Design Considerations and Application Examples for Embedded Classification Systems

Gesichtspunkte zum Entwurf und Anwendungsbeispiele eingebetteter Klassifikationssysteme

Der Technischen Fakultät der
Friedrich-Alexander-Universität Erlangen–Nürnberg

zur Erlangung des Grades

DOKTOR–INGENIEUR

vorgelegt von

Ulf Jensen

aus Offenbach a.M.

Als Dissertation genehmigt von der
Technischen Fakultät der
Friedrich-Alexander-Universität Erlangen–Nürnberg

Tag der mündlichen Prüfung:	27.01.2016
Vorsitzender des Promotionsorgans:	Prof. Dr. Peter Greil
Berichterstatter:	Prof. Dr. Björn Eskofier
	Prof. Dr. Thomas Jaitner

Zusammenfassung

Tragbare Sportler-Assistenz-Systeme sind eine weit verbreitete Technologie zur Leistungssteigerung im Sport. Die komplexe Signal- und Datenverarbeitung auf diesen Systemen wird dabei oft mit Methoden der Mustererkennung, wie Klassifikation, bewerkstelligt. Die Implementierung von Klassifikationsalgorithmen auf mobiler Hardware nennt man eingebettete Klassifikation. Technische Herausforderungen entstehen hierbei durch die limitierte Rechenleistung, Batteriekapazität und Größe mobiler Systeme.

Der Accuracy-Cost-Tradeoff beschreibt die beiden konkurrierenden Entwurfsziele von eingebetteten Klassifikationssystemen: die Genauigkeit und die Berechnungskosten eines Klassifikators. Ein Datenanalyse-System soll die maximale Genauigkeit erreichen und gleichzeitig auf einem mobilen System mit limitierten Ressourcen implementiert werden können. Genauigkeit und Kosten müssen daher in der Entwurfsphase von Algorithmen parallel betrachtet werden. Zudem ist es nötig, Klassifikationssysteme akkurat zu Modellieren, ihre zu erwartende Genauigkeit präzise abzuschätzen und ihre Energieeffizienz im Entwurf zu berücksichtigen. Das erste Ziel dieser Arbeit war es, die Lösung des Accuracy-Cost-Tradeoffs beim Entwurf eingebetteter Klassifikationssysteme mit verschiedenen Gesichtspunkten zu unterstützen.

Der Erfolg tragbarer Systeme im Sport basiert auf verschiedenen Chancen dieser Technologie. So kann ein tragbares System Felddaten aufzeichnen und ist nicht auf eine Laborumgebung oder einen Messplatz beschränkt. Es kann eine große Datenmenge mit realitätsnaher Variation genutzt werden, um die Leistung im Feld und deren Langzeitverlauf zu analysieren. Die Verarbeitung der Daten in Echtzeit erlaubt es, den Sportler und den Trainer im Feld mit extrinsischem Feedback zu unterstützen. Tragbare Systeme eröffnen daher neue Möglichkeiten sportliche Leistung zu analysieren. Das zweite Ziel dieser Arbeit war es, diese Möglichkeiten anhand von drei Anwendungen zu verdeutlichen um den Mehrwert der Systeme zu unterstreichen.

In dieser Arbeit werden mobile Anwendungen für plyometrisches Training, Golf Putting und Schwimmen vorgestellt. Sie bestehen aus Netzwerken drahtloser, körpernaher Inertialsensoren mittels derer kinematische Daten mobil verarbeitet werden können.

Die Anwendung für plyometrisches Training berechnete die Bodenkontaktzeit bei Niedersprügen. Der Datenanalyse-Algorithmus verwendete dabei ein Hidden Markov Modell, um den auftretenden Bodenkontakt mit

hoher Genauigkeit zu ermitteln. Die Bodenkontaktzeit ist ein wichtiger Trainingsparameter zur Beurteilung der Sprungleistung und visuell nicht zu ermitteln.

Die Golf Putting Anwendung beschreibt ein System zum Techniktraining beim Golf. Der entwickelte Algorithmus zur automatischen Erkennung von Schlägen und Berechnung von Schlagparametern stellt die Basis für Anwendungen mit extrinsischem Feedback dar. Langzeitaufzeichnungen von kinematischen Daten während des Golftrainings lieferten im Rahmen dieser Arbeit neue Einblicke in die Entwicklung der Schlagtechnik.

Die Schwimmanwendung stellt ein dezentes System zur Trainingsüberwachung vor. Es war in der Lage die vier verschiedenen Schwimmstile sowie Wenden und Pausen anhand der Kopfkinematik zu unterscheiden. Außerdem konnte mit dem System die Langzeitermüdung eines 90-minütigen Schwimmtrainings erkannt werden. Die Erkennung der Schwimmstile, Wenden und Pausen wurde als eingebettetes Klassifikationsystem auf einem tragbaren Sensorknoten implementiert. Diese Anwendung unterstreicht daher die Eignung der entwickelten Gesichtspunkte zur Lösung des Accuracy-Cost-Tradeoffs.

Diese Arbeit stellt verschiedene Gesichtspunkte zum Entwurf eingebetteter Klassifikationssysteme vor und unterstreicht anhand von Anwendungsbeispielen den Mehrwert tragbarer Sportler-Assistenz-Systeme. Der Fokus lag auf einer anwendungsorientierten Unterstützung zur Lösung des Accuracy-Cost-Tradeoffs und auf der technischen Umsetzung tragbarer Sportler-Assistenz-Systeme. Von den Erkenntnissen dieser Arbeit können sowohl Forschung als auch Industrie aus den Bereichen Sport, Gesundheit und Medizin profitieren.

Abstract

Wearable athlete support systems are a popular technology for performance enhancement in sports. The complex signal and data analysis on these systems is often tackled with pattern recognition techniques like classification. The implementation of classification algorithms on mobile hardware is called embedded classification. Thereby, technical challenges arise from the restricted computational power, battery capacity and size of such mobile systems.

The accuracy-cost tradeoff describes the two conflicting design goals of embedded classification systems; the accuracy and the computational cost of a classifier. A data analysis system should be as accurate as possible and, at the same time, as cheap as required for an implementation on a mobile system with restricted resources. Thus, accuracy and cost have to be simultaneously considered in the design phase of the algorithms. Furthermore, an accurate modeling of classification systems, a precise estimation of the expected accuracy and energy efficient algorithms are needed. The first main goal of this thesis was to develop design considerations to support the solution of the accuracy-cost tradeoff that occurs during embedded classification system design.

The success of wearable technology in sports originates in several application opportunities. A wearable system can collect data in the field without the restrictions of a lab environment or capture volume. Data with realistic variation and a high number of trials can be used to analyze the true field performance as well as its long-term progress over time. Real-time data processing allows to support athletes and coaches with augmented feedback in the field. Therefore, wearable systems enable new opportunities for the analysis of athletic performance. The second main goal of this thesis was to illustrate the benefits and opportunities of wearable athlete support systems with three application examples.

This thesis presents applications for plyometric training, golf putting and swimming. The applications were realized as body sensor networks (BSNs) and analyzed kinematic data that were acquired with inertial measurement units (IMUs).

The plyometric training application targeted the ground contact time calculation in drop jump exercises. The presented algorithm used a hidden markov model to calculate the ground contact with high accuracy. The ground contact time is an important training parameter to assess athlete performance and cannot be visually assessed.

The golf putt application realized a system for technique training in the field. It featured an algorithm for automatic putt detection and parameter extraction as a basis for augmented feedback training applications. Long-term kinematic golf training data provided new insights in the progress of golf putting.

The swimming application realized an unobtrusive swimming exercise tracker. The system was able to classify the four swimming styles, turns and breaks based on the head kinematics. Furthermore, an algorithm to classify long-term fatigue that occurs in a 90 min swimming exercise was presented. The swimming style, turn and break classification was implemented as embedded classification system on a BSN sensor node. This application underlines the applicability of the presented design considerations for solving the accuracy-cost tradeoff.

The contributions of this thesis are considerations for the design of embedded classification systems and application examples that illustrate the benefits of wearable athlete support systems. The focus was to provide practical support for the solution of the accuracy-cost tradeoff and to present the technical realization of wearable athlete support systems. The findings of this thesis can be beneficial for research as well as industrial applications in the sport, health and medical domain.

Acknowledgment

I would like to acknowledge the Braun GmbH for providing financial support for my initial industry-related research project. I would like to thank particularly Olaf Sörensen for his technical and organizational support during our project and his personal support afterwards. I would have never imagined to work on such a crazy and extraordinary project and I am really proud of having contributed to the resulting product.

I would like to acknowledge the Bavarian Ministry for Economic Affairs, Infrastructure, Transport and Technology and the European Fund for Regional Development that funded our research project "miLife - eine innovative Wearable Computing Plattform zur Datenanalyse von in Kleidung integrierten Sensoren für den Einsatz in Teamsport und Gesundheit". I would like to thank my research partners at Astrum IT and the University Hospital Erlangen for their support throughout the project. What we achieved would not have been possible without the huge effort they put into the project. In particular, I would like to thank my colleagues Heike Leutheuser and Dominik Schuldhaus for their project work and support in project management. Great job, thank you!

I would like to acknowledge the adidas innovation team (a.i.t.) for their support in different projects but in particular for the effort they put in the miLife project. It was great to work with you and I will also remember the time we laughed, played soccer or beach volleyball, went skiing and had a beer. I would especially like to thank Heiko Schlarb (Mr. Everything) for our helpful discussions and his outstanding support with everything. I would also like to thank Berthold Krabbe for his overall support and the unforgettable skiing trips. Dear adidas people, thank you for having me as a part of your team.

I would like to acknowledge my fellow researchers from the Institute for Sports Sciences at TU Dortmund for their support in our golf project. Thank you for having me as a guest and for your great contribution to my research. Dear Marcus and Markus, thank you for the thesis review and the game experience. Dear Thomas, thank you for volunteering as second referee for this thesis.

I would like to acknowledge all students I worked with throughout my research. These were Franziska Prade, Birgit Roithner, Anja Pohan, Fabian Weilbrenner, Andreas Decker, Felix Durlak, Jan Sandvoss and Axel Heinrich. The outcome of your theses or your student assistant work lives on in several publications and this thesis. You can be proud of this. Thank you for your great work and sorry for pushing so hard sometimes.

In particular, I would like to thank Matthias Ring and Alexander Ruppel for their outstanding work and contribution to my research. You guys were amazing.

I would like to acknowledge the whole team of the Pattern Recognition Lab for amazing four and a half years. Although it was exhausting from time to time, you were the outstanding environment for my research. Thank you for all the ideas, discussions and the advices you shared with me to improve my research. Thank you in particular for all the social events like exercising and partying. In particular, I would like to thank the computer vision and the speech group that I shared a special connection with. Thank you for sheltering the sports guys at the very beginning and for teaching me how to use feature selection, cross-validation, hidden markov models and linux.

I would like to acknowledge my fellow digital sports team members for all their support and their thesis review. Above all, I want to thank you for tolerating the huge amounts of irony, cynicism and dumb jokes that I generated every day. It was an outstanding pleasure to be a part of this team! I would particularly like to thank Joachim Hornegger for establishing the digital sports group and for giving me the chance to work at the lab. Dear Achim, without you, I wouldn't be where I am now.

