did not regulate very precisely the location and the orientation of the sensors on purpose. This provided some variance in the data to make the system more robust to sensor installation in practice and provide better generalization for the machine learning algorithms. This will also give the patients more freedom in putting the sensors on.

Note that data were not collected on a treadmill. The reason for this is that performing on a treadmill does not result in actual movement; therefore, accelerometer sensors do not sense actual moving overall. Moreover, walking on a treadmill would result in a very regular walk at
Figure 27: Data visualization for normalization data from initial sensors were divided by 32768 and data from EMG were subtracted by 128 and divided by 128.
Figure 28: Screenshot of a HuGaDB data file.
a constant speed on a perfectly smooth surface. However, in everyday life, a person walks on an uneven surface with small bumps and often turns. Therefore, the data would not contain natural noise, and thus, the machine learning classifier would overfit. This, in fact, was observed by Foerster and his colleagues in [42]. They used a nearest neighbor method for activity recognition based on wearable sensors. They achieved a 95.6% accuracy for ambulation activities on data that were collected in a laboratory; however, the accuracy dropped to 66% in a natural environment. This is the reason that HuGaDB data were collected in a natural environment.

4.8 Conclusions

The HuGaDB database can be used not only for activity recognition but also for studying how activities are performed and how the parts of the legs move relative to each other. Therefore, the data can be used (a) to perform healthcare-related studies, such as in walking rehabilitation or Parkinson’s disease recognition, (b) in virtual reality and gaming for simulating humanoid motion, or (c) for humanoid robotics to model humanoid walking.

HuDaDB has been downloaded several times since it was published, and the author has received several inquiries about the database from the HAR community. HuGaDB has been used in several other studies:

1. OpenHAR: A matlab toolbox for easy access to publicly open human activity data sets [129].
2. Gait Recognition via Machine Learning [80].
3. An artificial neural network framework for lower limb motion signal estimation with foot-mounted inertial sensors [134].

HuGaDB was presented at the Sixth International Conference on Analysis of Images, Social networks and Texts where the Author won the best talk award. HuGaDB article was published in Springer’s Q2 journal Lecture Notes in Computer Science [26].

4.9 Availability

The database is available free of charge at https://github.com/romanchereshnev/HuGaDB (455 Mb).
5 GaIn high-level controller

The GaIn system consists of two main parts: the high-level controller and the mid-level controller. The high-level controller is responsible for the recognition of the person’s current activity mode and intention to sit down or stand up. This is required because the gait inference is based on the movement of the thighs; however, sitting down and standing up intentions cannot be recognized from thigh movements. In fact, these sitting-related activities drive thigh movements. Therefore, GaIn employs EMG sensors placed on the skin over the vastus lateralis thigh muscles, and the patient can signal her/his sitting down or standing up intentions by increasing thigh muscle activity. The high-level controller aims to recognize these intentions.

This section introduces a new high-level controller for GaIn to predict sitting down and standing up intentions and also shows that this controller can be used for HAR tasks in general.

5.1 RapidHARe classification method

Recent HAR classification methods based on on-body inertial sensors, have achieved increasing performance; however, this is at the expense of longer CPU calculations and greater energy consumption. Therefore, these complex models might not be suitable for real-time prediction in mobile systems. In this section, a novel method called RapidHARe is presented for real-time and continuous human activity mode recognition. The proposed model is a small dynamic Bayesian network that does not utilize the Viterbi algorithm or other dynamic programming approaches for activity prediction, but instead utilizes the data distribution within a small context window. This method does not employ any dynamic-programming-based algorithms, which are notoriously slow for inference, nor does it employ feature extraction or selection methods the entire data sequence. This results in a quick method that does not require exhaustive CPU calculations. Therefore, RapidHARe is suitable for real-time recognition. Moreover, it is inexpensive for mobile systems and can be employed in elder-care support and long-term health-monitoring systems such as freeze-of-gait prediction, fall detection, robotic exoskeletons in health care, and surgery recovery.

RapidHARe is a dynamic Bayesian network whose structure is shown in Figure 29. The states, i.e., activities, denoted by $S$ and the probability of a state $s(t)$ at a given time $t$ with respect to a given observed context window $v(t), v(t-1), \ldots, v(t-K)$ of length $K$, is formulated by

$$P(s(t) \mid v(t), v(t-1), \ldots, v(t-K)) = \frac{\prod_{k=0}^{K} P(v(t-k) \mid s(t))P(s(t))}{\sum_{n \in S} \prod_{k=0}^{K} P(v(t-k) \mid s(t) = n)P(s(t) = n)}.$$  \hspace{1cm} (17)

Certainly, at the beginning of performance, when $t < K$, the context window is adjusted. RapidHARe does not use different $a \ priori$ class probabilities for different $P(s)$ for any $s \in S$. Thus, the model is not biased toward some states that are abundant in the training data. Therefore, the activity prediction should be based fully on the data, and the state probabilities $P(s)$ can be omitted from Eq. 17.
The state being performed at time $t$ can be predicted as follows:

$$
\hat{s}(t) = \arg\max_{s \in S} \{P(s \mid v(t), v(t-1), \ldots, v(t-K))\}.
$$

(18)

Since the optimum of Eq. 18 is invariant to normalization, the normalization factor can be omitted from Eq. 17. This gives a very simple model for activity prediction in the following form:

$$
\hat{s}(t) = \arg\max_{s \in S} \left\{ \prod_{k=0}^{K} P(v(t-k) \mid s) \right\}.
$$

(19)

This model can be implemented using the rolling-window technique for real-time continuous activity recognition; thus, the model remains fast for large $K$s, and redundant calculation of $P(v(t-k) \mid s(t)); (k > 0)$ can be avoided by using tables.

Figure 29: Illustration of an unfolded dynamic Bayesian network w.r.t. an activity series.

The distribution $P(V \mid S)$ with respect to a given state is modeled with Gaussian mixture models, and its parameters are trained using the expectation-maximization method. The training of GMMs was straightforward because training data were segmented.

Overall, RapidHARe is a simple and fast model that consumes little energy to recognize human activities.

### 5.2 Activity mode recognition using RapidHARe

The RapidHARe method is suitable for general HAR tasks. In this section a detailed calibration of the RepidHAR system is provided. In these experiments the HuGaDB dataset was utilized with eight classification tasks, namely, walking, running, going up, going down stairs, sitting, sitting down stairs, standing up, and standing. All 38 HuGaDB signals were used to predict a human activity. Please also note that no preprocessing step, feature extraction, or feature selection methods were used besides the feature scaling.

The first experiment was carried out to determine the values of the length of the context window and number of the Gaussian components of RapidHARe in $P(V \mid S)$ via grid search. In these tests, the covariance matrices $\Sigma$ in all Gaussian components were restricted to be diagonal. The results were evaluated in terms of accuracy and $F_1$ score and are shown in Figures 30 and 31. They indicate that a good performance can be achieved using $K = 26$ for the context window length. However, for the Gaussian components, it seems that for dynamic activities, such as
walking and running, the higher the number of Gaussian components, the better the performance. On the other hand, for static activities, such as sitting and standing, a large number of Gaussian components hinders the activity recognition. Therefore, the number of Gaussian components for $P(V | S)$ for the following activities was: walking, 18; running, 18; going up, 16; going down, 16; sitting, 2; standing up, 5; sitting down, 7; and standing, 4. The activity recognition results using these hyperparameters are shown in Table 8, and the model achieved 97.85% accuracy, 87.4% precision, 87.22% recall, and an 86.4% $F_1$ score.

Table 8: Results of activity recognition.

<table>
<thead>
<tr>
<th>Activity</th>
<th>Recall</th>
<th>Precision</th>
<th>$F_1$ score</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Walking</td>
<td>89.09</td>
<td>94.57</td>
<td>91.62</td>
<td>95.63</td>
</tr>
<tr>
<td>Running</td>
<td>96.05</td>
<td>91.52</td>
<td>93.18</td>
<td>99.32</td>
</tr>
<tr>
<td>Going up</td>
<td>92.31</td>
<td>81.23</td>
<td>85.8</td>
<td>96.37</td>
</tr>
<tr>
<td>Going down</td>
<td>89.14</td>
<td>86.25</td>
<td>87.25</td>
<td>97.24</td>
</tr>
<tr>
<td>Sitting</td>
<td>97.61</td>
<td>88.83</td>
<td>92.89</td>
<td>98.97</td>
</tr>
<tr>
<td>Sitting down</td>
<td>68.62</td>
<td>86.57</td>
<td>75.05</td>
<td>99.03</td>
</tr>
<tr>
<td>Standing up</td>
<td>72.51</td>
<td>73.25</td>
<td>70.95</td>
<td>98.76</td>
</tr>
<tr>
<td>Standing</td>
<td>93.75</td>
<td>97.26</td>
<td>95.35</td>
<td>97.66</td>
</tr>
<tr>
<td>Average</td>
<td>87.22</td>
<td>87.4</td>
<td>86.4</td>
<td>97.85</td>
</tr>
</tbody>
</table>

5.2.1 Continuous activity recognition

Next, it was examined how well RapidHARe performs on continuous activity recognition. For this reason, activity recognition was performed on a continuous series of activities. Then, the true and predicted activities were plotted on a time-line, shown in Figure 32. The results show that RapidHARe does predict continuous activities, and it does not predict scattered activities for neighboring frames except for a few frames.

However, it looks like, misclassification occurs on the borders in many cases. Furthermore, after enlarged the standing–sitting activity at 35.6 sec, as shown in Figure 33, there can be seen that RapidHARe method predicts sitting activity, at around 40.94 sec, a small fraction of a second earlier than it happened, according to the data annotation. It is unlikely that RapidHARe method
can predict the future. This phenomenon could be a result of inaccurate data segmentation made by the data controller and by the fact that it is difficult to exactly determine an activity border in 10–20 ms.

In order to mitigate this phenomenon, some border tolerance was allowed in the misclassification if it occurs on the activity border. Thus, up to 25 data frames (which is about half a second) were tolerated to be misclassified on the activity border if and only if the method correctly recognized the succeeding activity. Moreover, if there is tolerance of misclassification on the borders, then the performance measures will put an emphasis on more reliable estimation for the actual scattered misclassification made by the model, and it will be more tolerant of inaccurate data segmentation.

With tolerated misclassification on the border of up to 25 data frames, RapidHARe obtains 98.68% accuracy, 91.52% recall, 92.5% precision, and 91.34% $F_1$ on average over all activities. The detailed results for each activity are shown in Table 9.

In the rest of the experiments, border tolerance is equal to 25 data frames, unless otherwise specified.
Figure 33: Activity recognition at 35.6 s enlarged from Figure 30. The line represents the x-axis acceleration value recorded by accelerometer located on thigh.

Table 9: Continuous activity recognition allowing border tolerance.

<table>
<thead>
<tr>
<th></th>
<th>Walking</th>
<th>Running</th>
<th>Going up</th>
<th>Going down</th>
<th>Sitting</th>
<th>Sitting down</th>
<th>Standing up</th>
<th>Standing</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recall</td>
<td>94.59</td>
<td>97.49</td>
<td>94.77</td>
<td>91.58</td>
<td>98.9</td>
<td>74.18</td>
<td>85.25</td>
<td>96.27</td>
<td>91.52</td>
</tr>
<tr>
<td>Precision</td>
<td>96.84</td>
<td>93.21</td>
<td>87.03</td>
<td>92.89</td>
<td>92.29</td>
<td>90.86</td>
<td>88.66</td>
<td>98.17</td>
<td>92.5</td>
</tr>
<tr>
<td>$F_1$ score</td>
<td>95.61</td>
<td>94.88</td>
<td>90.17</td>
<td>91.86</td>
<td>95.38</td>
<td>80.47</td>
<td>85.69</td>
<td>97.12</td>
<td>91.34</td>
</tr>
<tr>
<td>Accuracy</td>
<td>97.66</td>
<td>99.5</td>
<td>97.54</td>
<td>98.29</td>
<td>99.33</td>
<td>99.24</td>
<td>99.44</td>
<td>98.48</td>
<td>98.68</td>
</tr>
</tbody>
</table>

5.2.2 Directional features

Examining the results in Table 9 show that the recognition performance of sitting down and standing up activities are relatively poor compared to other activities. To investigate the problem, the data recorded with a 3D accelerometer sensor located on the left thigh during standing, sitting, standing up, and sitting down activities were plotted. Data are shown in Figure 34. The figure reveals that data from static activities are precisely concentrated on counter sides, but the data from dynamic activities lay on top of each other and in-between the static activities. Therefore, it is difficult to distinguish the two dynamic activities. Thus, additional features were created to indicate changes in signal data. For a signal datum $a(t)$ at time $t$ from $x$ and $z$-axis accelerometer sensors located on both thighs, four additional features were created as $d_i(t) = a_i(t) - a_i(t - l)$, called directional features, where $i = \{1, 2, 3, 4\}$ indexes the aforementioned signals, and $l$ is a lag parameter denoting time offset. For instance, if $a_i(t)$ is the signal obtained at time $t$ from the $x$-axis accelerometer sensor located on the left thigh, then $d_i(t)$ indicates how much this signal has changed since time $t - l$. Thus, four additional features are obtained. The original 38-feature-data vectors concatenated with 4-feature-data vectors $d(t)$, yielding 42 features in total for every sample. These new features add extra information about the direction of movements.