There are two persons that highly contributed to this thesis. I would like to acknowledge Patrick F.-X. Kugler and Björn M. Eskofier for their roles as mentors, supervisors, colleagues but also friends. Dear Patrick, thank you for kickstarting my pattern recognition knowledge (actually my whole technical knowledge) and being a great roommate in the office and in hotels around the globe. Dear Björn, thank you for the morning runs, your support, your trust and for our great technical discussions. Thank you both for talking so much crap, our amazing travels, all the laughs, the big nights out and the unforgettable memories that we share. You made my time at the lab a complete success.

Finally, I would like to thank my family and friends for their support and patience throughout my research. You gave me the desperately needed energy to go on and complete this thesis. You made all this possible and encouraged me to go my way. You were my inspiration and motivation. I wouldn't have been able to complete this thesis without every single one of you.

The idea is really really simple, but what you get is almost like magic!

Ulf Jensen

Contents

Chapter 1

Introduction

This chapter introduces the field of work and the general scope of this thesis. A literature review summarizes the state-of-the-art that this thesis is based on and open research topics are subsequently derived and presented. The chapter further presents the contribution and the structure of this thesis.

First, the thesis is motivated and introduced. Second, state-of-the-art in literature and patents is presented. Third, the contributions of this thesis are summarized. Fourth, the structure of this thesis is presented.

1.1 Overview and Motivation

Coaches constantly work on the improvement of their athletes. Nowadays, many different aspects like nutrition, injury prevention and mental strength have to be considered to be successful in high-performance sports. Beside these emerging factors, traditional components of athletic training like kinesiology and motor learning are still of high importance. Coaches, but also athletes themselves, are seeking for ways to support and optimize athletic training and this is not only the case for professional but also for recreational athletes. Thereby, technical equipment ranging from cheap stop watches to expensive multi-camera systems is used. In the last years, wearable systems for sports and sports medicine became more and more important [Hans 09, Chan 12, Cham 15]. Such systems can be denoted as **wearable athlete support systems** as an athlete wears such a system to be monitored and supported during training and competition.

Thus, wearable technology is of great interest for the sports domain. Athletes are physically moving and therefore put a special demand on any technical system that is used to assess performance. Essential aspects for the technical systems that are used for athlete assessment and support are e.g. form factor, information access and data precision [Mend 92, Wagn 06]. Devices are only useful if they do not hinder the athletes in their performance. Information is only useful if it is accessible at the right time and with the right presentation. Data are only useful if they are precise enough to provide an additional value.

In turn, these requirements for technical systems put high demands on wearable technology and its underlying data analysis. However, wearable athlete support systems also provide several application opportunities compared to stationary systems. In the following sections, the two aspects of technical challenges and application opportunities are described in greater detail.

1.1.1 Technical Challenges

From a technical perspective, wearable athlete support systems are often realized as body sensor network (BSN) or embedded system. A BSN is a network of sensor nodes and processing nodes that are worn on the human body [Hans 09]. The term embedded system describes computing systems that are integrated in an enclosing product and that are often self-sufficient in their operation [Marw 03]. See chapter 3 for a more detailed introduction of these technologies.

These two types of systems have to be wearable and, at the same time, highly precise in data analysis. Therefore, sophisticated methods for data analysis have to run on resource-constraint hardware that is restricted in e.g. RAM and CPU. The field of pattern recognition and, more specifically, pattern classification provides such sophisticated algorithms for precise data analysis. However, algorithms have to be carefully selected and designed to be suitable for embedded data analysis on wearable devices as described in [Esko 09]. The aspect of classification methods on resource-constraint devices will be denoted by embedded classification.

A central design consideration for embedded classification systems can be deduced: the maximum accuracy in data analysis needs to be reached with minimal computational cost as hardware resources are limited due to the wearable system nature. This thesis describes this situation as the accuracy-cost tradeoff. A method to resolve the accuracy-cost tradeoff in the design phase of an embedded classification system is a central aspect of this thesis. The presented research closes the gap between the high-level algorithm complexity analysis using the Landau-notation

[Bach 94] and the low-level code instrumentation approach. The presented mid-level analysis expresses computational cost as number and type of mathematical operations and parameters and enables a simultaneous analysis of classification accuracy as well as computational cost. Beside the accuracy-cost tradeoff, this thesis will also investigate the related aspects of accurate classification accuracy estimation when using feature selection and the accurate modeling of sequential data analysis with hidden markov models (HMMs).

Another central design consideration for BSN applications is energy-efficiency [Hans 12]. Less power consumption can either increase the system runtime or decrease the system form factor as smaller batteries can be used. Energy-efficient solutions are also needed when systems are self-sufficiently powered without batteries using e.g. energy harvesting techniques. Special emphasis has to be put on the wireless radio activity in BSNs as wireless transmission may consume a majority of energy [Hans 12]. This thesis introduces several aspects for the energy-efficient design and implementation of embedded classification algorithms.

Accuracy-cost tradeoff and energy-efficiency are central design considerations for wearable athlete support systems. They need to be considered to ensure system wearability (e.g. form factor) and information fidelity (e.g. classification accuracy) which are in turn mandatory prerequisites for wearable systems.

1.1.2 Application Opportunities

Potentially, wearable systems have several beneficial characteristics regarding information access. Data can be analyzed on wearable systems and results are therefore available in real-time. Data from free field performance can be captured without the restrictions of lab equipment or capture volumes. Data from long-term recordings can be analyzed for trends and progress. Data from a theoretically unlimited amount of trials and conditions are available. Data are objective as they are analyzed from a technical system. All these potential aspects facilitate a different view on athlete performance as actual performance in the field can be assessed over a longer period of time, with realistic variations and instant result availability. Several classes of applications can be deduced from these capabilities and two examples will be presented in the next paragraphs.

One example are augmented feedback applications that provide additional information for the athlete and the coach. Augmented feedback can be useful in situations where the athlete cannot process the required information during performance, when the information cannot be assessed with human senses or when a coach is not available [Sigr 13]. Different aspects like feedback presentation, feedback frequency and the adaption of feedback according to training results need to be considered [Sigr 13]. See chapter 4.2.4 for a more detailed introduction. The augmented feedback capabilities of a wearable system are tightly coupled with real-time data processing.

Another example is long-term training analysis to understand the progress of athletic performance. Coaches are e.g. interested in the most efficient way to turn a novice into an expert. Hence, a good understanding of the crucial aspects differentiating novices and experts is needed and the progress of these aspects needs to be monitored. With wearable athlete support systems and their long-term recording and unlimited trial capability, different insights into the training progress are possible. Athletic performance can be constantly assessed to gain a better understanding of how crucial aspects vary and progress throughout the training process. We believe that this information helps to identify strengths and weaknesses of training progress and supports a more individual kind of training control.

1.1.3 Scope of Applications

This thesis investigates the technical aspect of designing accurate and energy-efficient embedded classification systems as well as the applicational aspects of augmented feedback and long-term training analysis. These aspects will be highlighted by means of three athlete support systems that are implemented as BSN applications.

The presented applications are based on kinematic data that are acquired with inertial measurement units (IMU). IMUs were traditionally used for navigation purposes but gained high popularity for applications in the medical domain in the last 25 years (see e.g. [Luin 05]). The driving force behind their success was the advancing miniaturization of microelectromechanical systems (MEMS) that are e.g. introduced in [Jone 13]. Based on the MEMS technology, small and low-power IMUs that feature accelerometers and gyroscopes are available at low cost. Beside miniaturized IMUs, powerful smartphones and low-power microcontrollers accelerated the advances in wearable technology. See chapter 3 for a more detailed introduction to the mobile sensing and data analysis technology.

This thesis utilizes wearable hardware components (IMU, BSN, smartphones) and contributes sophisticated data analysis algorithms for applications in the field of sports. These applications are plyometric training, golf and swimming. Each application has specific needs and puts specific demands on the underlying system architecture and data analysis. However, the presented solutions can be an inspiration for other applications or other domains.

1.1.4 Related Domains

Several related domains share the same technical challenges and application opportunities with wearable athlete support systems. One example is the activity, health and fitness domain with a growing number of applications like sports apps, health monitors and activity trackers. The user can choose from several smartphone apps (e.g. Runtastic, MyFitnessPal), devices (e.g. FitBit One, Jawbone Up) and platforms (e.g. Google Fit, Apple iHealth). There is a huge interest for measuring all kinds of physical activity and physiological data. These applications combine sports and medicine as recreational athletes, sports novices and patients are target groups. This field might directly benefit from the content of this thesis as algorithms and parameters are highly related. According to [Worl 09], four of the five most prevalent causes of death worldwide are high blood pressure, high blood glucose, physical inactivity and obesity. Physical activity can be used to fight or even cure these health issues. Applications that track activity, health and fitness and motivate people for more physical activity are therefore highly relevant for public health.

In the medical domain, wearable devices are used in applications for heart disease rehabilitation [Para 05], neurological diseases alleviation [Mazi 14] and protheses control [Zhan 12]. Thereby, signal modalities like kinematic, kinetic and physiologic data are analyzed. Although the analysis goals may be different (e.g. rehabilitation, monitoring, diagnosis, ...), applications can benefit from the methods and applications presented in this thesis. Further related domains of embedded classification can be found in chapter 1.2.1.

1.2 Literature and Patents

This section summarizes the state-of-the-art of the main aspects of this thesis. First, prior knowledge regarding the accuracy-cost tradeoff in embedded classification system design is covered. Second, plyometric training and the subsequent measurement of ground contact time is introduced. Third, literature and patents for the kinematic analysis of golf putting are summarized. Fourth, a prior art overview of wearable systems for kinematic swimming analysis is given.

1.2.1 Embedded Classification Literature

Pattern recognition methods are widely used in different data analysis tasks. A categorization and the definition of important terms is given in chapter 2. Commonly, the accuracy of a trained classification system is of main interest. This thesis describes the use of pattern recognition methods on embedded systems as embedded classification. In embedded classification, the computational cost of the analysis algorithm as well as its accuracy are of interest as hardware resources are limited. The computational resources (e.g. RAM, CPU) of embedded systems need to be sufficient to generally run the algorithm or to do so in a given time frame (real-time processing). Thus, a developer of such a system has to consider the two potentially conflicting factors of computational cost and classification accuracy.

The following literature review on the design of embedded classification systems is structured in two parts. First, studies and systems that feature embedded classification are introduced. These articles describe solutions to specific embedded classification problems. Second, literature that describes generic solutions and aspects to realize classification tasks on embedded hardware is introduced. Parts of this review have previously been published in [Jens 16b]. The section closes with deducing open research topics in the field of embedded classification system design.

Studies and Systems

One of the first research articles on embedded classification systems described the recognition of handwritten characters [Kerr 88]. The classification was realized with hierarchical binary decision tree classifiers using different feature sets. It was implemented on an 8-bit microcontroller. Another early work was human liver tissue classification using ultrasound data [Botr 88]. The decision whether tissue was normal or abnormal was taken with a decision rule classifier and a microcontroller-based system was realized.