To calibrate the lag parameter, a line search was run to obtain the best results using $l = 15$, which is equivalent to approximately a third of a second (data not shown). Thus, in the rest of the tests, $l = 15$ is used for the lag parameter.

The results obtained using the directional features are shown in Table 10, and they indicate a 50–65% decline in the overall error (cf. Table 9) for the measured metrics. However, closer investigation of the sitting down and standing-up activities reveals even greater improvement. For instance, the $F_1$ score increases from 80.47% to 93.43% for sitting down and from 85.69% to 96.94% for standing-up.
Figure 34: Data from x- and z-axis accelerometer located on left thigh. Data from y-axis accelerometer were nearly constant and thus are not shown.

Table 10: Results of activity recognition with directional features.

<table>
<thead>
<tr>
<th></th>
<th>Walking</th>
<th>Running</th>
<th>Going up</th>
<th>Going down</th>
<th>Sitting</th>
<th>Sitting down</th>
<th>Standing up</th>
<th>Standing</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recall</td>
<td>96.1</td>
<td>98.31</td>
<td>95.06</td>
<td>91.41</td>
<td>99.52</td>
<td>98.32</td>
<td>95.71</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Precisions</td>
<td>97.03</td>
<td>93.43</td>
<td>91.66</td>
<td>94.44</td>
<td>96.54</td>
<td>97.4</td>
<td>99.32</td>
<td>95.88</td>
<td></td>
</tr>
<tr>
<td>F₁ scores</td>
<td>96.46</td>
<td>95.4</td>
<td>92.74</td>
<td>92.46</td>
<td>98.0</td>
<td>96.94</td>
<td>95.88</td>
<td>95.55</td>
<td></td>
</tr>
<tr>
<td>Accuracy</td>
<td>98.13</td>
<td>99.58</td>
<td>98.33</td>
<td>98.49</td>
<td>99.72</td>
<td>99.86</td>
<td>99.41</td>
<td>99.15</td>
<td></td>
</tr>
</tbody>
</table>

5.2.3 State-of-the-art methods

Here, the HAR state-of-the-art methods that were used in comparative tests are introduced. Also, the experimental results of the grid search used to find the best hyperparameter settings are provided. The following methods were used: hidden Markov model, artificial neural network, and recurrent neural network.

In the HMM, the data emission probabilities were modeled with Gaussian mixture models. Initial state probabilities were equally 0.125. The state transition probability matrix is shown in Table 11. Between certain activities, the transition probabilities are set to zero to prohibit absurd transitions. For instance, a sitting cannot be followed by running without first standing up. The transition matrix was calibrated manually because HMM can prefer states based on a priori information obtained from the training data.

Table 11: Transition matrix for hidden Markov model.

<table>
<thead>
<tr>
<th></th>
<th>Walking</th>
<th>Running</th>
<th>Going up</th>
<th>Going down</th>
<th>Sitting</th>
<th>Sitting down</th>
<th>Standing up</th>
<th>Standing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Walking</td>
<td>0.99</td>
<td>0.0025</td>
<td>0.0025</td>
<td>0.0025</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0025</td>
</tr>
<tr>
<td>Running</td>
<td>0.0025</td>
<td>0.99</td>
<td>0.0025</td>
<td>0.0025</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0025</td>
</tr>
<tr>
<td>Going up</td>
<td>0.005</td>
<td>0</td>
<td>0.99</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.005</td>
</tr>
<tr>
<td>Going down</td>
<td>0.005</td>
<td>0</td>
<td>0</td>
<td>0.99</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.005</td>
</tr>
<tr>
<td>Sitting</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.99</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.01</td>
</tr>
<tr>
<td>Sitting down</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.01</td>
<td>0.99</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Standing up</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.99</td>
<td>0</td>
<td>0.01</td>
</tr>
<tr>
<td>Standing</td>
<td>0.002</td>
<td>0.002</td>
<td>0.002</td>
<td>0.002</td>
<td>0</td>
<td>0.002</td>
<td>0</td>
<td>0.99</td>
</tr>
</tbody>
</table>

A grid search was run on the number of GMM components vs. the window length used in the Viterbi algorithm in order to find the best hyperparameters. Parameters were initialized randomly,
and tests were repeated five times. The averaged results (along with the standard deviations (STD) in parentheses) are shown in Table 12. Results indicate that the best accuracy can be achieved using 30 Gaussian components with 50 data frames passed to the Viterbi algorithm. In experiments with HMMs, the same number of GMM components were used as for the RapidHARe for two reasons: First, this gives better performance with HMM, and second, the prediction speeds of HMM and RapidHARe become comparable. The choice of the window length is also critical. Long windows result in large lag times in prediction. Because the sampling rate is around 56 samples per seconds, the main drawback of long window length is that the system has to wait a long time to collect the adequate number of data samples before prediction. For instance, a window length 50 results in almost a 1 s lag time before any prediction can be made. However, the advantage of long windows is that the prediction can be made for a bigger data chunk, which reduces the prediction time per sample. In experiments, the window length was equal to 10 because this proved to be the best trade-off between accuracy and speed. Fewer data yielded worse accuracy, while longer blocks increased the prediction latency.

### Table 12: HMM grid search result.

<table>
<thead>
<tr>
<th>#GMM</th>
<th>Window length</th>
<th>Without Border Tolerance</th>
<th>With Border Tolerance</th>
<th>#Params</th>
<th>Time(µs)$^1$</th>
<th>Lag(s)$^2$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>$F_1$ Accuracy</td>
<td>$F_1$ Accuracy</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>50</td>
<td>77.82 (1.02) 96.37 (0.30)</td>
<td>80.57 (1.08) 96.88 (0.31)</td>
<td>18240</td>
<td>34.87 (0.04)</td>
<td>0.85</td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>77.56 (1.00) 96.32 (0.30)</td>
<td>80.43 (1.05) 96.85 (0.31)</td>
<td></td>
<td>54.10 (0.71)</td>
<td>0.43</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>76.64 (0.95) 96.17 (0.30)</td>
<td>79.36 (0.98) 96.68 (0.30)</td>
<td></td>
<td>114.05 (0.23)</td>
<td>0.17</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>75.78 (0.96) 95.99 (0.30)</td>
<td>78.36 (1.00) 96.47 (0.30)</td>
<td></td>
<td>213.46 (0.28)</td>
<td>0.09</td>
</tr>
<tr>
<td>20</td>
<td>50</td>
<td>77.33 (0.38) 96.03 (0.04)</td>
<td>80.04 (0.34) 96.53 (0.05)</td>
<td>12160</td>
<td>31.39 (0.40)</td>
<td>0.85</td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>76.93 (0.39) 95.96 (0.04)</td>
<td>79.75 (0.36) 96.47 (0.05)</td>
<td></td>
<td>47.70 (0.04)</td>
<td>0.43</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>76.15 (0.32) 95.80 (0.03)</td>
<td>78.86 (0.28) 96.29 (0.04)</td>
<td></td>
<td>105.59 (0.21)</td>
<td>0.17</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>75.19 (0.27) 95.57 (0.03)</td>
<td>77.73 (0.25) 96.03 (0.05)</td>
<td></td>
<td>195.65 (0.06)</td>
<td>0.99</td>
</tr>
<tr>
<td>10</td>
<td>50</td>
<td>76.87 (0.74) 95.78 (0.10)</td>
<td>79.58 (0.71) 96.22 (0.09)</td>
<td>6080</td>
<td>23.92 (0.05)</td>
<td>0.85</td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>76.44 (0.77) 95.73 (0.11)</td>
<td>79.22 (0.74) 96.16 (0.09)</td>
<td></td>
<td>42.01 (0.03)</td>
<td>0.43</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>75.57 (0.72) 95.59 (0.12)</td>
<td>78.27 (0.66) 96.01 (0.10)</td>
<td></td>
<td>93.41 (0.21)</td>
<td>0.17</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>74.81 (0.73) 95.41 (0.11)</td>
<td>77.36 (0.68) 95.81 (0.10)</td>
<td></td>
<td>179.37 (0.06)</td>
<td>0.99</td>
</tr>
<tr>
<td>5</td>
<td>50</td>
<td>75.28 (0.23) 94.87 (0.11)</td>
<td>77.92 (0.24) 95.29 (0.11)</td>
<td>3040</td>
<td>21.42 (0.11)</td>
<td>0.85</td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>74.68 (0.20) 94.79 (0.09)</td>
<td>77.21 (0.21) 95.20 (0.10)</td>
<td></td>
<td>39.04 (0.05)</td>
<td>0.43</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>73.28 (0.15) 94.57 (0.07)</td>
<td>75.64 (0.16) 94.94 (0.08)</td>
<td></td>
<td>88.38 (0.01)</td>
<td>0.17</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>72.31 (0.12) 94.30 (0.01)</td>
<td>74.52 (0.11) 94.63 (0.00)</td>
<td></td>
<td>171.12 (0.50)</td>
<td>0.99</td>
</tr>
</tbody>
</table>

Tests were repeated five times; mean results are shown along with STD in parentheses. Performance measures are averaged over activities. $^1$The number of parameters in the models to be trained. $^2$Time in micro seconds to predict the activity of a single data frame measured on a single-thread CPU. $^3$Time in seconds to wait to collect an adequate number of data samples.

A grid search was used over the following hyperparameters – (1) number of hidden units within a layer from 10 to 400; (2) number of hidden layers: 1 or 2; and (3) activation function: sigmoid or rectified linear unit (ReLU) – to find the best ANN architecture. The training was performed with an Adam optimizer and with early stopping. In the early stopping, the training stopped if the validation loss reduced less than 1e-6 in the last three epochs or if the average validation loss of the last 10 epochs was greater than the average validation loss of the preceding 10 epochs (that is, the cost tended to grow). An example for the learning curves along with the loss on the validation set is shown in Figure 35. Tests were repeated five times; average results are shown in Table 13, along with STD in parentheses. The results indicate that structures with the ReLU activation function performed poorly; however, two–layered structure with a sigmoid activation function
seemed to be overfit and slow in prediction. The best performance with ANN can be achieved using a single-layer network with a sigmoid activation function that has 200 hidden units, and this is the structure that was used in comparative tests.

Figure 35: Learning curve for early stopping. Training terminated after epoch 80 because of lack of improvement on the validation set.

For the best hyperparameter search for the RNN, a grid search was performed over the number of hidden units from 10 to 200 using sigmoid or ReLU activation functions. Tests were repeated five times, and the averaged results along with STD presented in Table 14. The results indicate that RNN can be considered rather slow. Moreover, ReLU seems to perform poorly compared to the sigmoid activation function. The best performance was achieved using 200 hidden units with a sigmoid activation function organized in a single layer. Thus, this is the structure for RNN that was used in comparative tests.

5.2.4 Comparison to the state-of-the-art methods

In this section, RapidHARe is compared against the state-of-the-art methods. Recognition performance was evaluated by recall, precision, $F_1$ score, and accuracy, the and main test results are summarized in Table 15. The best results were achieved using the RapidHARe method using directional features (RapidHARe-DF) and all features from all sensors when there was allowed tolerance on the border between activities. RapidHARe-DF has achieved a 94.27% $F_1$ score and 98.94% accuracy. Compared to ANN, RNN, and HMM, this decreased the $F_1$ score error rate by 46%, 66%, and 63% and the accuracy error rate by 41%, 55%, and 62%, respectively. Allowing border tolerance improves performance metrics. For instance, by allowing border tolerance, the RapidHARe-DF method reduced the $F_1$ score error rate by 52% and the accuracy error rate by 49% when compared to the case when border tolerance was not allowed. However, border tolerance for ANN, RNN, and HMM reduced the $F_1$-score error rate by 19%, 2%, and 15%, respectively, and the accuracy error rate by 15%, 2%, and 15%, respectively. This suggests that the ANN, RNN, and HMM methods tend to make more scattered misclassifications within the same activity rather than at the border between different activities.
Table 13: Artificial neural network grid search result.