Examples of industrial applications are a system for herbal remedy detection [Rahm 04] and an electronic tongue system for water classification [Garc 11]. In both cases, an array of sensors analyzed product samples on a portable embedded device that ran an artificial neural network classifier. Furthermore, a comparative study of classifiers was conducted to select the most suitable classifier [Garc 11]. Thereby, the percentage of memory used on a specific hardware platform and the receiver operating characteristics (ROC) curves of the classifiers were compared.

A multi-modal approach to detect the focus of attention in a speech processing application was developed [Hack 06]. The application ran on a mobile phone and used the integrated camera of the mobile to support the detection of "off-talk" which is speech data that were not designated for transmission during a phone call.

A research group described a wearable biosignal classification system capable of detecting the drowsiness of a car driver [Lin 08]. The electric activity of the brain using an electroencephalogram (EEG) was analyzed. Such applications are also denoted by brain computer interfaces (BCI).

The literature contains movement analysis applications, realized as embedded classification systems. One example is a real-time camera system to detect people and analyze their movement [Wolf 02]. These devices were called smart cameras. Another example is the natural control of protheses. Size and weight but also real-time processing of the cyber-physical systems are of great interest as devices have to be portable and react instantly. A system for upper extremity protheses was enhanced with pattern recognition methods to classify electromyographic (EMG) signals for continuous device control [Engl 03]. A similar system for the control of artificial legs has been developed recently [Zhan 12]. Beside the embedded microcontroller for movement control, the system featured a graphical processing unit (GPU) and a display for testing.

A different approach to movement classification was introduced as a gesture recognition system based on inertial measurements [Benb 02]. The algorithm classified hand gestures using accelerometer and gyroscope data. A virtual control device was implemented as sample application and the algorithm was implemented on a mobile personal digital assistant (PDA) with limited hardware resources.

Wearable computing is also an emerging aspect in the field of sports and biomechanics. The authors of [Esko 09] developed an instrumented sports shoe to sense the heel compression during outdoor running and adjust the shoe cushioning according to the surface the athlete is running on. The article also described aspects for the selection of embedded classification algorithms like general consideration of space and memory demand.

Generic Design Aspects

A common approach to estimate computational complexity is the Landau-notation (also: $\mathcal{O}$-notation) [Bach 94]. This measure expresses the complexity of an algorithm in relation to the input data size. While this measure has a profound theoretical background and is widely used to compare algorithms [Corm 01], it is a measure for the scalability of an algorithm and not for the actual costs of an implementation in terms of memory and execution time. As it does not consider the target hardware capabilities, the inspection level of the Landau-notation is too high to give practical and problem-dependent information during classification system design.

Another common approach to estimate computational cost and memory demand is code instrumentation and worst case execution time analysis [Pusc 00, Wilh 08]. In this approach, computation time and memory consumption are measured with a prototype implementation. This procedure is time-consuming and impractical if the decision for a specific algorithm has not yet been taken. Furthermore, measurements influence computation time and memory consumption and the result is strongly target dependent and not directly transferable to other hardware. Hence, it is impractical to use code instrumentation if the target hardware has not been defined yet.

Currently, there exists no practical cost measure for the selection of embedded classification algorithms. The inspection level of existing solutions is either too high (Landau-notation) or too low (code instrumentation). Comparing the number and type of mathematical operations is a common approach to evaluate and select algorithms. Examples can be found for matrix multiplication [Kapo 99], discrete Cosine transform [Chau 95, Vett 84], fast Fourier transform [Vett 84], Euler's transformation of series [Gabu 85] and angular radial transform [Hwan 06].

Specific algorithms were adapted to and implemented on embedded hardware. These findings provide an important building block of embedded classification research as the presented methodologies showed how algorithms have to be adapted to cope with the limited resources of embedded hardware or the real-time constraints of an embedded system. Different examples for the support vector machine (SVM) classifier can be found in the literature. Boni et al. provided a SVM implementation on an 8-bit microcontroller that can be used to turn low-power microcontrollers into smart sensors [Boni 07]. To reduce computational cost, complex mathematical operations were mapped to shift and addition operations. This reduces memory use and the number of multiplication operations

(CORDIC algorithm). Another approach to realize an embedded implementation of the SVM classifier was presented in [Angu 11]. There, the algorithm was described in a hardware description language to be implemented on reconfigurable hardware (field-programmable gate array, FPGA). One of the key aspects for the implementation was the conversion from the real domain to the integer domain. The article also featured and referenced mathematical considerations to adapt the SVM algorithm for an embedded implementation. A related approach with the focus on the hardware-software-codesign of the SVM was presented in [Pede 06]. The design considerations pondered whether functionality should be realized in hardware (better performance) or in software (more flexibility). These research efforts are an important contribution to the field of embedded classification as the SVM is a popular classification algorithm. However, one can never be sure whether a specific classifier (e.g. SVM) performs best in every classification problem (No Free Lunch theorem, [Duda 01]). That is why generally more than one classifier is tested in the system design phase. The presented research, however, only tackled the optimization and realization of a specific algorithm. The following paragraphs describe general considerations and methodologies for resource-constraint systems.

Generic approaches to enable algorithms for a use in embedded classification were described in literature. These are e.g. training subset selection for exemplar learning and model compression that both reduce the complexity of the classification model.

The main challenges for the class of exemplar learning (also: instance-based learning) strategies are the selection of an optimal training data subset or the creation of a suitable abstraction of the training data to reduce the memory demand. At the same time classification accuracy of the classification model should persist. Research efforts targeted the decrease of memory demand while increasing the noise robustness for the nearest neighbor algorithm [Aha 91]. A more recent publication proposed algorithms to reduce the memory demand, customize the exemplar set and increase classification robustness [Jain 08]. The proposed algorithms were also capable of performing incremental learning to gradually adapt the classifier. In summary, the research effort in exemplar learning considered a class of algorithms and mainly optimized their memory demand. It was shown how classification models can be adapted for memory-constraint environments while maintaining their accuracy.

Another possibility to reduce the complexity of classification models is model compression [Buci 06]. The authors described a methodology to train a compact artificial neural network for increased generalizability and decreased model complexity. They further underline the need of such a compression for the class of ensemble classifiers. Results revealed that compressed models reach comparable results consuming less memory and computation time. However, the presented compression approach is limited to a single classifier and a large amount of training data has to be created. Model compression can reduce the memory demand and runtime of algorithms. However, it is not a generic methodology that can be applied to every classification model.

Cost-sensitive feature selection [Tan 93] is a research area that is closely related to the accuracy-cost tradeoff. Classification accuracy is thereby regarded as being dependent on the underlying features. A cost value is associated with the extraction of each feature and the overarching idea is to minimize the cost for taking the right decision. A crucial step is the choice of an adequate cost for the feature extraction. Abstract measures [Iswa 06], specific financial costs [Weis 13] and the computational cost [Tan 93] were proposed. Different challenges like combined cost-sensitive learning/feature selection [Ji 07, Cebe 10], feature groups [Pacl 02] and sequential sensing [Ji 07] were described. Cost-sensitive feature selection can be used for simultaneously reducing classification cost while maintaining or increasing model accuracy. However, the methodology considers only the feature extraction step of an embedded classification system.

An application example to reduce the number of floating point operations per seconds (FLOPS) in a classification system was presented in [Ahma 12]. The authors optimized the tradeoff between the number of FLOPS and the classification accuracy in the feature extraction phase. The computational complexity of each feature was analyzed and used to steer the selection of a feature set. This idea is in line with practical computational complexity measures proposed in [Duda 01]. There, the number and type of basic mathematical operations and the memory needed are mentioned as suitable measures to compare algorithm complexity of specific implementations. However, such a comparison has neither been implemented in a design support tool nor evaluated for its practical applicability. The analysis in [Ahma 12] showed that the method achieved promising results for the feature selection task in a BCI application.

An approach to solve the accuracy-cost tradeoff for a specific classification problem, estimating the running speed of an athlete, was presented in [Chri 13]. The authors measured the runtime of different configurations of the classification algorithm on specific hardware. Thereby, the sampling frequency, sensor axes combination, the feature set and classifier parameters were investigated. The different configurations were compared using the classification accuracy and the machine cycle count. To solve the accuracy-cost tradeoff, the authors used code instrumentation and measured both accuracy and cost with a specific software implementation on dedicated hardware.

Open Research Topics

The literature review revealed different embedded classification applications in various fields like electronic sensing, movement analysis, prothesis control and BCI control. However, in each application, the design and implementation of the pattern recognition system was performed for a specific problem and data characteristic. More general approaches provided embedded implementations for selected algorithms (e.g. SVM, nearest neighbor) or algorithm classes (e.g. exemplar learning) and represent valuable building blocks for future applications. However, specific classifiers might not be the optimal choice according to the No Free Lunch theorem. It was found that feature extraction algorithms have a major impact on memory demand and computation cost of a classification system. The practical analysis of algorithms using mathematical operations for cost-sensitive feature selection [Ahma 12] were the inspiration for a design tool for embedded classification systems. It should provide a comparison of cost (computational cost and memory demand) and accuracy (classification rate) for different classification pipeline configurations (e.g. feature set, classifier). This accuracy-cost comparison can help to select the right algorithms according to the problem at hand. In contrast to [Chri 13], we will present a more generic assessment of the accuracy and cost of classification systems without the need of specific implementation and dedicated hardware. Our analysis will tackle the challenge on a higher level and is therefore designed to support embedded system development in an earlier design phase.

We developed such an analysis methodology [Jens 16b] and automatized the comparison of accuracy and cost in a software package. Chapter 5 describes the methods that were used and introduces the embedded classification software toolbox. An example application that was developed using this methodology can be found in chapter 8.

1.2.2 Embedded Classification Patents

The patent research revealed a high amount of patents that protect apparatuses that implement an embedded classification system. An overview of these patents is beyond the scope of this work. A patent particularly protecting embedded classification or embedded machine learning algorithms was not found.

1.2.3 Plyometric Training Literature

BSN applications can be used for athletic assessment in the field. Trainers and coaches can benefit from a technical system when e.g. performance measures cannot be visually assessed with sufficient precision. This chapter motivates a BSN application for plyometric training and describes the state-of-the-art in the field. For his application, useful fundamentals of kinesiology are described in chapter 4.1.

Plyometric training addresses the speed and strength components of physical fitness. The ability to perform a movement with maximum power in the shortest amount of time is called reactive strength [Wein 10]. The effectiveness of plyometric exercises is determined by the characteristics of the muscle stretch-shortening cycle (SSC). The SSC is influenced by the elastic energy storage of a muscle as well as the activity of the stretch reflex and the golgi tendon reflex. The duration of the SSC determines the biomechanics of the movement and subsequently the training effects for the athlete. These different aspects are described next.

The SSC was categorized in fast (SSC < 250 ms) and slow (SSC > 250 ms) [Schm 92]. When considering the lower extremities, training of a fast SSC is e.g. beneficial for sprinting movements and training of a slow SSC is e.g. beneficial for vertical jump movements [Flan 08]. Thus, plyometric training with a SSC below 250 ms trains the speed component of reactive strength while a SSC above 250 ms trains the strength component of reactive strength. Due to the short duration, the precise and objective SSC measurement is a valuable augmented feedback parameter for coaches and athletes.