<table>
<thead>
<tr>
<th>#Units</th>
<th>Without Border Tolerance</th>
<th>With Border Tolerance</th>
<th>#Params</th>
<th>Time (µs)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$F_1$ Accuracy</td>
<td>$F_1$ Accuracy</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Two hidden layers with sigmoid activation function</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>400</td>
<td>85.34 (0.27) 97.78 (0.04)</td>
<td>87.65 (0.29) 98.1 (0.04)</td>
<td>179208</td>
<td>45.27 (0.77)</td>
</tr>
<tr>
<td>300</td>
<td>85.86 (0.46) 97.82 (0.02)</td>
<td>88.23 (0.56) 98.13 (0.04)</td>
<td>104408</td>
<td>34.51 (0.4)</td>
</tr>
<tr>
<td>200</td>
<td>85.45 (0.41) 97.8 (0.04)</td>
<td>87.77 (0.34) 98.11 (0.03)</td>
<td>49608</td>
<td>24.03 (0.11)</td>
</tr>
<tr>
<td>100</td>
<td>86.4 (0.09) 97.84 (0.01)</td>
<td>88.82 (0.14) 98.16 (0.02)</td>
<td>14808</td>
<td>13.51 (0.24)</td>
</tr>
<tr>
<td>50</td>
<td>86.5 (0.63) 97.77 (0.08)</td>
<td>88.93 (0.68) 98.08 (0.08)</td>
<td>4908</td>
<td>8.35 (0.31)</td>
</tr>
<tr>
<td>20</td>
<td>85.22 (0.23) 97.4 (0.03)</td>
<td>87.59 (0.25) 97.67 (0.03)</td>
<td>1368</td>
<td>4.74 (0.07)</td>
</tr>
<tr>
<td>10</td>
<td>80.15 (0.35) 96.44 (0.06)</td>
<td>81.97 (0.37) 96.63 (0.06)</td>
<td>588</td>
<td>3.69 (0.05)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>One hidden layer with sigmoid activation function</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>400</td>
<td>86.73 (0.39) 97.88 (0.04)</td>
<td>89.09 (0.49) 98.18 (0.04)</td>
<td>18808</td>
<td>23.2 (0.16)</td>
</tr>
<tr>
<td>300</td>
<td>86.97 (0.23) 97.88 (0.03)</td>
<td>89.3 (0.27) 98.18 (0.04)</td>
<td>14108</td>
<td>18.52 (0.43)</td>
</tr>
<tr>
<td>200</td>
<td>86.98 (0.04) 97.88 (0.02)</td>
<td>89.4 (0.08) 98.19 (0.01)</td>
<td>9408</td>
<td>13.01 (0.04)</td>
</tr>
<tr>
<td>100</td>
<td>86.51 (0.19) 97.78 (0.03)</td>
<td>88.79 (0.26) 98.07 (0.04)</td>
<td>4708</td>
<td>7.77 (0.08)</td>
</tr>
<tr>
<td>50</td>
<td>85.77 (0.28) 97.62 (0.04)</td>
<td>88.1 (0.3) 97.89 (0.04)</td>
<td>2358</td>
<td>4.9 (0.05)</td>
</tr>
<tr>
<td>20</td>
<td>83.19 (0.55) 97.07 (0.05)</td>
<td>85.3 (0.68) 97.31 (0.06)</td>
<td>948</td>
<td>3.48 (0.05)</td>
</tr>
<tr>
<td>10</td>
<td>77.69 (0.23) 95.89 (0.04)</td>
<td>79.37 (0.2) 96.05 (0.04)</td>
<td>478</td>
<td>3.02 (0.14)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Two hidden layers with ReLU activation function</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>400</td>
<td>82.37 (0.19) 97.35 (0.03)</td>
<td>84.51 (0.15) 97.66 (0.03)</td>
<td>179208</td>
<td>8.09 (0.06)</td>
</tr>
<tr>
<td>300</td>
<td>81.75 (0.15) 97.23 (0.03)</td>
<td>83.95 (0.15) 97.54 (0.02)</td>
<td>104408</td>
<td>6.63 (0.04)</td>
</tr>
<tr>
<td>200</td>
<td>81.73 (0.22) 97.22 (0.08)</td>
<td>83.91 (0.24) 97.54 (0.08)</td>
<td>49608</td>
<td>5.33 (0.02)</td>
</tr>
<tr>
<td>100</td>
<td>81.66 (0.87) 97.23 (0.11)</td>
<td>83.92 (0.88) 97.55 (0.11)</td>
<td>14808</td>
<td>4.33 (0.16)</td>
</tr>
<tr>
<td>50</td>
<td>81.74 (0.24) 97.23 (0.01)</td>
<td>84.04 (0.27) 97.54 (0.01)</td>
<td>4908</td>
<td>3.77 (0.03)</td>
</tr>
<tr>
<td>20</td>
<td>82.7 (0.75) 97.16 (0.09)</td>
<td>84.99 (0.84) 97.44 (0.09)</td>
<td>1368</td>
<td>3.23 (0.04)</td>
</tr>
<tr>
<td>10</td>
<td>78.27 (0.62) 96.25 (0.09)</td>
<td>80.15 (0.63) 96.44 (0.09)</td>
<td>588</td>
<td>3.13 (0.01)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>One hidden layer with ReLU activation function</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>400</td>
<td>83.29 (0.48) 97.45 (0.04)</td>
<td>85.56 (0.5) 97.77 (0.03)</td>
<td>18808</td>
<td>5.57 (1.32)</td>
</tr>
<tr>
<td>300</td>
<td>83.2 (0.33) 97.43 (0.02)</td>
<td>85.38 (0.38) 97.74 (0.03)</td>
<td>14108</td>
<td>4.14 (0.02)</td>
</tr>
<tr>
<td>200</td>
<td>83.7 (0.42) 97.49 (0.03)</td>
<td>85.92 (0.43) 97.79 (0.03)</td>
<td>9408</td>
<td>3.78 (0.1)</td>
</tr>
<tr>
<td>100</td>
<td>85.07 (0.26) 97.56 (0.06)</td>
<td>87.34 (0.25) 97.85 (0.06)</td>
<td>4708</td>
<td>3.38 (0.06)</td>
</tr>
<tr>
<td>50</td>
<td>84.9 (0.41) 97.46 (0.05)</td>
<td>87.16 (0.44) 97.73 (0.05)</td>
<td>2358</td>
<td>2.9 (0.08)</td>
</tr>
<tr>
<td>20</td>
<td>82.67 (0.37) 97.0 (0.03)</td>
<td>84.79 (0.34) 97.24 (0.03)</td>
<td>948</td>
<td>2.78 (0.04)</td>
</tr>
<tr>
<td>10</td>
<td>77.97 (0.49) 96.13 (0.08)</td>
<td>79.63 (0.5) 96.29 (0.08)</td>
<td>478</td>
<td>2.76 (0.04)</td>
</tr>
</tbody>
</table>

5.3 The high-level controller for GaIn

The main goal of the high-level controller is to recognize the intention of the person—whether he/she would like to sit down, stand up, or walk—and initiate the appropriate procedure. The high-level controller can be in five different states. These transitions and states are illustrated in Figure 19. In principle, some activity transitions are forbidden in order to reduce the error of the intention recognition. For instance, if the user is walking, then the system cannot perform a sitting down activity without first stopping and standing. When the user is sitting, then GaIn cannot infer walking-related activities without first standing up and standing.

The high-level controller consists of three different RapidHARe modules, which are described below.
Table 14: Recurrent Neural network.

<table>
<thead>
<tr>
<th>#Units</th>
<th>Without Border Tolerance</th>
<th>With Border Tolerance</th>
<th>#Params</th>
<th>Time (µs)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>200</td>
<td>82.97 (1.28)</td>
<td>97.58 (0.16)</td>
<td>83.31 (1.29)</td>
<td>97.64 (0.16)</td>
</tr>
<tr>
<td>150</td>
<td>80.15 (1.89)</td>
<td>97.21 (0.36)</td>
<td>80.57 (1.85)</td>
<td>97.29 (0.36)</td>
</tr>
<tr>
<td>100</td>
<td>78.93 (1.42)</td>
<td>96.79 (0.46)</td>
<td>79.29 (1.47)</td>
<td>96.86 (0.46)</td>
</tr>
<tr>
<td>75</td>
<td>82.09 (2.09)</td>
<td>97.44 (0.23)</td>
<td>82.43 (2.09)</td>
<td>97.52 (0.23)</td>
</tr>
<tr>
<td>50</td>
<td>75.54 (1.43)</td>
<td>96.57 (0.33)</td>
<td>75.88 (1.47)</td>
<td>96.64 (0.34)</td>
</tr>
<tr>
<td>20</td>
<td>73.83 (2.86)</td>
<td>96.49 (0.3)</td>
<td>74.12 (2.86)</td>
<td>96.55 (0.31)</td>
</tr>
<tr>
<td>10</td>
<td>70.81 (2.96)</td>
<td>95.73 (0.56)</td>
<td>71.13 (2.99)</td>
<td>95.79 (0.57)</td>
</tr>
</tbody>
</table>

One layer with sigmoid activation function

<table>
<thead>
<tr>
<th>#Units</th>
<th>Without Border Tolerance</th>
<th>With Border Tolerance</th>
<th>#Params</th>
<th>Time (µs)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>200</td>
<td>65.0 (6.15)</td>
<td>92.39 (1.59)</td>
<td>65.27 (6.2)</td>
<td>92.45 (1.59)</td>
</tr>
<tr>
<td>150</td>
<td>74.02 (5.01)</td>
<td>95.03 (1.36)</td>
<td>74.35 (5.08)</td>
<td>95.09 (1.37)</td>
</tr>
<tr>
<td>100</td>
<td>72.06 (3.05)</td>
<td>94.33 (0.79)</td>
<td>72.35 (3.07)</td>
<td>94.39 (0.79)</td>
</tr>
<tr>
<td>75</td>
<td>69.34 (3.48)</td>
<td>94.02 (0.79)</td>
<td>69.58 (3.53)</td>
<td>94.08 (0.8)</td>
</tr>
<tr>
<td>50</td>
<td>70.19 (1.53)</td>
<td>93.94 (0.54)</td>
<td>70.44 (1.53)</td>
<td>94.0 (0.54)</td>
</tr>
<tr>
<td>20</td>
<td>61.68 (6.27)</td>
<td>93.64 (1.12)</td>
<td>61.93 (6.31)</td>
<td>93.7 (1.12)</td>
</tr>
<tr>
<td>10</td>
<td>34.13 (4.54)</td>
<td>87.86 (0.97)</td>
<td>34.25 (4.56)</td>
<td>87.9 (0.98)</td>
</tr>
</tbody>
</table>

One layer with ReLU activation function

<table>
<thead>
<tr>
<th>Method</th>
<th>Without Border Tolerance</th>
<th>With Border Tolerance</th>
<th>#Params</th>
<th>Time (µs)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RapidHARe</td>
<td>87.14 (0.23)</td>
<td>87.75 (0.14)</td>
<td>86.48 (0.2)</td>
<td>97.83 (0.2)</td>
</tr>
<tr>
<td>RapidHARe-DF</td>
<td>88.93 (0.12)</td>
<td>88.01 (0.13)</td>
<td>87.9 (0.14)</td>
<td>97.92 (0.14)</td>
</tr>
<tr>
<td>ANN</td>
<td>84.55 (0.06)</td>
<td>89.01 (0.04)</td>
<td>86.98 (0.04)</td>
<td>97.88 (0.04)</td>
</tr>
<tr>
<td>RNN</td>
<td>80.87 (1.33)</td>
<td>87.21 (1.28)</td>
<td>82.97 (1.28)</td>
<td>97.58 (1.28)</td>
</tr>
<tr>
<td>HMM</td>
<td>82.59 (0.11)</td>
<td>83.14 (0.09)</td>
<td>81.54 (0.09)</td>
<td>96.76 (0.09)</td>
</tr>
</tbody>
</table>

Table 15: Main classification results.

<table>
<thead>
<tr>
<th>Method</th>
<th>Without Border Tolerance</th>
<th>With Border Tolerance</th>
<th>#Params</th>
<th>Time (µs)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RapidHARe</td>
<td>91.44 (0.17)</td>
<td>92.82 (0.17)</td>
<td>91.4 (0.27)</td>
<td>98.65 (0.27)</td>
</tr>
<tr>
<td>RapidHARe-DF</td>
<td>94.59 (0.13)</td>
<td>94.58 (0.12)</td>
<td>94.27 (0.12)</td>
<td>98.94 (0.12)</td>
</tr>
<tr>
<td>ANN</td>
<td>86.82 (0.01)</td>
<td>89.13 (0.01)</td>
<td>89.4 (0.08)</td>
<td>98.19 (0.08)</td>
</tr>
<tr>
<td>RNN</td>
<td>81.11 (1.34)</td>
<td>87.59 (1.34)</td>
<td>83.31 (1.34)</td>
<td>97.64 (1.34)</td>
</tr>
<tr>
<td>HMM</td>
<td>85.03 (0.14)</td>
<td>86.11 (0.14)</td>
<td>84.34 (0.14)</td>
<td>97.24 (0.14)</td>
</tr>
</tbody>
</table>

Performance measures are averaged over activities. 1RapidHARe using directional features (DF). 2The number of parameters in the models to be trained. 3Time in micro seconds to predict the activity of a single data frame measured on a single-thread CPU.

5.3.1 Sitting-standing module

The first RapidHARe module, denoted $C_{ss}$, is to recognize if the user is (1) sitting, (2) standing, or (3) performing other walking-related activity modes. This module uses one Gaussian component for each activity mode, based on four input features, namely: the raw x- and z-axes, accelerometer data from left and right thighs ($a(l,t,x)$, $a(l,t,z)$, $a(r,t,x)$, $a(r,t,z)$), respectively. Using equation 19, the prediction of this module can be formulated as

$$\hat{s}_{ss}(t) = \arg\max_{s_{ss} \in S_{ss}} \left\{ \prod_{k=0}^{K} P_{ss}(v_{ss}(t-k) | s_{ss}) \right\},$$  \hspace{1cm} (20)$$

where $v_{ss}(l) = [a(l,t,x)(l), a(l,t,z)(l), a(r,t,x)(l), a(r,t,z)(l)]$ is a four-dimensional vector for $l = t - K \rightarrow t$, is composed of the raw accelerometer signals; the states are $S_{ss} = \{ \text{“standing”}, \text{“sitting”}, \text{“}$$
bulatory”, and $P_{ss}(v \mid s)$ is actually a Gaussian density distribution, respectively. The length of the context window $K$ was 20.

5.3.2 Sitting-down module

The second module, denoted $C_{sd}$, is to recognize sitting down intention vs. walking intention while standing from EMG sensor data and the differences of the accelerometer data. The sitting-down intention was modeled with five Gaussian components, while standing and walking activity modes were modeled with two Gaussian components. The prediction in this module is based on the signals from the EMG sensors and the difference of the thighs’ positions.

During muscle activities, the EMG sensors record highly oscillating patterns. The useful information for intention recognition is the frequency of the oscillation and not the actual measured raw data. Therefore, the standard deviations (STD) of the gradients of the EMG signals are measured and calculated formally as:

$$\gamma_w(t) = \text{STD}_{i=0}^{w} \{ \partial \epsilon(t - i) \}, \quad (21)$$

where $t$ indicates the time frame, $w$ denotes the length of the “history,” and $\epsilon$ indicates a value observed with an EMG sensor (see: Table 4). In the experiments, $w = 10$ seemed to be an adequate value for the context window length. During static activities (standing and sitting), the STD of EMG changes will be close to zero, and during dynamic activities (standing up and sitting down), STD will have a large positive number.

Unfortunately, EMG sensors are sensitive and often produce corrupted data with various artifacts [18]. An example of an artifact can be seen in Figure 36(A) as a “waving” signal. Several methods have been proposed to restore the original signal [6, 159]; however, the original signal values are not necessary for GaIn. GaIn is only required to recognize the intention to sit down or stand up. Therefore, using the derivatives of the EMG signals is a plausible trick to overcome artifact artifacts, as can be seen in Figure 36(B).

Another issue is that the person in standing position may start walking again instead of sitting down and produces high muscle activity while moving one of his/her legs forward. Therefore, this module also has to distinguish walking mode from sitting down intention based on EMG and inertial sensor data. Thus, the standard deviation of the differences between the positions of the right and left thighs as a new feature might be a good indicator to distinguish walking from sitting down intention. This new feature is denoted with $\nu$ and formulated as follows:

$$\nu_x(t) = \text{STD}_{i=0}^{w} \{|a_{(t,t,x)}(t - i) - a_{(r,t,x)}(t - i)|\}, \quad (22)$$

The feature $\nu_z(t)$ is defined likewise. An example of this feature extraction can be seen in Figure 37.

Using equation 19, the prediction of this module can be formulated as
5.3.3 Standing-up module

Finally, the third module, denoted $C_{su}$, is to recognize the user’s intention to stand up from a sitting position based on EMG sensor data. The intention was modeled with 10 Gaussian components, while sitting was modeled with one Gaussian component. The input of this module is the standard deviation of the EMG signal changes compared to five past references ($\gamma_5(t)$), as was used in the previous model.

Using equation 19, the prediction of this module can be formulated as

$$\hat{s}_{su}(t) = \arg\max_{s_{su} \in S_{su}} \left\{ \prod_{k=0}^{K} P_{su}(v_{su}(t - k) \mid s_{su}) \right\},$$

(24)

where $v_{su}(l) = [\gamma^L_5(l), \gamma^R_5(l), \nu_u(l), \nu_s(l)]$ is a two-dimensional vector for $l = t - K \rightarrow t$, and $S_{su} = \{ \text{“sitting”}, \text{“standing up”} \}$. The $P_{su}(v \mid \text{“sitting”})$ is a mixture of 10 Gaussian models, while $P_{su}(v \mid \text{“standing up”})$ is a mixture of two Gaussian models. The length of the context window $K$ was 20.

Table 16 summarizes the features used and model architecture of each module.
Figure 37: Difference between X-axis of left and right tights accelerometer (A) during standing up and sitting down (notice the legs are parallel), (B) during standing and walking. Standard deviation with context window equal to 5 signals of difference between X-axis of left and right tights accelerometer (C) during standing up and sitting down, (D) during standing and walking.

Table 16: Features used in GaIn.

<table>
<thead>
<tr>
<th>Model</th>
<th>Features</th>
<th>Window length</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sitting-Standing $C_{ss}$</td>
<td>X-axis accelerometer data from left thigh: $a_{(l,t,x)}$, X-axis accelerometer data from right thigh: $a_{(r,t,x)}$, Z-axis accelerometer data from left thigh: $a_{(l,t,z)}$, Z-axis accelerometer data from right thigh: $a_{(r,t,z)}$</td>
<td>20</td>
</tr>
<tr>
<td>Sitting down $C_{sd}$</td>
<td>EMG sensor data variance from the left thigh $\gamma_{L}^l(t)$, EMG sensor data variance from the right thigh $\gamma_{R}^r(t)$, Variance of the differences on x-axis accelerometer data $\nu_x(t)$, Variance of the differences on z-axis accelerometer data $\nu_z(t)$</td>
<td>20</td>
</tr>
<tr>
<td>Standing up $C_{su}$</td>
<td>EMG sensor data variance from the left thigh $\gamma_{L}^u(t)$, EMG sensor data variance from the right thigh $\gamma_{R}^u(t)$</td>
<td>20</td>
</tr>
</tbody>
</table>

1 All raw features are scaled to $(0, 1)$. 2 Approximately one-third of a second.
5.3.4 The controller algorithm

The pseudo-code of the high-level controller algorithm is shown in Algorithm 1.

**Algorithm 1:** GaIn high-level controller algorithm. The parameters used in experiments are set to $K_{su} = 25, K_{sd} = 45, K_{sit} = 50, K_{stand} = 145$, and $\alpha = 50\%$. The operator $\approx_\alpha$ is defined to be true if its argument is true in $\alpha$ percent of cases.

```plaintext
/* Initialization */
t ← 0
Calculate $\phi_{start}^L, \phi_{start}^R$ using eq. 26 and predict initial state // Possibly “sit”
/* Main loop */
while True do
    Calculate $\hat{s}_{ss}(t)$, $\hat{s}_{su}(t)$, $\hat{s}_{sd}(t)$ using eqs. 20, 24 and 23
    /* Gait inference for ‘‘stand’’ or ‘‘ambulatory’’ state */
    if $\hat{s}_{ss}(t) =$ “ambulatory” or $\hat{s}_{ss}(t) =$ “stand” then
        $y(t) \leftarrow R(v_{rnn}(t))$ // Infer shank position using RNN, see Chapter 6.
    end
    /* Transition from ‘‘stand’’ to ‘‘sit down’’ state */
    if $\hat{s}_{sd}(t-1) =$ “sit down”(\(\forall l = 0..K_{sd}\)) and $\hat{s}_{ss}(t-l) \approx_\alpha$ “stand”(\(\forall l = K_{sd}..K_{stand}\)) then
        SittingDownProcedure()
    end
    /* Transition from ‘‘sit’’ to ‘‘stand up’’ state */
    if $\hat{s}_{su}(t-1) =$ “stand up”(\(\forall l = 0..K_{su}\)) and $\hat{s}_{ss}(t-l) \approx_\alpha$ “sit”(\(\forall l = K_{su}..K_{sit}\)) then
        StandingUpProcedure()
    end
    t ← t + 1
end
```

5.3.5 Experimental evaluation of the high-level controller in GaIn

The activity recognition was evaluated with $\text{Precision} = \frac{TP}{TP+FP}$ and $\text{Recall} = \frac{TP}{TP+FN}$ metrics, where $TP$, $FP$, and $FN$ denote the number of true positive, false positive, and false negative predictions, respectively. In addition, the $F_1$ scores were calculated and reported.

The goal of GaIn’s high-level controller is to understand whether the person intends to sit down or stand up. The triggering may be not as fast as healthy users’ performance. However, in this work, if the system predicts intention with some lag, it is counted as a $TP$ result. Moreover, all high-level controller metrics were calculated based on person intention, not based on data samples. Thus, $TP$ is counted when the activity is correctly predicted (or triggered) during the same ground truth activity. $FP$ is counted when the activity is predicted (or triggered) during an other ground truth activity. And finally, $FN$ is counted when GaIn misses an activity recognition.

The results, summarized in Table 17, show that standing and sitting position recognition can be achieved with high accuracy; however, it is easier to recognize standing up intention than sitting down intention. The GaIn system achieved 99% recall and 99% precision for recognizing standing up intention, but it achieved only 68% recall and 99% precision for sitting down activity. The reason is that the muscle activity in both thighs is very low in a sitting position, thus it is effortless to recognize standing up intention form the sudden increase in muscle activity. However, muscle
Table 17: Classification results for each participant.

<table>
<thead>
<tr>
<th>Participant ID</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>standing up recall</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.93</td>
</tr>
<tr>
<td>standing up precision</td>
<td>1</td>
<td>1</td>
<td>0.94</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.94</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.94</td>
<td>1</td>
<td>1</td>
<td>0.99</td>
</tr>
<tr>
<td>sitting up $F_1$</td>
<td>1</td>
<td>1</td>
<td>0.97</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.97</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.97</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.97</td>
</tr>
<tr>
<td>sitting down recall</td>
<td>1</td>
<td>1</td>
<td>0.78</td>
<td>1</td>
<td>1</td>
<td>0.60</td>
<td>1</td>
<td>0.30</td>
<td>0.19</td>
<td>1</td>
<td>0.40</td>
<td>0.82</td>
<td>0.34</td>
<td>0.86</td>
<td>0.5</td>
<td>0.88</td>
<td>0.46</td>
<td>0.48</td>
<td>0.60</td>
</tr>
<tr>
<td>sitting down precision</td>
<td>1</td>
<td>1</td>
<td>0.90</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.93</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.90</td>
<td>1</td>
<td>1</td>
<td>0.99</td>
</tr>
<tr>
<td>sitting down $F_1$</td>
<td>1</td>
<td>1</td>
<td>0.88</td>
<td>0.95</td>
<td>0.67</td>
<td>0.32</td>
<td>1</td>
<td>0.67</td>
<td>0.90</td>
<td>0.93</td>
<td>0.92</td>
<td>0.11</td>
<td>0.93</td>
<td>0.71</td>
<td>0.55</td>
<td>0.75</td>
<td>0.75</td>
<td>0.82</td>
<td>0.93</td>
</tr>
<tr>
<td>standing recall</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.96</td>
<td>1</td>
<td>0.97</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>standing precision</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.94</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.94</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>standing $F_1$</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.97</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.97</td>
<td>1</td>
<td>0.98</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>sitting recall</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.95</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>sitting precision</td>
<td>1</td>
<td>0.78</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.99</td>
</tr>
<tr>
<td>sitting $F_1$</td>
<td>1</td>
<td>0.88</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.98</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Note that some participants (e.g., ID=5,14) yield poor results due to weak EMG sensor signals. These cases could be circumvented using individual EMG signal calibration.