In literature, the SSC of the plyometric drop jump exercise was often investigated. This exercise consists of a drop from an elevated box or platform and a subsequent vertical jump. The drop height commonly lies in the range of 20 cm to 60 cm [Wals 04]. The jump is performed as fast and as high as possible and its SSC duration coincides with the ground contact time (GCT) of the jump. In the following, only the GCT will be considered as it coincides with the SSC. It was reported that one effect of plyometric training is a shortening of the drop jump GCT [Maka 10]. Furthermore, it was found that the GCT has the largest effect on power, work and moments during a drop jump [Wals 04]. These findings suggested to enhance the attention on the GCT in plyometric training [Wals 04]. The GCT is also used as part of the reactive strength index (RSI, [McCl 03])

$$\mathrm{RSI} = \frac{\text{Jump Height [m]}}{\text{GCT [s]}} \tag{1.1}$$

that measures plyometric performance [McCl 03].

Traditionally, the GCT of a drop jump is determined by the analysis of the underlying ground reaction forces. Therefore, data are acquired with a force plate or a contact mat and analyzed offline after the training or recording session. Alternative attempts were made to compute the GCT and subsequently the RSI with inertial sensing technology.

In the work of Patterson et al. [Patt 10], the root mean square (RMS)

$$\mathrm{RMS} = \sqrt{x_1^2 + x_2^2 + x_3^2} \tag{1.2}$$

value of a 3-D accelerometer signal $\mathbf{x}$ was calculated, filtered and analyzed with a slope tracking and thresholding method to deduce the GCT. The determination of the GCT for different jumping techniques and varying GCT values as in the aforementioned paper proved to be difficult. The take-off, the point in time when the subject leaves the ground in the countermovement of the jump, was identified as main source of error [Patt 10].

Another publication described a pilot study with ten subjects where the GCT and the subsequent flight time was estimated from 9-DOF inertio-magnetic measurement data [Jait 14]. The on-node implementation of the drop jump analysis algorithm achieved a jump detection rate of 94%, a mean average GCT error of 3.4 ms and a mean average flight time error of 5.2 ms. The article did not provide insights in the embedded data analysis and did not fully describe the experimental evaluation.

The literature review revealed that the drop jump exercise GCT is a crucial parameter for reactive strength training. A mobile computation for training in the field without the need of a stationary force plate or pressure mat is of great interest as coaches cannot estimate these parameters by visual inspection. BSNs with their wearable nature and wireless data transmission capabilities enable such a wearable athlete support system.

We developed an alternative method to compute the drop jump GCT with a BSN application [Jens 14b]. Chapter 6 introduces an application that computed the drop jump GCT from 6-D inertial sensor data using a HMM.

1.2.4 Golf Putting Kinematics Literature

BSN applications can be used as mobile data logger to capture athletic performance in the field. When performance analysis runs on the mobile system, BSNs can even be used as augmented feedback application. This chapter reviews prior art of golf putt analysis systems that compute kinematic parameters within the BSN application. Such a system enables the collection of a large amount of data for training progress analysis as well as feedback training. The chapter further summarizes the state-of-the-art in kinematic putting analysis. For this application, useful fundamentals in motor learning and control are available in chapter 4.2.

Golf putting takes place on the green, the low cut smooth grass section of the golf course situated around the hole. Normally, putting is the final approach to the hole. It comprises a slower and more fine-grained movement execution than the golf swing. Putting was investigated from different disciplines of sports science. Sports economics experts showed that the improvement in putting performance is most beneficial for tournament success [Alex 05]. Training experts concluded that different aspects like green reading, putter geometry and movement execution determine the putt distance and direction accuracy and therefore the outcome of the putt [Karl 10]. The controlled and precise movement proved to be suitable to study motor control and learning strategies that give a valuable insight in human motor learning [Crai 00].

The following literature research focuses on the movement execution aspect for performance optimization in putting. These kinematic aspects are also the main interest to study motor learning. The remainder of this section is structured in four parts. First, the literature of putting technique and motor control is summarized. Second, the aspect of internal and augmented feedback in motor learning and training practice is covered. Third, systems for kinematic analysis are introduced. Fourth, open research topics in the kinematic analysis of golf putting are deduced. Parts of this section have previously been published in [Jens 15b].

Putting Technique and Motor Learning

The literature reveals different aspects of movement control in golf putting. In early findings, different putting styles were compared regarding their performance outcome considering the distance from the hole [Gwyn 93].

Thereby, the putting result for different techniques was compared but the kinematic parameters were not investigated. The authors hypothesized there was no perfect putting style that is suitable for every individual. These results are in line with a more recent study that showed that individuals do not always use the same technique and that technique does not necessarily correlate with putting performance or general playing ability (handicap) [McLa 13].

Different studies investigated the kinematic differences between novices and experts [Dela 97, Para 00, Lee 08b, Sim 10]. One goal was to elicit the key aspects of putting and use these findings to substantiate motor learning theories or deduce training focus.

In [Para 00], club head and body movements were investigated and the authors found nine statistically significant parameters describing the differences between experts and novices. These were mainly displacement parameters, which means that the movement amplitude of the experts and novices varies.

Other studies compared novices and experts to answer the question of how force and amplitude are modulated for different putt distances [Dela 97, Sim 10]. The results revealed that the distance is controlled with kinematic parameters like movement amplitudes and velocity profiles. It was further stated that the movement variability differs between experts and novices. Additionally, it would be of great interest for the field to assess the kinematic training progress of novices [Sim 10]. Thus, researchers are also interested in the learning path, the progress during skill learning, and not only in the outcome of a learning process. This learning path is expected to give insights into the process of a novice becoming an expert. As traditional video processing is labour intensive, the analysis of large data sets representing the training progress is a challenging task.

Another study compared experts with less skilled golfers to analyze their head movement [Lee 08b]. The authors investigated whether an allocentric (moving head and club in the same direction) or egocentric (moving the head in opposite direction to the club) coordination was used. In their study, the authors challenged the general teaching advice of minimizing head movements. They proposed to practice an egocentric coordination as it is more natural and easier to learn than a no-head-movement approach.

Aspects of Feedback

The golf putting literature contains different results regarding the importance of task-intrinsic feedback sources [Aksa 83, Coel 00]. The results by Aksamit et al. implied that kinesthetic feedback is of equal or greater importance than visual feedback for novice players [Aksa 83]. The study investigated the putting accuracy of beginners under the conditions of eyes on ball, eyes on target and occluded vision. It was hypothesized that blindfolded players develop a different strategy for movement control and that occluding the eyes in an early learning stage might be beneficial to reduce the overload of information processing.

In contrast, the results by Coello et al. implied that vision is the essential feedback modality and kinesthetic feedback is a weak source of information for expert players [Coel 00]. Their findings are based on an experiment where subjects wore special glasses that occluded the putting path. Only the ball on impact and the hole were visible for them. These findings underline the different needs for task-intrinsic feedback for different skill levels. Sources for augmented feedback were not investigated regarding their value in different learning stages.

A practical aspect of task-intrinsic feedback in training, the most beneficial frequency of terminal feedback, was discussed in [Ishi 08]. The study in the aforementioned paper investigated different feedback frequency setups as the outcome of the putt was occluded from the player in a defined number of putts. The result was that a reduced relative frequency of terminal feedback was most beneficial in learning the accuracy of the golf putt. It is assumed that more frequent feedback promotes that subjects rely on the feedback and a reduced frequency stimulates the kinesthetic system of the player.

A related research aspect in motor learning and feedback is the focus of attention. Generally, it is distinguished between internal and external focus. External focus means to direct the learner to the effect of their movement whereas internal focus means to direct the learner to the movement itself. These conditions were compared in a golf pitch task in the field with either focus on the arm movement (internal) or the club swing (external). Results revealed that learning was more effective when instructing subjects to use the external focus of attention [Wulf 99]. Further studies illustrated that the external focus condition is also more beneficial for expert players and also more beneficial than no focus restriction at all [Wulf 07].

A kinematic augmented feedback training application and the evaluation of feedback training was introduced in [Kooy 13]. The presented application was capable of training the putt tempo, a kinematic parameter that is defined as the ratio between backswing and foreswing duration. Gyroscope data were collected at the club shaft and subsequently analyzed to extract the putt phases. From the phase information, the tempo parameter was deduced. Augmented feedback was given with a PC interface. The research study showed that subjects were able to improve the value of the tempo parameter.

Systems for Kinematic Recording and Analysis

Different modalities were used to capture the kinematics of putting. The literature contains e.g. marker-based video tracking [Sim 10], infrared marker tracking [Lee 08b], ultrasound signal analysis [Marq 07], posturography with pressure mats [McLa 08] and inertial sensor analysis [King 08]. We will focus on mobile systems that analyze kinematic parameters and are capable of augmented feedback. A special emphasis is put on systems containing IMUs.

A widely used system is the SAM PuttLab® (Science&Motion, Rüsselsheim, Germany) [Marq 07]. The kinematic measurement is based on an ultrasound sender unit on the club shaft and a receiver unit that has to be positioned close to the playing area. Data are streamed to a laptop where they were analyzed. Due to the measuring principle, the system is sensitive to loud noise and wind. In [Marq 07], a system resolution and precision of 0.1 mm and 0.1° were reported. The system collects different kinematic parameters that are analyzed regarding deviation from the optimal value (score) and intra-individual variation (consistency). Some parameters like stroke path and point of impact on the putter face need calibration measurements before data collection. Additionally, putts need to be performed from a predefined distance (4 m). To generate feedback, a large database of professional players was created and serves as basis for the feedback from the system. Drawbacks of the system are its stationarity and sensitivity to loud noise or wind. Furthermore, the putt detection algorithm was not described in detail. It might not be suitable for non-experienced players as the described training data were collected with professional players.

Another training system for golf putt kinematics, TOMI® (Pure Motion Inc., Southlake, USA), was validated for system performance [MacK 10]. The kinematic measurement is based on an infrared light sender unit on the club shaft and a receiver unit. Data are streamed to a laptop where they are analyzed. The validity and reliability analysis were performed with the help of a high-speed camera. It comprised face angle, stroke path, putter speed and impact spot. Results were determined using the limits of agreement technique and delivered encouraging results. A practical evaluation showed that the measurement errors of the system were negligible for a meaningful kinematic putt analysis. As the TOMI system is a commercially available system, no scientific description of the parameter computation and analysis is given. A drawback of the system is its stationarity.

A promising modality for a mobile kinematic system for putt analysis is inertial sensor technology. Early work on the topic and a basic validation of sensor data were reported in [Fitz 06]. Later, King et al. gave an overview of the limitation of ultrasound, infrared and video-based systems and introduced the design, data analysis and performance analysis of a fully integrated measurement system in the club shaft [King 08]. Their system determined velocity, position and orientation parameters during putting. It was validated with a putting robot that included an optical encoder as reference. Error values of 3 mm and 0.5° were determined in golf-specific movements and proved the validity of the system. The biggest drawback of the system is that data are analyzed offline although wireless communication capabilities are integrated. Furthermore, the sensory system cannot be removed from the club.