Activity is already present in a standing position, which makes it more challenging to distinguish a patient’s simple balancing or walking efforts from a sitting down intention. Nevertheless, incorrect activity prediction can result in different impacts on the patient. When the GaIn system incorrectly recognized a standing up activity while the user is sitting, then the system simply stretched the robotic prosthetic leg, resulting in no harm to the patient. However, when a sitting down intention is predicted while the user is simply standing, then the patient would fall and may suffer serious injury. Thus, it is more important to achieve lower false alarm (high precision) than missed alarm (high recall) rates for sitting down activity. Therefore, the decision threshold was calibrated so that the activity recognition module achieved as high as 99% precision at the expense of a recall, which decreased to 68%. As a consequence, users may need to produce clearer and longer signals to the system for sitting down, but this results in GaIn causing fewer injuries from falling.

To examine the prediction latency, a histogram of the activity recognition lag time is plotted in Figure 39. On average, it takes 602 milliseconds to recognize standing up intention (shown in Figure 39A), while it takes 846 milliseconds to recognize sitting down intention (shown in Figure 39B). It should be noted that standing up recognition has lower variance than sitting down recognition, which is concluded from the width of the distributions in Figure 39. Note that the higher lag time for sitting down recognition is a result of the threshold calibration, as discussed above. It should be noted that the predicted length of the sitting down and standing up activities in the figures is irrelevant here, because length would depend on how the actual robotic prosthetic legs perform these movements.

Finally, as was mentioned, the quality of the EMG signals greatly depends on the physical properties of the user’s skin. Some users generated poor EMG signals (see the results for participant ID=5,14 in Table 17) that hampered the activity recognition consistently, while some users generated good-quality EMG signals (see the results for participant ID=1,6 in Table 17), resulting in almost perfect activity recognition. Therefore, to mitigate dependency on the EMG signals, the high-level controller can be calibrated for each patient individually in the future.
Figure 38: Activity recognition in GaIn with EMG signal with artifact (A), weak signal (B), weak signal with artifact (C), and good signal (D). Signals obtained from participants ID=1, 7, 12, 16, respectively. Note that plot D consists of three continuing panels.

Figure 39: Activity recognition latency in seconds (s) for standing up (A) and sitting down (B).

5.3.6 Implementation details

The classification methods were implemented using the Python scikit-learn package (version 0.18.1). The RNN/LSTM was implemented with the Keras library (version 2.1.2) using the TensorFlow framework as backend (version 1.4.0). Training and testing were carried out on a PC equipped with an Intel Core i7-4790 CPU, 8 GB DDR-III 2400 MHz RAM, and an Nvidia GTX Titan X GPU.
5.4 Conclusions

The goal of the GaIn system is to predict the position of the shanks based on the thighs’ movements. However, in a real-life application, sitting down and standing up intentions cannot be recognized from thigh movements. To circumvent this, EMG sensors were placed on the skin over the vastus lateralis thigh muscles; therefore, the patient could signal her/his intentions by increasing thigh muscle activity. GaIn’s high-level controller achieved 99% precision and recall in recognizing standing up intention and achieved 99% precision and 68% recall in recognizing sitting down intention. For safety reasons, the decision rule was adjusted accordingly to maintain low false alarms (high precision) at the expense of high missed alarms (low recall). As a result, users may need to produce clearer signals to indicate sitting down intention.

The Author developed a novel activity mode and intention recognition method used in GaIn as a high-level controller, called RapidHARe. This method is also suitable for HAR tasks in general.

RapidHARe does not employ any dynamic programming-based algorithms, which are notoriously slow for inference, nor does it employ feature extraction or selection methods. In comparative tests, it was shown that RapidHARe is an extremely fast predictor, one-and-a-half times faster than artificial neural network (ANN) methods, and more than eight times faster than recurrent neural networks (RNNs) and hidden Markov models (HMMs). Moreover, in performance, RapidHARe achieves an $F_1$ score of 94.27% and accuracy of 98.94%, and when compared to ANN, RNN, HMM, it reduces the F1-score error rate by 45%, 65%, and 63% and the accuracy error rate by 41%, 55%, and 62%, respectively. Therefore, RapidHARe is suitable for real-time recognition on mobile devices.

Finally, here there will be some comments on deep-learning technologies related to HAR. Recently, deep-learning technologies, deep LSTM, and deep convolutional LSTM have emerged for activity recognition systems with superb performance, mainly in ADL and GR [109, 58]. These methods are capable of learning features automatically from the data [20]. The price of this skill is that they consist of millions of model parameters that are more difficult to train, and most importantly, they result in longer prediction times and require more CPU time compared to inexpensive models such as decision trees. On the other hand, the Author argues that these methods have too high of a capacity for HAR and HGA problems, and thus, they overfit. These problems involve only a few thousand input features, and the “complexity” of the underlying data manifold is rather low. LSTM methods have the capacity to remember the activity performed sometime ago, which might be useful for recognizing daily activities, such as scrambling eggs or washing dishes. However, for HAR- and HGA-related problems, such skills are not needed because in most cases, the current activity is independent of activities performed some time ago. For instance, if the next activity is going to be walking up the stairs, then it is because there are stairs ahead, and this fact is independent of previous activities, such as whether the user was sitting or running before. This idea is supported by the studies in [109, 58]. Both studies have reported improvement in performance for ADL using deep LSTM methods. However, in freezing-of-gait prediction tasks, Hammerla et al. have reported a 76% $F_1$ score in Table 2 in [58], while a simple method such as random forests and C4.5 using smartly crafted features has achieved.
an $F_1$ score over 95% on the same dataset, as shown in Table 2 in [101]. Similar conclusions can be reached from the results presented in Table 2 in [150], where the nearest-neighbor and random forest methods outperform multilayer perceptrons in test scenarios (which the authors termed “impersonal” and “hybrid”) in which training and test data were recorded by different users. Thus, these results support this argument, and, therefore, deep models of high capacity for HAR and HGA problems do not seem to be justified. Thus, smartly designed features used along with computationally inexpensive models can provide faster and more energy-efficient methods with low latency for this field.
6 Gait mid-level controller

The Gain mid-level controller aims to infer human gait and to predict shanks movements. It should be noted, that GaIn has only one mid-level controller for all gait activities and phases.

The shank movement prediction was modeled with recurrent neural networks (RNNs) [48] with LSTM units [62]. RNNs are universal mathematical tools for modeling statistical relationships in sequential data. The RNN consisted of 50 LSTM hidden units in one hidden layer. The learning objective for the RNN was to minimize the squared error between the predicted and the true shank angle. For the training, the input sequential data were chunked into 15 long data segments. The RNN uses a four-component input vector, in which each component corresponds to the angle and the angular speed of the left and right thighs, respectively.

6.1 LSTM

LSTM cells use two types of memory units to represent the past information of sequential data: one to capture short-term dependencies denoted by $h$ and the other to capture long-term dependencies called state $c$. State $c$ runs through the whole time and an LSTM performs four steps to update its data using so-called gates. The gates are: input gate, forget gate, input modulation, and output gate. The structure of an LSTM cell is shown in Figure 41. One of the main advantage of LSTMs is that each gate is differentiable, so their operations can be learned from data. The gates and the data manipulation steps are defined as follows:

- The forget gate calculates which information should be removed from state $c_t$ based on the hidden unit $h_{t-1}$ and the current input $x_t$. It is defined formally as $f_t = \sigma(W_f[v_t, h_{t-1}] + b_f)$, where $\sigma$ denotes the sigmoid function. The output $f_t$ can be considered as a bit vector, which indicates the components of the state vector $c$ to be forgotten. For instance, $f_t \approx 1$ indicates that the value of the $i$th component of $f_t$ will be kept and $f_t \approx 0$ indicates that the value of that component will be forgotten.

- The input gate controls which information from the input should be kept and stored in the state vector $c_t$ at time step $t$. It is formally defined as $i_t = \sigma(W_i[v_t, h_{t-1}] + b_i)$ and can be interpreted as a binary mask vector.

- The input modulation gate calculates a new candidate state vector $\tilde{c}_t = tanh(W_g[v_t, h_{t-1}] + b_g)$.

- The new state vector $c_t$ is then calculated by $c_t = f_t \cdot c_{t-1} + i_t \cdot \tilde{c}_t$.

- The output gate decides which parts of the cell state go to the output. It is calculated by $o_t = \sigma(W_o[v_t, h_{t-1}] + b_o)$.

- The new hidden state $h_t$ is formed from the new cell state whose values are first pushed between -1 and 1 using the $tanh$ function, and then multiplied by the values of the output gate. Formally, $h_t = o_t \cdot tanh(c_t)$.
Finally, the emission or the output of the cell (i.e. in our case the predictions for the position of the shank) is calculated using \( y_t = \tanh(W_y h_t + b_y) \).

In the above, \( \sigma \) denotes the sigmoid function, ‘\( \cdot \)’ represents element-wise or Hadamard product of vectors, \([.,.]\) denotes vector concatenation, \(W\)'s denote weight matrices, and \(b\)'s denote the corresponding biases whose values are to be learnt from data.

![Illustration of a folded (A) and an unfolded (B) recurrent neural network structure.](image)

In our work, the observed data \( v_t \) is a 4-component vector, in which each component corresponds to the angle and the angular speed of the left and right thighs, respectively. The angular data were calculated from two triaxial gyroscopes and accelerometer sensors located on the right and left thighs using the methods described by Pedley [113]. The RNN was trained to predict the angles of both shanks.

![Structure of LSTM cell. The operators \( \cdot \), \( + \), \([.,.]\), and \( \tanh \) in boxes represent element-wise multiplication, addition, concatenation, and \( \tanh \) operations on vectors, respectively.](image)

### 6.2 Feature extraction methods

Raw data obtained from the gyroscope and accelerometer sensors were filtered with a moving average. This was performed to remove the bias drift of inertial sensors [54]. The gait inference method is based on the thigh angle and angular speed data in the sagittal plane. The initial angle
degrees for thigh and shank are calculated based on the accelerometer data and Earth’s gravity [113]. Formally, the start angle of the left thigh ($\theta$) is calculated with

$$\theta_{\text{start}}^L = \arctan \left( \frac{a_{l,t,y}}{\sqrt{a_{l,t,x}^2 + a_{l,t,z}^2}} \right),$$

(25)

where $a_{l,t,y}$, $a_{l,t,x}$, and $a_{l,t,z}$ denote the values of the accelerometer sensors located on the left thigh. The start angle of the left shank ($\phi$) is calculated via

$$\phi_{\text{start}}^L = \arctan \left( \frac{a_{l,s,y}}{\sqrt{a_{l,s,x}^2 + a_{l,s,z}^2}} \right),$$

(26)

where $a_{l,s,y}$, $a_{l,s,x}$, and $a_{l,s,z}$ denote the values of the accelerometer sensors located on left shank. The angular velocities are from the gyroscope data. Using the angular velocities of the left thigh and shank $\omega_{l,t,y}$ and $\omega_{l,s,y}$ at time $t$, the angles of the thigh $\theta(t)$ and shank $\phi(t)$ at time $t$ can be calculated as follows [113]:

$$\theta^L(t) = \theta_{\text{start}}^L + \int_0^t \omega_{l,t,y}(x)dx,$$

(27)

$$\phi^L(t) = \phi_{\text{start}}^L + \int_0^t \omega_{l,s,y}(x)dx.$$  

(28)

The variables for the right leg $\theta^R(t)$, $\phi^R(t)$, $\theta_{\text{start}}^R$, $\phi_{\text{start}}^R$ are calculated the same way, but with information obtained from the right leg. Consequently, the gait inference module can be formulated as

$$y(t) = R(v_{\text{inn}}(t)),$$

(29)

where $v_{\text{inn}}(t) = [\theta^L(t), \theta^R(t), \omega_{l,t,y}(t), \omega_{r,t,y}(t)], y_t = [\phi^L(t), \phi^R(t)]$ contains the angle prediction for the right and left shanks at time $t$, and $R$ denotes a recurrent neural network with LSTM units. It should be noted that the index “$t$” is overriding here and not to be confused. It stands for both “tight” in $\omega_{l,t,y}$ and $a_{r,t,y}$ and denotes “time” in $\theta^L(t)$, and $\phi^L(t)$.