Building up on the findings of King et al. [King 08], a removable IMU-based system to be mounted on the shaft was developed and described in [Burc 10]. The authors put special emphasis on the golfer's needs for a mobile coaching system. Furthermore, an advanced data analysis technique using a Kalman Filter was described. Drawbacks of the system are the facts that data were only analyzed in static trials and that data analysis cannot be performed in real-time.

Open Research Topics

Different interesting aspects and open questions in the field of golf putting exist. So far, differences of novice and expert players and therefore the results of learning were analyzed. However, it would be of great interest to assess the learning path that athletes undergo. A better understanding of the learning progress could reveal the right training programs according to the current skill level of an individual. Training progress analysis could also help to control training programs according to individual technique variation and learning speed. Therefore, a good understanding of the effectiveness of different training approaches and the effects of training is needed. Furthermore, augmented feedback training applications are needed and their effectiveness has to be investigated. The design of such applications (feedback modality, feedback frequency,...) has to be carefully chosen und assessed.

From a technical point of view, systems for training in the field are of great interest as sports are not performed in a research lab. It might be reasonable to control environment variables for research studies but providing systems for training in the field are certainly needed. Such a system needs to be portable and should address the athlete's needs of automatic and real-time data analysis. Key aspects for such systems are an easy setup and calibration, automatic putt detection as well as instant parameter extraction and display.

We developed such a BSN system for kinematic putt analysis in the field [Jens 15b]. It is introduced in chapter 7 and provides automatic putt detection and kinematic parameter extraction. The same chapter provides the kinematic training progress analysis of a large dataset collected with the aforementioned system.

1.2.5 Instrumented Golf Putter Patents

Several patents cover inventions in the field of golf swing data analysis. In the following overview, we focused on patents that deal with data analysis and instrumented equipment in the field of putting. Patents are listed in chronological order of publication.

- The inventor of [Perl 03] claims sporting equipment that contains a unit that is capable of sensing the motion, velocity, angular orientation and force of the equipment. The sensing unit includes at least one accelerometer and one gyroscope. Data can be transmitted to a command station. This patent also covers instrumented golf putters.

- A patent was filed in [Stor 04] covering a method and arrangement (sensors and computer unit) to detect the motion parameters of a moving object like sporting equipment. The detection is based on an inertial navigation system that measures acceleration, angular velocity and the effect of gravity attraction. This patent also covers instrumented golf putters.

- In [Bosc 06], the inventor describes a golf putting analysis system including an instrumented club and a wirelessly connected instant display device (computer interface). The sensed data are described as acceleration, deceleration, putter path, rotation, lie angle, loft angle, swing tempo, club head distance, ball speed factor and their combinations.
- The authors of [Eyes 06] describe an intelligent golf putting system for improving the putting performance with sensing hardware integrated in the club shaft. Inertial data (3-D accelerometer, 3-D gyroscope) is wirelessly transmitted to a link box to compute and display swing information.
- In [Cho 09], a patent was filed that describes an instrumented golf club. It is capable of receiving input about the distance from ball to hole, sensing the gradient information of the surface and displaying information on an integrated display. With this device, a player can train putting on an imaginary hole.
- The patent in [Stit 10] describes a system for concurrent feedback for golf putting. An instrumented golf club measures a variety of swing parameters and wirelessly transmits them to a portable computer device where parameters are instantly shown against preferred parameter values. The sensing hardware is described as two accelerometers that are mounted on the shaft and the head of the club.
- An image-capturing apparatus mounted on a training putter was invented [Ikka 13]. The camera-based system is capable of determining the correct face perpendicular line and the current deviation from this line. Additional sensors (e.g. gyroscope, accelerometer) are contained in the sensing unit. The deviation information is displayed to the player.

The patent overview revealed many inventions in the field of instrumented sports equipment and specifically golf putters. Some patents are more generic [Perl 03, Stor 04, Eyes 06] while others specifically describe the extracted parameters [Bosc 06, Cho 09, Stit 10, Ikka 13]. In essence, the field is well protected by patents using inertial data analysis.

Three aspects are not fully covered by existing patents. First, the process of putt detection using inertial sensor data is not specifically described in any patent. Second, mainly fully integrated systems are described using specific hardware that is sometimes not removable. Third, the training progress is not described to be supported for longer periods. These findings conform to the literature research in chapter 1.2.4. Our research contributes to the state-of-the-art in providing a BSN system for automatic putt detection and analysis with a removable system that consists of off-the-shelf components. In addition, an analysis of long-term training progress is presented. The golf putt analysis application is presented in chapter 7.

1.2.6 Swimming Kinematics Literature

Size and positioning of BSN components should be optimized to minimize their influence on athlete performance. This is especially the case for the aquatic environment of swimming. This chapter introduces the state-of-the-art in kinematic swimming analysis with BSN systems and motivates an unobtrusive swimming analysis system capable of running on embedded hardware. For this application, the corresponding fundamentals of embedded computing in BSN are described in chapter 3, fundamentals of augmented feedback are described in chapter 4.2.4 and the fundamentals of kinesiology are described in chapter 4.1.

The analysis of swimming motion is a crucial factor for its performance enhancement. Two physical principles contribute to faster and more efficient swimming: the reduction of water resistance and the increase of propelling force [Bach 12]. Video analysis is the common unobtrusive modality to analyze swimming movements in the field [Le S 11], offering descriptive feedback for coaches and athletes alike [Lee 12]. However, drawbacks are a limited capture volume, marker occlusion and bad image quality (lighting, bubbles, water-air intersection, ...), parallax errors and a significant amount of time to process the data [Call 09, Le S 11, Lee 12]. Inertial sensing was proposed as alternative or complementary methodology to measure swimming kinematics [Lee 12]. This section summarizes the literature of IMU-based motion analysis in swimming. Special emphasis is put on the real-time analysis, feedback capabilities and an unobtrusive sensor placement.

First, the focuses of the main research groups in the field are summarized. Second, findings regarding unobtrusive and energy-efficient analyses are described. Third, open research aspects are deduced and presented.

Research Focuses

The following paragraphs provide an overview of the different research focuses in kinematic swimming analysis with inertial sensors. Results are grouped by the leading researcher and presented in ascending order regarding the relevance for this thesis.

Ohgi (Keio University, Japan) and his colleague Ichikawa (University of Tsukuba, Japan) are pioneers in developing logging, analysis and display devices for swimming analysis. Their research focused on the analysis of front crawl and breaststroke swimming with a custom wrist-worn accelerometer device [Ohgi 02b, Ohgi 02c, Ohgi 03]. Data were validated with parallel video recordings and qualitatively described swimming kinematics. Their contribution to the field were the design of appropriate IMU hardware, stroke phase determination and validation as well as investigations on the effects of fatigue. More recent activities are the integration of a gyroscope sensor and the visualization of stroke kinematics [Naka 10].

The group of Fulton, Pyne and Burkett (University of the Sunshine Coast, Australia) investigated kicking movements in swimming with special emphasis on paralympic swimmers. This group focused their research on the basics of leg kicking using data logging and offline analysis. The validity of their gyroscope-based kick-count algorithm was assessed with video analysis. Their algorithms performed better when subjects solely swam with the legs compared to standard freestyle swimming with arms and legs [Fult 09b]. A follow-up using a single sensor on the calf revealed insights in the potential of kicking to optimize swimming performance [Fult 09a]. The authors concluded that the kick-rate is under-utilized in freestyle swimming. Further research investigated net force production to optimize kick training and technique [Fult 11].

Bächlin et al. (ETH Zürich, Switzerland) presented a multi-sensor approach with feedback capabilities called SwimMaster [Bach 12]. Their research focused on counting swim events (start, turn, strokes, ...), analyzing swim technique (body rotation and balance) and the design and integration of a feedback interface. Sensors were placed on the upper and lower back as well as on the wrist during a front crawl swimming test with 18 subjects. The authors contributed to the field with a multi-sensor analysis of swimming motion and a wearable feedback system for swimming. Data were analyzed offline so that the concurrent feedback loop was not closed.

Dadashi et al. at Lausanne university (EPFL, Switzerland) analyzed front crawl and breaststroke swimming motion with respect to stroke phases [Dada 13b], energy expenditure [Dada 14] and instantaneous velocity estimation [Dada 12]. A multi-sensor system was used for estimating the stroke phases (sacrum, wrists) and the energy expenditure (sacrum, wrists, tibia), while a single sensor at the sacrum was used for velocity estimation. Each IMU data logger included a 3-D accelerometer and a 3-D gyroscope. The proposed analyses are based on established mathematical methods such as Kalman Filter, HMMs and Bayesian Linear Models as well as high-quality reference systems, well suited for precise and fine-grain offline investigations.

James et al. (Griffith University Brisbane, Australia) proposed a wearable sensor platform for the aquatic swimming environment and a corresponding swimming analysis. The contribution of this group was suitable hardware and basic algorithms for IMU data analysis of the swimming motion. The first sensor generation was used to detect important events in swimming such as strokes and turns [Jame 04]. In a later study, the authors used a single sensor (3-D accelerometer) on the sacrum to detect these events [Dave 08]. Their validation contained six subjects (40 datasets) and was performed with video recordings, manual event counts and manual timing as reference. The second sensor generation consisted of a 3-D accelerometer and a 3-D gyroscope sensor with wireless communication capabilities. The platform included analysis software and the integration of video [Jame 11]. Further research covered the comparison of 2-D video, 3-D motion analysis and the second generation of IMU data recorded on an on-land swimming bench [Lee 11]. Additionally, the second generation sensors were used to estimate lap velocity profiles using acceleration data [Stam 13].

Several swimming analysis topics were investigated by the group of Le Sage, Slawson, Conway, Justham and West (Loughborough University, UK). The two main contributions of this group are a comprehensive multi-modal (e.g. video, pressure sensor) and multi-athlete swimming analysis system that offers augmented concurrent feedback of IMU data as well as the implementation of signal processing algorithms on a sensor node to reduce transmission bandwidth. Early work focused on acceleration analysis using an IMU worn at the lower back for stroke counting and individual stroke analysis [Slaw 08]. Later, a comprehensive multi-modal system including force plates, video cameras, pressure mats and wireless IMU nodes was set up to monitor the main parts of competitive swimming (start, free swimming, turn) [Le S 12, Chak 13]. This enabled detailed analyses for specific movements such as the tumble turn [Slaw 12]. Furthermore, the system was capable of analyzing data in real-time to provide augmented feedback for coaches during training intervention. An important contribution was the development of live data streaming from the wireless IMUs (3-D accelerometer and 2-D gyroscope) to a base station in the pool environment. The key aspect of this platform was the implementation of signal processing algorithms on the sensor node in order to stream preprocessed data instead of the complete data from sensor node to base station [Le S 11]. The authors implemented low-level algorithms like IIR filtering, zero-crossing analysis and thresholding on the sensor nodes. Some algorithms needed to be adapted according to the swimming styles. Validation was performed using four subjects.