### 6.3 Gait inference results

The results for gait inference are shown in Figure 42 for various walking-related activities such as walking, running, and taking the stairs up and down. The dashed lines show the true angle of the shank, while the solid line shows the prediction for the shank angle. The line segments going upward correspond to swing phases and line segments going downward correspond to stance phases in the gait cycle. The error, the difference between the true and the predicted movements, is indicated by the shaded area. It appears that the errors occur at the peaks and troughs which correspond to the turning point between the swing and stance phases. The color of the background indicates activity performed. Note that these activity labels were not incorporated.
into the training procedure; they are presented simply for illustration purposes. The prediction errors for different activities are listed in Table 18. The error was calculated for each activity over all data of all users. The average of the prediction errors for the shank angles across different activities is 4.55°. Figure 43 shows the coordination and coordination variability of the true (red) and predicted (blue) shank angles with respect to the thigh angles. Here, the same data are used as for Figure 3. This scatter plot shows that the predicted shank angles are in accordance with the true shank angles. However, the predicted shank angles do not span over the range of the true angles in some cases. For instance, in the plot C, the predicted shank angles do not reach the extremes of the true angles. This is the error which occurs at peaks and troughs in Figure 42. Inference results for different activities and participants are shown in Figures 44, 45, and 46. More inference results can be found in Appendix A.

Table 18: Gait inference error.

<table>
<thead>
<tr>
<th>Activity</th>
<th>Walking</th>
<th>Running</th>
<th>Going up</th>
<th>Going down</th>
<th>Standing</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>4.988</td>
<td>5.648</td>
<td>5.820</td>
<td>5.148</td>
<td>1.174</td>
<td>4.555</td>
</tr>
<tr>
<td>STD</td>
<td>0.910</td>
<td>2.212</td>
<td>1.299</td>
<td>1.158</td>
<td>0.457</td>
<td>1.207</td>
</tr>
</tbody>
</table>

Error measured in absolute difference between the true and the predicted shank angles in degrees.

6.4 Variance in different phases

People walk differently, resulting in variance in gaits [30]. Moreover, gait varies over different cycles for the same person as well. Figure 3 shows this natural variance. This variance prevents the achievement of 100% accuracy in gait prediction for someone’s gait based on other people’s gait data. It has also been noticed that variance in the swing phase is larger than in the stance phase [155]. This is as expected, since the stance phase is more important in walking stability, while legs may move more freely in the swing phase [155]. This fact is also observed in HuGaDB. In Figure 47, panel A shows the shank angles of the gait cycle in the stance phase (blue lines) and the variance (red line), and panel B shows the same information for the swing phase. The figure shows that the variance is higher in the swing phase. Therefore, higher prediction error is expected for the swing phase than for the stance phase. In fact, the mean shank degree prediction error across all activities is 4.783 (STD: 1.171) in the stance phase and 6.182 (STD: 1.680) in the swing phase. Table 19 shows detailed prediction errors for different activities.

Table 19: GaIn inference error.

<table>
<thead>
<tr>
<th>Activity</th>
<th>Walking</th>
<th>Running</th>
<th>Going up</th>
<th>Going down</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>Swing phase</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mean</td>
<td>5.826</td>
<td>6.420</td>
<td>6.738</td>
<td>5.744</td>
<td>6.182</td>
</tr>
<tr>
<td>STD</td>
<td>1.0817</td>
<td>2.750</td>
<td>1.437</td>
<td>1.452</td>
<td>1.680</td>
</tr>
<tr>
<td>Stance phase</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mean</td>
<td>4.268</td>
<td>4.967</td>
<td>5.140</td>
<td>4.758</td>
<td>4.783</td>
</tr>
<tr>
<td>STD</td>
<td>0.800</td>
<td>1.700</td>
<td>1.215</td>
<td>0.969</td>
<td>1.171</td>
</tr>
</tbody>
</table>

Error measured in degrees.
Figure 42: GaIn inference for various walking-related activities. The activity is indicated by the
background color for the reader, but this information was not used for prediction. The shank
degree is predicted based on thigh angles (not shown). Solid black line shows the predicted, the
dashed line shows the true angles of the right shank, while the shaded area between them indicates
the prediction error. Plots for the left leg are similar.
6.5 Inference error around activity change

The errors around activity changes were closely examined, for instance, when a walking user started running. The gait inference errors were measured in a range of ±15 data samples (equivalent to half of a second) around the true activity change. It was found that the shank degree prediction error is 5.44°, which is not especially larger than general prediction error. The detailed results for different activity transitions are shown in Table 20.

Table 20: Average shank degree prediction error at activity transitions.

<table>
<thead>
<tr>
<th>Activity transition</th>
<th>Mean</th>
<th>STD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Walking → Running</td>
<td>5.79</td>
<td>2.297</td>
</tr>
<tr>
<td>Walking → Going up</td>
<td>5.34</td>
<td>1.417</td>
</tr>
<tr>
<td>Walking → Going down</td>
<td>5.68</td>
<td>0.959</td>
</tr>
<tr>
<td>Walking → Standing</td>
<td>4.50</td>
<td>0.742</td>
</tr>
<tr>
<td>Running → Walking</td>
<td>5.31</td>
<td>2.352</td>
</tr>
<tr>
<td>Going up → Walking</td>
<td>5.15</td>
<td>1.661</td>
</tr>
<tr>
<td>Going up → Standing</td>
<td>7.24</td>
<td>0.837</td>
</tr>
<tr>
<td>Going down → Going up</td>
<td>6.21</td>
<td>0.479</td>
</tr>
<tr>
<td>Going down → Walking</td>
<td>6.22</td>
<td>1.734</td>
</tr>
<tr>
<td>Standing → Going up</td>
<td>5.75</td>
<td>1.331</td>
</tr>
<tr>
<td>Standing → Walking</td>
<td>4.20</td>
<td>3.065</td>
</tr>
<tr>
<td>Standing → Going down</td>
<td>6.11</td>
<td>2.242</td>
</tr>
<tr>
<td>Mean</td>
<td>5.44</td>
<td>1.471</td>
</tr>
</tbody>
</table>

The degree error was measured in ±15 sample interval (around half a second long range) at the true activity transition border.

6.6 Gait inference results for one leg

Several studies provide methods and results for predicting the gait trajectory for a single leg wherein the prediction is based on the position and the movements of the sound leg; therefore, the Author provides here results obtained with GaIn for inferring the shank movement of a single leg. In this case, the input to GaIn constituted of data obtained from the movements of the thighs.
Figure 44: GaIn inference for walking and going up for participant ID=1. (A) gait inference for left leg, (B) gait inference for right leg. The activity is indicated by the background color for the reader, but this information was not used in the methods. The shank degree is predicted based on thigh angles (not shown). Solid black line shows the predicted, the dashed line shows the true angles of the right shank, while the shaded area between them indicates the prediction error. Plots for the left leg are similar.
Figure 45: GaIn inference for walking and going down for participant ID=6. (A) gait inference for left leg, (B) gait inference for right leg.

Figure 46: GaIn inference for walking and running for participant ID=11. (A) gait inference for left leg, (B) gait inference for right leg.
Figure 47: Shank angles of different participants in stance phase (A) and swing phase (B).

The results show that the prediction error for one leg is roughly half of the error obtained for prediction of two shanks (cf. Table 18). The average error decreases from 4.55° to 3.247°. The results are shown in Table 22 (cf. Table 19).

Table 21: Gait inference error for one leg prediction.

<table>
<thead>
<tr>
<th></th>
<th>Walking</th>
<th>Running</th>
<th>Going up</th>
<th>Going down</th>
<th>Standing</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>3.577</td>
<td>4.814</td>
<td>3.632</td>
<td>3.635</td>
<td>0.96</td>
<td>3.247</td>
</tr>
<tr>
<td>STD</td>
<td>0.868</td>
<td>1.191</td>
<td>0.537</td>
<td>0.771</td>
<td>0.364</td>
<td>0.576</td>
</tr>
</tbody>
</table>

Error measured in absolute difference between the true and the predicted shank angles in degrees.

Table 22: GaIn inference error in stance and swing phases for one leg prediction.

<table>
<thead>
<tr>
<th></th>
<th>Swing phase</th>
<th>Stance phase</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Walking</td>
<td>Running</td>
</tr>
<tr>
<td></td>
<td>STD</td>
<td>1.289</td>
</tr>
<tr>
<td>Stance phase</td>
<td>Mean</td>
<td>2.792</td>
</tr>
<tr>
<td></td>
<td>STD</td>
<td>0.609</td>
</tr>
</tbody>
</table>

Error measured in degrees.

Interestingly, the error around activity mode transition does not decrease considerably, as is shown in Table 23 (cf. Table 20).

### 6.7 Conclusions

GaIn was designed to predict the movements of the lower legs based on the movements of both thighs as a basis for building non-invasive, robotic lower limb prostheses for patients suffering from double trans-femoral amputation. It was shown that the shank angle can be predicted using recurrent neural networks with LSTM memory cells using thigh degrees as input. Experimental results showed that the GaIn system is highly accurate, and it achieved 4.55° prediction error on average.

The error was closely examined around activity mode transition, for instance, when a walking person started running. The gait inference error in a range of ± 15 data samples (equivalent to
Table 23: Average shank degree prediction error at activity transitions for one leg prediction.

<table>
<thead>
<tr>
<th>Activity transition</th>
<th>Mean</th>
<th>Std</th>
</tr>
</thead>
<tbody>
<tr>
<td>Walking → Running</td>
<td>6.865</td>
<td>1.955</td>
</tr>
<tr>
<td>Walking → Going up</td>
<td>4.03</td>
<td>0.932</td>
</tr>
<tr>
<td>Walking → Going down</td>
<td>4.692</td>
<td>0.581</td>
</tr>
<tr>
<td>Walking → Standing</td>
<td>3.192</td>
<td>0.964</td>
</tr>
<tr>
<td>Running → Walking</td>
<td>4.595</td>
<td>1.234</td>
</tr>
<tr>
<td>Going up → Walking</td>
<td>3.976</td>
<td>0.761</td>
</tr>
<tr>
<td>Going up → Standing</td>
<td>4.94</td>
<td>0.594</td>
</tr>
<tr>
<td>Going down → Going up</td>
<td>4.58</td>
<td>1.208</td>
</tr>
<tr>
<td>Going down → Walking</td>
<td>5.194</td>
<td>1.532</td>
</tr>
<tr>
<td>Standing → Going up</td>
<td>4.595</td>
<td>3.059</td>
</tr>
<tr>
<td>Standing → Walking</td>
<td>3.24</td>
<td>1.547</td>
</tr>
<tr>
<td>Standing → Going down</td>
<td>3.962</td>
<td>1.105</td>
</tr>
<tr>
<td>Mean</td>
<td>4.418</td>
<td>0.628</td>
</tr>
</tbody>
</table>

The degree error was measured in ± 15 sample interval (around half a second long range) at the true activity transition border.

half of a second) around the true activity change is not significantly higher than in general, and it is as low as $5.44^\circ$. 
7 Conclusions

This thesis presented the GaIn gait inference system, which is suitable for controlling robotic prosthetic legs for patients suffering at most from double transfemoral amputation. The GaIn system consists of two controllers: a high-level controller, called RapidHAR, for recognizing the activity mode or the intention of the patients, and a mid-level controller to infer the actual trajectory of a robotic prosthetic leg. In addition, the Author of this dissertation provided a database suitable for analyzing human gait in great detail that is free of charge.