Unobtrusive and Energy-efficient Kinematic Analysis

Different possible IMU sensor positions such as sacrum [Dave 08, Le S 11], wrist [Bach 12] or shank [Fult 09a] have been proposed so far. Authors usually reported that sensors did not disturb the athletes or did not change the movement biomechanics. However, this fact was not proven or investigated in more detail. In general, water resistance should be minimized and comfort of the athlete should be maximized to reduce the influence on the performance as much as possible. Furthermore, as wireless transmission is a challenge in the aquatic environment of a swimming pool, sensing and feedback device are ideally located at the same position to avoid the need of radio data transmission. The head was investigated as an alternative sensor position for swimming movement analysis [Pans 10]. The authors showed a kinematic analysis of 3-D accelerometer data and proved the applicability of the data for lap, breathing stroke and swimming stroke counting. Preliminary results were reported for front crawl, breaststroke and backstroke swimming. More recently, GPS and accelerometer data collected at the head were analyzed to estimate split times and stroke counts in outdoor swimming [Bean 14]. Unfortunately, the acceleration data analysis was not described in detail. The head as feedback position was investigated by Bächlin et al. [Bach 12]. Their system included goggle-mounted LEDs as well as acoustic and haptic feedback devices on the wrist. They concluded that visual and haptic are the most promising feedback modalities. Most recently, a device to measure and feedback heart rate on the swimming goggles was developed [Inst 14]. In summary, these findings suggest that the head is an unobtrusive, comfortable and valuable sensor location for kinematic swimming analysis and feedback display.

For augmented concurrent feedback, data need to be processed in real-time. Thus, efficient algorithms for embedded signal processing need to run on hardware with limited resources. Siirtola et al. investigated the influence of sampling frequency (50 Hz, 25 Hz, 10 Hz, 5 Hz) and sensor position (wrist, back) on the accuracy of swimming event, swimming style and swimming intensity detection [Siir 11]. Linear and quadratic classification methods were used to determine the swimming style and swimming intensity. The authors proved that accurate swimming motion tracking is possible with low sampling rate. However, an estimation of resource consumption or even an embedded implementation of the algorithms was not conducted.

The feedback possibilities in swimming are challenging. Arms and legs are constantly moving, additional equipment and wiring increases water resistance and wireless data transfer is disturbed by the water. Fickenscher et al. tackled these challenges with a new technique of wireless, short range and low rate data transmission based on infrared (IrDA) [Fick 11]. It was specifically developed for the use in an aquatic swimming environment. This technique is a promising approach for connecting components in a BSN for swimming. This technique neglects the need that sensor and feedback coincide but requires low data transfer rates.

Open Research Topics

Research efforts highlighted the need for an augmented concurrent feedback system for swimming. However, neither of the systems featured a closed feedback loop like the Swim Master [Bach 12] or the feedback was only accessible for coaches using displays at the poolside like in [Le S 12]. Other open issues were the suitability of algorithms for wearable IMU hardware, the achievable precision with on-node processing and potential approaches for communication footprint minimization. Real-time analysis on the sensor node is beneficial in different scenarios such as when sensor and feedback location coincide (e.g. [Bach 12, Inst 14]), when data are streamed to a wearable display (using [Fick 11]) or streamed to a base station like in [Le S 11]. An important aspect for the efficiency of on-node processing is the combination of sampling rate and sensor location [Siir 11]. A framework to implement, test and validate different algorithms would be of great interest for the field.

The head is a promising sensor location to minimize water resistance and maximize user comfort. A head-worn sensor positioning was previously investigated and the corresponding analysis achieved encouraging results [Pans 10]. Simple measures such as lap count, split times and swimming style but also more advanced measures like swimming stroke count and breathing stroke count can be deduced from head kinematics. However, precision needs to be further assessed in a large scale study for all swimming and turn styles. Furthermore, the algorithms need to be implemented on the sensor node.

The effect of fatigue is another open research topic in the context of swimming [Ohgi 02a]. Researchers are interested in the kinematic changes through fatigue and coaches would benefit from an instantaneous detection of fatigue signs to fine-tune training programs. Such a system would also be of great interest for amateur athletes to train in the absence of a coach and to maintain the right intensity zone.

We analyzed kinematic swimming data collected at the unobtrusive head position in a scientific study [Jens 16a]. The findings are presented in chapter 8. Special emphasis was put on the accuracy-cost tradeoff of the algorithm to enable an on-node implementation which is beneficial in the aquatic environment of swimming. This applications was implemented under the design considerations presented in chapter 5. We further analyzed the changes in head kinematics with fatigue.

1.2.7 Swimming Monitoring Patents

Several inventions in the field of swimming monitoring are patented. The following chronological overview focuses on wrist- and head-worn devices for swimming data analysis.

- In [Rabi 85], a swimming lap counter based on an earth magnetic field sensor was invented. The patent protects a system that comprises a wrist-worn device and a liquid crystal display mounted on the swimming goggles.

- The patent in [Mysl 88] describes swimming goggles with an integrated display that can be used to display timing information. Special emphasis is put on the design of the display so that data is visible. The analysis is automatically started on water contact.

- In [Taba 97], a swimming goggle with an integrated display and an arm band was invented. The underlying computing hardware comprises an accelerometer to detect laps and calculate movement information.

- Mynhardt filed a patent to protect a turn detector that gives acoustic feedback [Mynh 00]. The technical part comprises a magnetic field sensor worn in the ear and a shock sensor at the ankle.

- The patent described in [Gent 03] protects a system to analyze repetitive movements and provide acoustic and visual feedback. Accelerometer data were sensed to e.g. determine split times and movement or breathing patterns. The system is worn on the head or back and is not limited to swimming.

- A system that is capable of monitoring swimming strokes with a pressure sensor was patented [Leun 06]. The device was designed as watch-type device and contains a breaststroke and non-breaststroke detection mode.

- Another wrist-worn device for lap counting and swimming statistics display is protected by a patent described in [Chan 07]. The analysis is based on data from a "compass" and data are processed on the device to provide augmented feedback.

- A patent protecting a multi-sensor data logger for swimming and other sports was filed [Mack 08]. Data like GPS, acceleration and physiological signals are logged and transmitted to a remote computer to be accessed by coaching staff. A related system additionally sensing angular displacement and direction of movement was also patented [MacI 08].

- A watch device capable of inertial sensing and GPS positioning was protected by a patent described in [Down 09]. Both sensor modalities may be fused, data are processed and displayed on the device and may also be wirelessly transmitted.

- The analysis of accelerometer data with a wrist-worn device that can give feedback was patented [Irla 10]. The goal of the device is the determination of metrics like stroke/lap count and energy expenditure for different sports, especially swimming motions.

- The inventors of [Yuen 12] protect a device to measure and process acceleration and the electrocardiogram of a swimmer for monitoring. Data can be wirelessly transmitted. As swimming parameters, stroke count, turn detection, stop detection and swim style detection are described.

- The analysis of inertial and magnetic sensor data with HMMs to determine the swimming style was protected [Jall 12]. The proposed wearable system is capable of processing data online (e.g. filtering, HMM analysis) and can be worn on the wrist, ankle or head.
- A lap counter for swimming comprising a head-worn and a wrist-worn sensor was protected [Muel 12]. Both sensor modalities have processing capabilities and sense an accelerometer and compass. The wrist device is used for feedback. Information can be transmitted between devices via wireless radio.

The patent overview revealed numerous inventions in the field of data analysis for swimming applications. The common modalities like inertial sensing (accelerometer, gyroscope), pressure sensing, magnetic sensing, satellite positioning (GPS) and physiological sensing (ECG) are covered. Moreover, the obvious positioning on the head, wrist and trunk are covered and concurrent feedback is protected by many devices. In essence, the field is well protected with patents using inertial data analysis.

The idea of body-worn augmented feedback devices for lap counting, style detection in swimming and the likes is protected by multiple patents. However, the actual development and implementation of such a system includes a careful selection of hardware components and signal analysis algorithms to make devices suitable for the aquatic environment. Our contribution to the field is a methodology for sophisticated, energy-efficient data analysis for swimming and other applications. It can be seen as a building block to realize the systems described in the patents above.

Another aspect that is not protected by patents is fatigue analysis. Guiding the athlete to train efficiently is an important aspect in the absence of a coach. Fatigue has to be analyzed in real-time and on the wearable device to support the athlete during exercise. Thus, the design of efficient algorithms is also needed for this analysis task. This work contributes to the field with a sophisticated and energy-efficient data analysis methodology for fatigue detection. The swimming kinematics analysis application is presented in chapter 8.

1.3 Contribution

This thesis contributes to the field of wearable athlete support systems. More specifically, the technical challenges of the accuracy-cost tradeoff in embedded classification system design as well as accurate and energy-efficient data analysis are tackled on the example of three different BSN applications. These applications are plyometric training with the goal of ground contact time measurement, golf putt analysis with the goal of automatic putt detection, parameter extraction and training progress analysis as well as swimming exercise tracking and fatigue detection using head kinematics. The specific contributions are:

1. The determination of algorithm cost is currently either performed with high-level algorithm complexity analysis (Landau-notation) or with low-level measurements (code instrumentation). This thesis introduces the use of the number and type of mathematical operations and parameters for the practical mid-level analysis of embedded classification system cost. A methodology to deduce and use these cost measures to support the solution of the accuracy-cost tradeoff is presented in chapter 5.2. This contribution was published in [Jens 16b]. An application that was designed on the basis of the accuracy-cost tradeoff design considerations is presented in chapter 8.

2. The accuracy of embedded classification systems is a crucial aspect in the design of wearable athlete support systems and this thesis contributes two design considerations. These are the accurate estimation of classification accuracy in experiments with feature selection (chapter 5.3) and the accurate modeling of HMMs for sequential data analysis (chapter 5.5). An application based on the accurate modeling of HMMs is described in chapter 6. An application based on the estimation of classification accuracy in experiments with feature selection is described in chapter 8. Findings on the feature selection topic were published in [Jens 12a].

3. Energy-efficient solutions are crucial to maintain the wearability and runtime constrains of athlete support systems. This thesis contributes design considerations for energy-efficient data analysis and implementation of BSN applications. Chapter 5.4 introduces the underlying techniques and chapter 8 describes a system that implements these techniques. The considerations on energy efficiency were published in [Jens 15a, Jens 16a].

4. Athlete support systems are a valuable additional source of information when important training parameters cannot be precisely assessed with human observations. This thesis contributes a methodology to precisely measure ground contact times in the drop

jump plyometric exercise (chapter 6). A related analysis was published in [Jens 14b]. The ground contact time parameter is crucial for plyometric training control and the presented analysis based on BSN recordings facilitates a wearable athlete support system.

5. The long-term analysis of training progress enables a new view on athletic performance and can support a better understanding of motor learning and individualized training control. Data from field performance and a high number of trials reflect the true performance of an athlete and are more beneficial compared to the potentially limited amount of data collected in a lab environment. Furthermore, augmented feedback can boost the motor learning process. This thesis contributes a system and analysis methodology that realizes these application opportunities on the example of golf putting (chapter 7) that was published in [Jens 15b]. The BSN application featured automatic putt detection and parameter extraction. A high number of 3902 trials of novice golfers during a repetitive training intervention was used to reveal relevant parameters and their progress. Furthermore, the system can act as an important building block for augmented feedback applications as putt parameters are available in real-time. Parts of this work were published in [Jens 11, Jens 14a].