One question still remains: is GaIn really suitable for the task it was designed for? In practice, several challenges will be faced, and to some extent, they will affect the overall system’s behavior. These issues cannot be foreseen now. For instance, how much will the angle misprediction affect the person’s balance while walking? How will the system behave when the patient wants to correct misprediction? Certain types of amputations may also damage the thigh muscle partially, and it might not provide sufficient signals, so the EMG sensors will need to be placed over other muscles. However, previous works in the prosthetic field have shown that humans are capable of walking with robotic prosthetic legs. Moreover, it should be noted that the participants were able to realize effective volitional knee control in the absence of any significant proprioceptive and haptic feedback using an active knee control prosthetic [65].

The Author is aware of the shortcomings related to the EMG sensors and to their sensitivity. The strength of the signals may depend on variability of the skin, or it may even change when the users perform physical activity and starts to sweat. This phenomenon was observed during HuGaDB data collection, and many other researchers have also noticed this during their research. Please also note that this study does not focus on EMG sensors and their related complications. The GaIn system will certainly inherit many problems and benefits associated with EMG sensor and it is not our aim to fix these issues here. However, one also might consider replacing the EMG sensors with other ones, for instance, air pressure sensors and air bladders, for muscle activity monitoring. For more information on these, see [71]. These air pressure sensors are considered more robust in muscle activity monitoring than EMG signals. It should be noted that the GaIn system does not require the quantification of the muscle activity. GaIn basically just needs the information of whether the muscle is active, and this decision is made on the variance of the signal changes and whether the variance exceeds a certain threshold or not.

Certainly, the system will not work well if the sensors are not installed properly. Note that when the HuGaDB data were recorded few years ago, some variance in sensor installation was allowed. The location and the orientation of the sensors were not regulated very precisely. This provided some variance in the data and, probably, this makes the GaIn system more robust to sensor installation in practice and provides better generalization for the machine learning algorithms. This will also give the patients more freedom in putting the sensors on.

Please also note that this dissertation did not discuss sensors and their related complications. Moreover, this thesis did not investigate any invasive sensors due to the absence of open datasets and difficulties with collection of real data in this type of field.
In one of the earliest studies of human gait analysis in 1995 by Sepulveda and his colleagues [126], two conclusions were drawn, namely, (1) one needs two separate neural networks for swing and stance phases, respectively, and (2) the ANN model requires calibration to the patient. In this doctoral dissertation, the Author shows that these conclusions do not hold anymore, and the Author (1) built a single neural network which is capable of inferring the gait in both the swing and stance phases for several locomotive modes and (2) trained a machine learning system that generalizes well to new users. For future research, perhaps the most interesting open question remains whether this trend will continue – specifically, whether it is possible to train a single model in an end-to-end fashion using deep learning techniques which encapsulates all high-, mid-, and low-level controllers along with gait phases into a single black box. The Author guesses that it is possible; however, it might require much more data from many more patients.

7.1 Main results of this thesis

The GaIn system comprises three main parts: (1) a dataset suitable for training and testing, (2) a high-level controller to recognize the patient’s activity modes and intentions, and (3) a gait inference method to generate the trajectory for robotic prosthetic legs. Below, the novelty and the Author’s results are summarized in three thesis points, and a summary of the supporting articles can be found in Table 24.

1. HuGaDB: the dataset for training the GaIn system [26]. Unfortunately, existing datasets for HGA and HAR were not adequate for the aim of this research project, because they did not contain detailed information on the movements of the parts of the legs. This dataset is unique in the sense that HuGaDB is the first to provide human gait data in great detail, mainly from inertial sensors, and contains segmented annotations for studying the transitions between different activities. The Author constructed the HuGaDB dataset, of which the main and novel characteristics are the following:

   (a) The HuGaDB dataset provides information about each part of the human leg during several walking-related activities in great detail, from inertial and EMG sensors. Six inertial sensors (each sensor consisted of one 3D-axis accelerometer and one 3D-axis gyroscope) were mounted on the left and right thigh, shin, and foot, respectively, and a pair of EMG sensors were mounted on the left and right thighs. Therefore, HuGaDB gives detailed information on how each part of the legs moves and how the parts move relative to each other.

   (b) The HuGaDB dataset contains continuous recordings of combinations of activities, and the data are segmented and annotated with the label of the activity currently performed. Thus, this dataset is suitable for analyzing both human gait and transition activities.

   (c) The data were collected from 18 participants in total. These participants were healthy young adults: four females and 14 males, average age of 23.67 (STD: 3.69) years, an average height of 179.06 (STD: 9.85) cm, and an average weight of 73.44 (STD: 16.67) kg. In total, they provide around 10 hours of data recording.
The HuGaDB article was published in Springer’s Q2 journal *Lecture Notes in Computer Science*: [26] and it became quite popular among researchers. HuGaDB has been cited by [129, 80, 134, 13, 12] as of 29 March 2019.

2. RapidHARe: the Author developed a novel activity mode and intention recognition method used in GaIn as a high-level controller called RapidHARe [28]. This method is also suitable for HAR tasks in general.

   (a) RapidHARe is based on a dynamic Bayesian network. RapidHARe has low prediction latency (A-1),\(^3\) is of fast and computationally inexpensive (A-2), provides smooth recognition (A-3), and generalizes well to new users (A-4).

   (b) RapidHARe outperforms all other state-of-the-art HAR methods in accuracy and speed (A-5). RapidHARe reduces the \(F_1\)-score error rate by 45\%, 65\%, and 63\% and the accuracy error rate by 41\%, 55\%, and 62\% when it is compared to artificial neural networks, recurrent neural networks, and hidden Markov models, respectively.

   (c) RapidHARe is used in the high-level controller to predict the patient’s intention for stand up and sit down mainly from data obtained by EMG sensors placed on the skin over the vastus lateralis thigh muscles. The controller achieved 99\% precision and recall in recognizing standing up intention and 99\% precision and 68\% recall in recognizing sitting down intention.

3. GaIn: a gait inference system that is suitable for controlling robotic prosthetic legs [27].

   (a) The GaIn framework can be used in lower limb prostheses for patients suffering from double transfemoral amputation, in exoskeleton design, etc. In contrast, most other methods are only suitable for controlling one prosthetic leg.

   (b) The GaIn system is based on the observation that the movement of the thigh and shin is highly but non-linearly correlated during regular walking-related activities. This is illustrated in Figure 3. No other method relies on this assumption; in fact, other methods usually extract more data from the sound leg as well.

   (c) GaIn infers the shin position based on the position and movement of the thighs using recurrent neural networks with long-short-term memory units. GaIn achieves a prediction error as low as 4.55° on average on natural terrain and generalizes well to new users. In contrast, other methods are often calibrated and tested on the same patient on treadmills.

   (d) The GaIn system does not assume a fully periodic gait (B-1);\(^4\) it can infer gait for several ambulatory activities (B-2, B-6), has small prediction error during activity transitions (B-3), and does not rely on information about the patients, such as length of limbs, weights, etc. (B-4). These are in contrast to some other methods in the scientific field.

\(^3\)Cf. the list in section 1.2.

\(^4\)Cf. the list of drawbacks in section 1.3.
(e) The gait inference model for several ambulatory modes is encapsulated into one single neural network. Other approaches often use different mid-level controllers for different gait phases and activity modes.

(f) The GaIn article was published in the Sensors journal, which is ranked as Q2 by Scopus.

Table 24: Summary of the support publications for this dissertation.

<table>
<thead>
<tr>
<th>Article title</th>
<th>Scopus Journal Quartile</th>
<th>Citations</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>HuGaDB: human gait database for activity recognition from wearable inertial sensor networks (Best talk award)</td>
<td>Q2</td>
<td>5</td>
<td>[26]</td>
</tr>
<tr>
<td>Roman Chereshnev and Attila Kertész-Farkas</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lecture Notes in Computer Science</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RapidHAR: a computationally inexpensive method for realtime human activity recognition from wearable sensors</td>
<td>Q3</td>
<td>1</td>
<td>[28]</td>
</tr>
<tr>
<td>Roman Chereshnev and Attila Kertész-Farkas</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Journal of Ambient Intelligence and Smart Environments</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GaIn: human gait inference for lower limbic prostheses for patients suffering from double trans-femoral amputation</td>
<td>Q2</td>
<td>N/A&lt;sup&gt;5&lt;/sup&gt;</td>
<td>[27]</td>
</tr>
<tr>
<td>Roman Chereshnev and Attila Kertész-Farkas</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sensors</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<sup>1</sup>The PhD candidate is the main author in all of these articles.

<sup>2</sup>All articles have been published in international research journals in English language.

<sup>3</sup>Ranking is based on Scopus.

<sup>4</sup>Independent citations only, as of March 2019.

<sup>5</sup>It has been published very recently.
Acknowledgments

First and foremost, I would like to thank my supervisor, Dr. Attila Kertész-Farkas. This work would not have been possible without his useful comments, his constant encouragement to pursue my studies, and our fruitful discussions. I am deeply grateful to him for the time that he spent helping me. It was an honor to be his PhD student.

I would like to thank all the people who participated in the data collection for HuGaDB. Thanks to each of them for taking the time to voluntarily participate in the data collection procedure. Furthermore, I would also like to thank Elena Artemenko for drawing the concept of the robotic leg. I thank Timur Bergaliyev and his lab members Sergey Sakhno and Sergey Kravchenko from the Laboratory of Applied Cybernetic Systems at BiTronics Lab for their technical support on using sensors.

I would also like to express my gratitude to Elena Dmitrieva and Emin Abdullaev for scrutinizing and correcting the Russian version of the summary from a linguistic point of view.

Last but not least I would like to thank all my family members: my father, my grandmother, and my brother. I deeply thank them all for their unconditional support during the writing of this thesis.
# List of abbreviations and conventions

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADL</td>
<td>Activities daily living</td>
</tr>
<tr>
<td>ANN</td>
<td>Artificial neural network</td>
</tr>
<tr>
<td>CLME</td>
<td>Complementary limb motion estimation</td>
</tr>
<tr>
<td>CMU-MMAC</td>
<td>Carnegie Mellon University multi-modal activity database</td>
</tr>
<tr>
<td>EEG</td>
<td>Electroencephalography</td>
</tr>
<tr>
<td>EMG</td>
<td>Electromyography</td>
</tr>
<tr>
<td>FFT</td>
<td>Fast Fourier transformation</td>
</tr>
<tr>
<td>GaIn</td>
<td>Gait inference</td>
</tr>
<tr>
<td>GMM</td>
<td>Gaussian mixture model</td>
</tr>
<tr>
<td>GR</td>
<td>Gesture recognition</td>
</tr>
<tr>
<td>HAR</td>
<td>Human activity recognition</td>
</tr>
<tr>
<td>HASC</td>
<td>Human activity sensing consortium</td>
</tr>
<tr>
<td>HGA</td>
<td>Human gait analysis</td>
</tr>
<tr>
<td>HGI</td>
<td>Human gait inference</td>
</tr>
<tr>
<td>HMM</td>
<td>Hidden Markov models</td>
</tr>
<tr>
<td>HuGaDB</td>
<td>Human gait database</td>
</tr>
<tr>
<td>IMU</td>
<td>Inertial measurement unit</td>
</tr>
<tr>
<td>LDA</td>
<td>Linear discriminant analysis</td>
</tr>
<tr>
<td>LSB</td>
<td>Least significant bit</td>
</tr>
<tr>
<td>LSTM</td>
<td>Long short-term memory</td>
</tr>
<tr>
<td>MAREA</td>
<td>Movement analysis in real-world environments using accelerometers dataset</td>
</tr>
<tr>
<td>ML</td>
<td>Machine learning</td>
</tr>
<tr>
<td>OUAR</td>
<td>Object usage based HAR</td>
</tr>
<tr>
<td>PAMAP2</td>
<td>Physical activity monitoring for aging people dataset</td>
</tr>
<tr>
<td>PCA</td>
<td>Principal component analysis</td>
</tr>
<tr>
<td>PSAR</td>
<td>Physical sensor activities recognition</td>
</tr>
<tr>
<td>ReLU</td>
<td>Rectified linear unit</td>
</tr>
<tr>
<td>RFID</td>
<td>Radio frequency identification</td>
</tr>
<tr>
<td>RNN</td>
<td>Recurrent neural networks</td>
</tr>
<tr>
<td>SVM</td>
<td>Support vector machines</td>
</tr>
<tr>
<td>STD</td>
<td>Standard deviation</td>
</tr>
<tr>
<td>USC-HAD</td>
<td>University of Southern California human activity dataset</td>
</tr>
<tr>
<td>WARD</td>
<td>Wearable Action Recognition Database</td>
</tr>
<tr>
<td>WEKA</td>
<td>Waikato environment for knowledge analysis</td>
</tr>
<tr>
<td>WSAR</td>
<td>Wearable sensor based HAR</td>
</tr>
</tbody>
</table>
Bibliography


List of figures

1. Phases of a full gait cycle during locomotion. Image source: [139].
2. Concept of robotic prosthetic legs for patients suffering from double trans-femoral amputation. Circles show the location of EMG sensors, and boxes show the location of accelerometers and gyroscopes.
3. Correlation between shank and thigh movement over several gait cycles in different activities. The angles of the thigh and shank are measured to the horizontal line.
5. The overview of the recognition algorithm proposed by Zhu et al. Image source: [165].
6. Illustration of the augmented-feature extraction for activity-recognition method. Image source: [82].
8. Location of IMU sensor in USC-HAD. Image source: [163].
9. Location of sensors in MAREA dataset. Image source: [83].
10. Classification performance (AUC of ROC) of different number of GMM components with PCA and LDA dimension reduction to (A) one, (B) two, and (C) three dimensions for frame length. Note the y-axis scaling for each dimension. Image source: [143].
11. Visualization of four states of the P300 system. A fifth state is detected by the system when the participant is not looking at the screen. Image source: [36].
12. A finite-state model of gait. Each box represents a state and the transition conditions between states are specified. Image source: [135].
13. Biomechanical knee torque data (target, blue) and estimate (prediction, red) within a full gait cycle. Image source: [39].
14. Estimates of the hip ($\varphi_h$) and knee angles ($\varphi_k$) based data obtained from the same participant. PCA reconstruction is based on angle data (A), angles and angular velocities (B), angles, angular velocities and angular accelerations (C). The number of principal components was 2, 4, and 5. The dashed blue line is the original trajectory, the black dots indicate estimates. Image source: [142].
15. Estimates based on averaged reference data. (A) Estimation of the hip ($\varphi_h$) and knee angles ($\varphi_k$). (B) Kalman filtering of the knee angle ($\varphi_k$) and the angular velocity ($\dot{\varphi}_k$). The dashed blue line is the original trajectory, the black dots the estimation, and the solid red line shows the estimation obtained with Kalman-filter. Image source: [142].
Estimation of knee motion only. Depicted are estimates of the knee angle and angular velocity based on the data obtained from the same participant (A), and on averaged gait data (B). The number of principal components used is 7 in both cases. The dashed blue line is the original trajectory and the solid red line is estimation obtained with Kalman filter. The source of images: [142].

Human gait data estimation of healthy participants. Image source: [141].

Knee angle trajectories during treadmill walking with the C-Leg (A) and with a CLME-controlled active knee joint (B). Sound and prosthetic knee joint angles are normalized. Image source: [142].

Activity transition graph of the GaIn control system.

Gait cycle variance during walking. (A) Gait cycles produced by the same user multiple times. (B) Gait cycles produced by different users. Legend indicates the source of the data. Data are scaled to the range \([-1, +1]\). Data are taken from the HuGaDB dataset [26].

Screenshot of GaIn during gait inference. Around 56 data frames add up to 1 second. See the full video at: https://youtu.be/aTeYPGxncnA.

Gait inference and activity recognition using GaIn.

Location of sensors. (A) EMG sensors are shown as circles, while boxes represent inertial sensors. (B) Mounted sensors. The black waist bag contains the Arduino driver.

Screenshot of the CollectionContinuesGUI program.

LowerLimbActivityTesting screenshot while data collection.

The screenshot of the Editor program during choosing files for data format conversion.

Data visualization for normalization data from initial sensors were divided by 32768 and data from EMG were subtracted by 128 and divided by 128.

Screenshot of a HuGaDB data file.

Illustration of an unfolded dynamic Bayesian network w.r.t. an activity series.

Accuracy w.r.t. the number of Gaussian components and the length of the context window.

$F_1$ scores w.r.t. the number of Gaussian components and the length of the context window.

Continuous activity recognition.

Activity recognition at 35.6 s enlarged from Figure 30. The line represents the x-axis acceleration value recorded by accelerometer located on thigh.

Data from x- and z-axis accelerometer located on left thigh. Data from y-axis accelerometer were nearly constant and thus are not shown.

Learning curve for early stopping. Training terminated after epoch 80 because of lack of improvement on the validation set.

(A) EMG signal with artifacts. (B) same EMG signal after derivation.
Difference between X-axis of left and right tights accelerometer (A) during standing up and sitting down (notice the legs are parallel), (B) during standing and walking. Standard deviation with context window equal to 5 signals of difference between X-axis of left and right tights accelerometer (C) during standing up and sitting down, (D) during standing and walking.

Activity recognition in GaIn with EMG signal with artifact (A), weak signal (B), weak signal with artifact (C), and good signal (D). Signals obtained from participants ID=1, 7, 12, 16, respectively. Note that plot D consists of three continuing panels.

Activity recognition latency in seconds (s) for standing up (A) and sitting down (B).

Illustration of a folded (A) and an unfolded (B) recurrent neural network structure.

Structure of LSTM cell. The operators ·, +, [..], and tanh in boxes represent element-wise multiplication, addition, concatenation, and tanh operations on vectors, respectively.

GaIn inference for various walking-related activities. The activity is indicated by the background color for the reader, but this information was not used for prediction. The shank degree is predicted based on thigh angles (not shown). Solid black line shows the predicted, the dashed line shows the true angles of the right shank, while the shaded area between them indicates the prediction error. Plots for the left leg are similar.

Predicted and true shank angles as function of thigh position over several gait cycles in different activities. The input data used was the same as at Figure 3. The caption below each figure indicates the types of activities performed on the plot.

GaIn inference for walking and going up for participant ID=1. (A) gait inference for left leg, (B) gait inference for right leg. The activity is indicated by the background color for the reader, but this information was not used in the methods. The shank degree is predicted based on thigh angles (not shown). Solid black line shows the predicted, the dashed line shows the true angles of the right shank, while the shaded area between them indicates the prediction error. Plots for the left leg are similar.

GaIn inference for walking and going down for participant ID=6. (A) gait inference for left leg, (B) gait inference for right leg.

GaIn inference for walking and running for participant ID=11. (A) gait inference for left leg, (B) gait inference for right leg.

Shank angles of different participants in stance phase (A) and swing phase (B).

Participant 1 GaIn inference for walking and running. (A) gait inference for left leg, (B) gait inference for right leg.

Participant 1 GaIn inference for walking and going down. (A) gait inference for left leg, (B) gait inference for right leg.

Participant 6 GaIn inference for walking and going up. (A) gait inference for left leg, (B) gait inference for right leg.
51 Participant 6 GaIn inference for walking and running. (A) gait inference for left
leg, (B) gait inference for right leg. .............................................. 114
52 Participant 8 GaIn inference for walking and going down. (A) gait inference for left
leg, (B) gait inference for right leg. .............................................. 115
53 Participant 8 GaIn inference for walking and running. (A) gait inference for left
leg, (B) gait inference for right leg. .............................................. 115
54 Participant 8 GaIn inference for walking and running. (A) gait inference for left
leg, (B) gait inference for right leg. .............................................. 116
55 Participant 10 GaIn inference for walking and going down. (A) gait inference for
left leg, (B) gait inference for right leg. .............................................. 117
56 Participant 10 GaIn inference for walking and going down. (A) gait inference for
left leg, (B) gait inference for right leg. .............................................. 117
57 Participant 11 GaIn inference for walking and doing down. (A) gait inference for
left leg, (B) gait inference for right leg. .............................................. 118
58 Participant 11 GaIn inference for walking and doing up. (A) gait inference for left
leg, (B) gait inference for right leg. .............................................. 118
59 Participant 1 GaIn inference for one leg walking and going up. ......................... 119
60 Participant 1 GaIn inference for one leg walking and going down. .................... 119
61 Participant 1 GaIn inference for one leg for walking and running. .................... 119
## List of tables

<table>
<thead>
<tr>
<th>Table No.</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Comparison of RL by Wen et al. [151] against GaIn in walking activity.</td>
<td>11</td>
</tr>
<tr>
<td>2</td>
<td>Comparison of RL by Wen et al. [151] against GaIn in walking activity.</td>
<td>36</td>
</tr>
<tr>
<td>3</td>
<td>Characteristics of HuGaDB.</td>
<td>47</td>
</tr>
<tr>
<td>4</td>
<td>HuGaDB raw signals.</td>
<td>48</td>
</tr>
<tr>
<td>5</td>
<td>Characteristics of the HuGaDB participants.</td>
<td>52</td>
</tr>
<tr>
<td>6</td>
<td>Description of the file naming convention.</td>
<td>52</td>
</tr>
<tr>
<td>7</td>
<td>Description of the data file header.</td>
<td>53</td>
</tr>
<tr>
<td>8</td>
<td>Results of activity recognition.</td>
<td>59</td>
</tr>
<tr>
<td>9</td>
<td>Continuous activity recognition allowing border tolerance.</td>
<td>61</td>
</tr>
<tr>
<td>10</td>
<td>Results of activity recognition with directional features.</td>
<td>62</td>
</tr>
<tr>
<td>11</td>
<td>Transition matrix for hidden Markov model.</td>
<td>62</td>
</tr>
<tr>
<td>12</td>
<td>HMM grid search result.</td>
<td>63</td>
</tr>
<tr>
<td>13</td>
<td>Artificial neural network grid search result.</td>
<td>65</td>
</tr>
<tr>
<td>14</td>
<td>Recurrent Neural network.</td>
<td>66</td>
</tr>
<tr>
<td>15</td>
<td>Main classification results.</td>
<td>66</td>
</tr>
<tr>
<td>16</td>
<td>Features used in GaIn.</td>
<td>69</td>
</tr>
<tr>
<td>17</td>
<td>Classification results for each participant.</td>
<td>71</td>
</tr>
<tr>
<td>18</td>
<td>Gait inference error.</td>
<td>78</td>
</tr>
<tr>
<td>19</td>
<td>GaIn inference error.</td>
<td>78</td>
</tr>
<tr>
<td>20</td>
<td>Average shank degree prediction error at activity transitions.</td>
<td>80</td>
</tr>
<tr>
<td>21</td>
<td>Gait inference error for one leg prediction.</td>
<td>83</td>
</tr>
<tr>
<td>22</td>
<td>GaIn inference error in stance and swing phases for one leg prediction.</td>
<td>83</td>
</tr>
<tr>
<td>23</td>
<td>Average shank degree prediction error at activity transitions for one leg prediction.</td>
<td>84</td>
</tr>
<tr>
<td>24</td>
<td>Summary of the support publications for this dissertation.</td>
<td>88</td>
</tr>
</tbody>
</table>
Appendix A. GaIn system inference visualization

In this section the reader can find additional gait inference results obtained with GaIn for various participants, legs and activity modes.

Figure 48: Participant 1 GaIn inference for walking and running. (A) gait inference for left leg, (B) gait inference for right leg.
Figure 49: Participant 1 GaIn inference for walking and going down. (A) gait inference for left leg, (B) gait inference for right leg.
Figure 50: Participant 6 GaIn inference for walking and going up. (A) gait inference for left leg, (B) gait inference for right leg.
Figure 51: Participant 6 GaIn inference for walking and running. (A) gait inference for left leg, (B) gait inference for right leg.
Figure 52: Participant 8 GaIn inference for walking and going down. (A) gait inference for left leg, (B) gait inference for right leg.

Figure 53: Participant 8 GaIn inference for walking and running. (A) gait inference for left leg, (B) gait inference for right leg.
Figure 54: Participant 8 GaIn inference for walking and running. (A) gait inference for left leg, (B) gait inference for right leg.
Figure 55: Participant 10 GaIn inference for walking and going down. (A) gait inference for left leg, (B) gait inference for right leg.

Figure 56: Participant 10 GaIn inference for walking and going down. (A) gait inference for left leg, (B) gait inference for right leg.
Figure 57: Participant 11 GaIn inference for walking and doing down. (A) gait inference for left leg, (B) gait inference for right leg.

Figure 58: Participant 11 GaIn inference for walking and doing up. (A) gait inference for left leg, (B) gait inference for right leg.
Figure 59: Participant 1 GaIn inference for one leg walking and going up.

Figure 60: Participant 1 GaIn inference for one leg walking and going down.

Figure 61: Participant 1 GaIn inference for one leg for walking and running.