6. A lot of wearable athlete support systems implement data analysis on an embedded system to maintain the system's demands on wearability and information access. This thesis contributes an example for such an application based on the design considerations presented in chapter 5. In the application presented in chapter 8, head kinematics are analyzed for the purpose of exercise tracking and fatigue detection. The exercise tracking functionality was implemented as embedded classification system on the node of a popular BSN platform. The on-node exercise tracking application was published in [Jens 16a]. The presented methodology and toolchain can directly be used for other data analysis questions in the field of wearable computing.

1.4 Structure of this Work

This thesis consists of three major parts. First, existing and relevant fundamentals of the underlying disciplines are presented.

- **Chapter 2** introduces pattern recognition basics using the example of the classification pipeline. In addition, sequential data analysis using HMMs is described.
- **Chapter 3** summarizes the mobile sensing and computing technology that is used in wearable computing systems. Special emphasis is put on the technology that was used throughout this thesis.
- **Chapter 4** describes the fundamentals of kinesiology as well as motor learning and control. Special emphasis is put on augmented feedback.

Second, the design considerations to address the technical challenges of embedded classification systems are presented.

- **Chapter 5** describes design considerations for embedded classification system. Thereby, a methodology to support the solution of the accuracy-cost tradeoff in embedded classification system design is presented. Special emphasis is put on the accurate estimation of the classification accuracy when using feature selection and on the accurate modeling of HMMs for sequential data analysis. This chapter also contains methods for the energy-efficient design and implementation of BSN data analysis algorithms.

Third, application examples showing the opportunities of embedded classification systems in the field of sports are presented. Thereby, the above-mentioned technical challenges are considered.

- **Chapter 6** introduces an application to precisely measure the ground contact time in drop jumps. Therefore, a HMM analysis is used.
- **Chapter 7** describes a mobile golf putt analysis system that featured automatic putt detection and parameter extraction in real-time. As an example of long-term field data analysis, data from a motor learning study was analyzed for the purpose of training progress analysis.
- **Chapter 8** introduces an application for swimming exercise tracking and fatigue detection based on head kinematics. The embedded classification for exercise tracking was implemented on a BSN sensor node and evaluated regarding energy-efficiency.

The final **chapter 9** summarizes, discusses and concludes the results in the context of the complete thesis. Possible future research is summarized in the outlook.

Chapter 2

Pattern Recognition Fundamentals

This chapter introduces the fundamentals of pattern recognition, a methodology for sophisticated data analysis. It defines the general terminology and nomenclature with a focus on pattern classification. The content of this chapter provides the basis for the design considerations described in chapter 5.

As human beings, we sense the environment with various senses. For example, we use our eyes and ears to recognize faces, understand spoken words or read handwritten letters [Duda 01]. Our brain performs complex processes to learn and execute these tasks.

Complex tasks of recognition, classification and analysis are also important problems in engineering and scientific disciplines like biology, medicine, artificial intelligence, and remote sensing [Jain 00]. Therefore, pattern recognition methods model and implement these processes for automatic signal processing. Teaching a technical system to perform these kinds of tasks is also called machine learning. This section illustrates the basics of a technical pattern recognition system. First, we introduce pattern recognition terms. Second, the general structure and nomenclature is illustrated using the classification pipeline [Niem 90]. Third, the sequential pattern recognition and analysis technique of HMMs is introduced [Rabi 89].

The nomenclature throughout this thesis uses *italic font* for 1-D variables and **bold font** for vectors and matrices. Units are shown in normal font.

2.1 Introduction and Terms

Pattern recognition provides a methodology that describes how machines can observe, learn to distinguish and categorize patterns of interest [Jain 00]. Traditional pattern modalities range from images over speech signals to handwritten text. The term pattern describes an instance of interest that is further processed. Pattern recognition can be further subdivided into different tasks:

- Data mining [Witt 11]: The automatic or semiautomatic process of discovering patterns in data. These patterns have to be meaningful in a sense that they lead to some advantage which is usually an economic one. The interpretation of the pattern is of main interest.
- Pattern analysis [Niem 90]: This is the in-depth analysis to deduce interesting events within the pattern or the derivation of a description of the pattern in an adequate level of abstraction. The content and subparts of the pattern are of main interest.
- Pattern classification [Niem 90]: This term denotes the categorization of a pattern to either predefined classes (supervised classification) or data clusters and groups (unsupervised classification). The category of the pattern is of main interest.

This work focuses on classification, the primary goal of pattern recognition [Jain 00] and will specifically target supervised classification. The structure of a supervised classification system can be described as classification pipeline [Niem 90].

2.2 Classification Pipeline

The classification pipeline describes the main steps of a classification system and will be introduced in this section. Each technical step will be supported by an example description of human perception as descriptive illustration. This human perception model is simplified to support the understanding of the technical steps.

The processing steps of a technical pattern recognition system are illustrated in the classification pipeline (Fig. 2.1) which will be subsequently described in more detail. The general structure, introduced in [Niem 90], was adopted to the processing steps presented in this thesis. From a high-level perspective, the pipeline describes the data processing to transform sensed input into a class decision output.

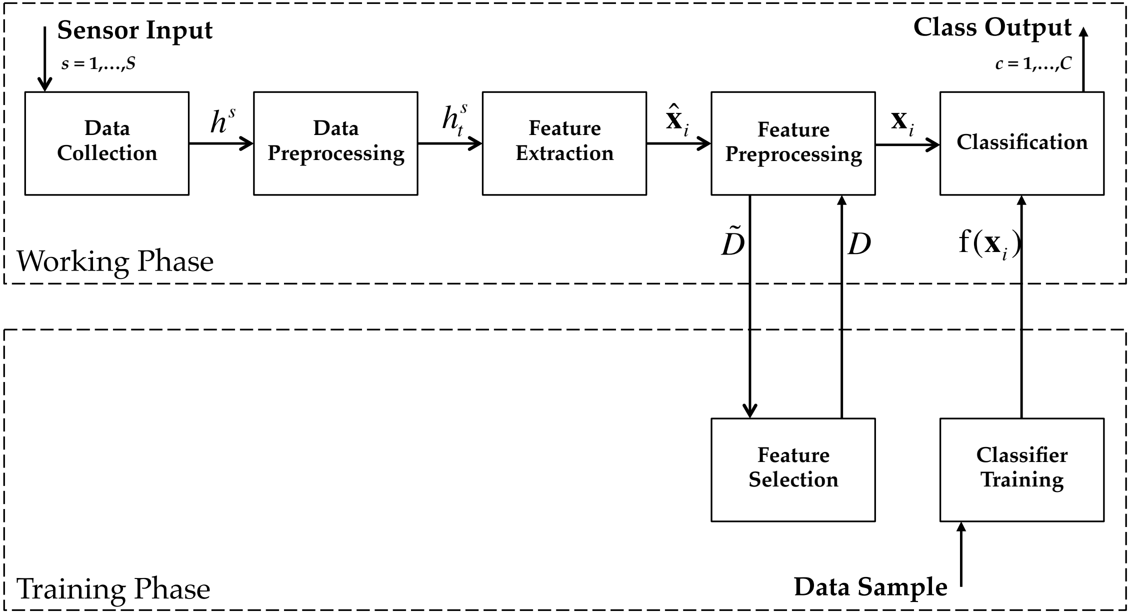

Figure 2.1: Classification pipeline consisting of working and training phase (adopted from [Niem 90]). In this thesis, the variable S denotes the sensors that collect digital 1-D signals h^s. Segmented data intervals are denoted by h_t^s and feature vectors with either $\mathbf{x}_i$ (not preprocessed) or $\mathbf{x}_i$ (preprocessed). The dimension of the feature space is denoted by $\tilde{D}$ (complete feature set) and D (selected feature set) respectively. The trained classifier is denoted by $\mathrm{f}(\mathbf{x}_i)$ and C denotes the predefined classes.

The sensor input and class output of the working phase of the classification pipeline (Fig. 2.1) are the conceptional interfaces to access a trained classification system. The goal of the system is to transform a given input to an adequate output which is a class decision. An example of these interfaces can be described with the human perception analogy. Amongst other senses, we use our eyes to recognize a face and process the perceived information to remember the name of the person we face. The sensor input of a technical system represents the analog entity to the eyes in human perception and the class output is the name of the person.

The different phases of the classification pipeline, the working phase and the training phase (Fig. 2.1) describe how a classification system is created and how it takes the class decision. The training phase enables the classification system to perform decisions. This step is also called model creation or classifier training. The trained classifier is e.g. a statistical model, a mathematical function or a set of decision rules. The trained classifier enables decision making in the working phase. The working phases describes the decision making steps of the classification system once it is trained. Working phase and training phase also have a counterpart in human perception. The training phase corresponds to the phase when meeting a previously unknown person. The unknown person is introduced and the name is linked with the person's face that is sensed with the eyes. The working phase is processed when meeting the same person again at a later point in time. The visual expression of the person's face is processed to determine the name. This processing reveals the correct name (right class) or the incorrect name (wrong class) during working phase.

The following detailed description starts with the working phase steps that incorporated data collection, data preprocessing, feature extraction, feature preprocessing and classification. The training phase consisted of feature selection and classifier training and is described afterwards. Special emphasis is put on the phases of feature extraction, feature selection and classification.

2.2.1 Data Collection

The first step of the working phase describes the input data for further processing. There are numerous types of different sensors like cameras, microphones or kinematic sensors. In this thesis, data input of S sensors was considered as simultaneous input. We assumed that sensor data were synchronized and different sensors are denoted by $s = 1, \ldots, S$. When dealing with continuous analog signals, data collection samples and quantifies this input for further digital processing. In this thesis, the effort for signal digitalization was not considered. Thus, each sensor dimension s delivered a digital 1-D value signal h^s which was discrete in time and value.

In the human perception analogy, the face recognition system consists of $S = 2$ sensors, the eyes. Data from both eyes are processed simultaneously. In contrast to a digital system, humans sense a continuous picture of the environment.

2.2.2 Data Preprocessing

Data preprocessing can be used to make the input more suitable for further processing [Niem 90]. Examples are filtering and segmentation. This work considered segmentation while other data preprocessing techniques are beyond the scope of this thesis. The input signals were segmented according to a constant length T. The segmentation step resulted in a pattern that can be used for further processing. Each pattern comprised the values h_t^s of each sensor dimension s containing the samples t with $t = 1, \dots, T$. The result of the data preprocessing phase described in this thesis was a S-dimensional pattern of length T. This step of the classification pipeline processes a pattern consisting of sensor data. Therefore, the pattern is represented in data space. A transformation to another space will be described in the next section.

A data preprocessing step in human perception using information from the eyes would be to focus the view on the person's face. In analogy to the described interval length of a technical classification system, human's vision needs sufficient time to be able to recognize a person. Recognizing a person when catching a glimpse in a crowded railway station is complicated. It is easier to recognize a face when watching a person unhurriedly.

2.2.3 Feature Extraction

Feature extraction determines characteristic properties of a pattern. It is therefore a transformation from data space where the pattern comprised sensor values, to feature space where the pattern comprises feature values. A pattern in feature space is often multi-dimensional and therefore also called feature vector. It is a characteristic representation of the pattern in data space. This thesis describes feature extraction as follows: a pattern in data space consisting of h_t^s with $t = 1, \dots, T$ is transformed to a feature vector $\mathbf{x}_i$. The index i with $i = 1, \ldots, I$ describes subsequent patterns. Individual features of a single feature vector are denoted by $\hat{x}_i^d$ with $d = 1, \ldots, \tilde{D}$ denoting the dimension of feature space and the index i with $i = 1, \ldots, I$ describing subsequent patterns. The result of the feature extraction step is a pattern in feature space.

There are different methods for feature generation. They can be categorized in heuristic methods and analytic methods like in [Niem 90]. Heuristic features are motivated by intuition and experience [Niem 90]. The generation of analytic features is the systematic approach of deducing optimal features [Niem 90]. Specific feature sets for different classification tasks evolved. Examples are mel-frequency cepstral coefficients (MFCC) for speech and audio classification [Theo 09] and Haar-like features for face detection [Viol 01]. Features for the classification task described in this thesis, biosignal classification, were proposed in [Ciac 93]. This thesis focuses on time-domain features representing signal characteristics as well as statistical moments and was motivated in [Jens 12b]. The feature extraction methods investigated in this thesis are described in more detail in chapter 5.2.2.

Human visual face recognition uses different characteristics to recognize persons. These might be face shape, nose size, eye color, distance between eyes or others. As in different classification tasks, a different feature set is used when humans try to recognize whales. There, the shape, size and color of the dorsal fin might be the most discriminant features.

2.2.4 Feature Preprocessing

Feature preprocessing is another step to create more suitable patterns for further processing. Compared to data preprocessing, this step is performed in feature space. This thesis considered the feature preprocessing techniques of outlier removal and data normalization. Outlier removal is a technique to remove patterns that are very far from the mean value of the patterns [Theo 09]. Data normalization addresses the fact that values of different features lie in different dynamic ranges [Theo 09]. Outliers and different feature value ranges both influence classifier training in an undesired way [Theo 09]. In this thesis, each feature vector $\mathbf{x}_i$ was either transformed to $\mathbf{x}_i$ (data normalization) or removed (outlier removal).

Techniques for data normalization are linear techniques like normalization for zero mean and unit variance or linear scaling in predefined ranges [Theo 09]. Non-linear techniques are suitable if data are not evenly distributed around the mean. These techniques are described in [Theo 09].

This thesis focused on linear scaling in predefined ranges which is described in chapter 5.2.3.

An overview of techniques for outlier removal is given in [Barn 87]. A specific algorithm that gained popularity in the pattern recognition community is RANSAC [Fisc 81]. It describes an iterative method to select the correct model parameters of a series of measurements in the presence of outliers. One application of the RANSAC algorithm was described as image processing. This thesis, however, focuses on outlier removal based on quartiles.

Humans are also capable of recognizing a face when the person is far away or if the face is not homogeneously illuminated. These variations can be seen as different feature value ranges and the human visual system can obviously cope with these variations.

2.2.5 Classification

The final step of the working phase is classification. In this step, the classification decision is taken. In this thesis, a classifier is represented as a function $\mathrm{f}(\mathbf{x}_i)$ that takes the feature vector $\mathbf{x}_i$ as input argument. The result of the function evaluation is a class c with $c = 1, \ldots, C$ with C being the set of predefined classes. The function $\mathrm{f}(\mathbf{x}_i)$ comprises all necessary steps and computations for taking the final classification decision.

The analog step in visual face detection is when a human decides for the name of the person he or she faces. The number of classes in human perception is the number of persons one can differentiate.

Different paradigms for supervised classification can be categorized in linear and non-linear classifiers as well as classification based on the Bayes decision theory as described in [Theo 09]. The following paragraphs give an overview for each category. Classifiers can be part of one or more categories or might have variations belonging to different categories. A clean categorization and detailed description of classifiers is beyond the scope of this thesis.

Classification Based on the Bayes Decision Theory

Bayes decision theory classification is based on the idea to regard features as random variables and decide for the class which is most probable. In the following section, probabilities are denoted by P and probability density functions (pdf) by p. In this classification paradigm, the class with the highest posterior probability $P(c|\mathbf{x_i})$ among all classes $c = 1, \ldots, C$ is selected [Theo 09].

According to the Bayes rule

$$P(c|\mathbf{x}_i) = \frac{P(c)\,p(\mathbf{x}_i|c)}{p(\mathbf{x}_i)} \tag{2.1}$$

the posterior probability can be computed with the prior class probability $P(c)$ and the class-conditional pdf $p(\mathbf{x}_i|c)$. The pdf of the patterns, $p(\mathbf{x}_i)$, can be neglected in subsequent considerations as it is the same for all classes [Theo 09].

This category of classifiers can be subdivided into parametric classifiers in which the distribution type of the class-conditional pdf is known or assumed and non-parametric classifiers in which this pdf is unknown [Jain 00]. An example for a parametric classifier is the naive bayes classifier which will be explained in more detail in chapter 5.2.4. An example for a non-parametric classifier is the nearest neighbor classifier which is explained in the same chapter.

Linear Classifiers

Linear classifiers construct a linear decision boundary to separate classes. The decision boundary is the border between two classes that is used to take the class decision. Different ideas to optimize the position of this boundary exist for linear classification. One idea is to maximize the margin around this boundary as optimization criterion (support vector machine (SVM), [Theo 09]). Another idea is to select the boundary so that feature vectors of the same class are close together while feature vectors of different classes are highly separated (fisher linear discriminant, [Duda 01]). An example for a linear classifier is classification via logistic regression introduced in chapter 5.2.4.

Nonlinear Classifiers

Different techniques to construct a nonlinear decision boundary exist. One popular example are decision trees [Theo 09]. The general concept of these classifiers is to learn a decision tree where inner nodes determine the path through the tree based on feature values and tree leafs contain the class decision. The features of an unknown pattern are then used to travers the tree to one of the nodes representing the class decision. An example for a nonlinear classifier based on decision trees is PART which will be described in chapter 5.2.4.

Another example for a nonlinear classifier is AdaBoost [Poli 07]. This algorithm constructs a single final classifier using a set of partial classifiers.

These partial classifiers are iteratively selected based on misclassification in the previous iteration step. The classification of an unknown pattern is performed with evaluating all partial classifiers to take the final class decision. The AdaBoost classifier is described in more detail in chapter 5.2.4.

2.2.6 Feature Selection

Feature selection is performed for different reasons like reaching better classification accuracy, improving classifier generalization or reducing the computational complexity in classification [Theo 09]. The general idea of feature selection is to remove features that do not influence or even decrease the classification accuracy. In the nomenclature of this thesis, the data dimension $\tilde{D}$ is reduced to dimension D in the feature selection step.

Human vision might focus on specific parts of the face to recognize another person. These might be parts that make face differentiation easy or enable a fast person recognition.

Techniques for feature selection can be categorized in filter, wrapper and embedded approaches [Guyo 03]. Filter approaches are classifier independent. They provide a generic selection of features as a preprocessing step. The resulting feature set can be used together with different classifiers. Filters use statistical characteristics like correlation or mutual information as described in [Guyo 03] or entropy as described in [Witt 11] to select a discriminant feature set for classification. An example for a filter approach for feature selection is branch-and-bound search that is described in chapter 5.2.6.

Wrapper approaches are classifier dependent. Features are selected according to their predictive power in combination with a specific classifier. A selected feature set for one classifier might not achieve maximum performance when used with another classifier. The general challenges of the wrapper approach are how to search the feature space and how to evaluate and stop the selection process [Guyo 03]. An example for a wrapper approach for feature selection is best first search that is described in chapter 5.2.6.

Embedded approaches are also classifier dependent and integrated in the classifier training process. This can be achieved e.g. with an integration in the objective function that is optimized during classifier training [Guyo 03]. This integration might have the advantage of being more efficient as available data is not split in training and validation set and the classifier is not trained from scratch on every feature subset evaluation. See [Guyo 03] for examples of embedded approaches for feature selection.

2.2.7 Classifier Training

Before a technical system can differentiate between classes, it learns the class differences in a training process. The learning approaches highly depend on the classifier and its underlying paradigm for group separation. Most classifiers have in common that a suitable data sample for computing a robust, accurate classifier is needed [Niem 90]. A classifier can only differentiate what was learned in the training phase and the model that was learned during training is used in the working phase. An exception to this statement is the nearest neighbor classifier that does not contain a training phase. See chapter 5.2.4 for details of the nearest neighbor classifier.

The human visual system also needs training to detect faces in a robust and repeatable way. This might be an iterative process such that we are able to detect faces more reliably when we see a person multiple times.

The process of classifier training will be illustrated using the example of the SVM which will be investigated in more detail in chapter 5.2.4. The idea of this classifier is to find a decision boundary so that the margin between classes is maximized (Fig. 2.2). This maximum margin decision boundary is designed to enhance the generalization of the classifier. Generalization means that the trained classifier is able to discriminate unknown data with high accuracy [Theo 09].

This visual idea is transferred to a mathematical optimization problem to maximize the margin around the decision boundary. It is a nonlinear optimization problem which is subject to linear inequality constraints. See [Theo 09] for a detailed mathematical description that is beyond the scope of this thesis.

The name of this classifier is inferred from the training feature vectors on the margins that are closest to the decision boundary. They are called support vectors and represent the critical elements of the training sample to define the decision boundary [Theo 09]. The presented concept of the SVM can be enhanced to handle non separable training samples, separation of more than two classes and a non-linear decision boundary using the kernel trick. These enhancements are shown in [Theo 09, Vapn 95, Scho 01]. The working phase of this classifier is described in more detail in chapter 5.2.4.

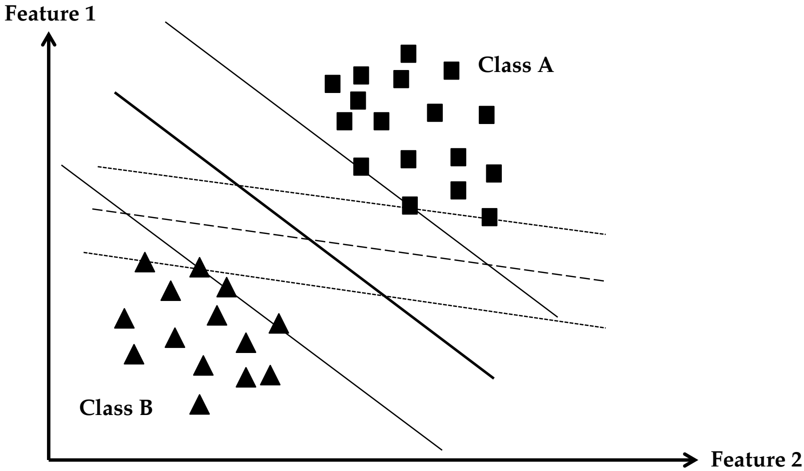

Figure 2.2: Idea of margin maximization in support vector machine (SVM) classifier training. The figure shows the linear case of a separable two class problem in two-dimensional space. A suboptimal decision boundary and the corresponding margin are shown with dashed lines. The maximum margin decision boundary and corresponding margins is shown with solid lines.

2.3 Sequential Data Analysis Using a Hidden Markov Model (HMM)

The classification pipeline (Fig. 2.1) illustrated the different steps to classify a single self-contained pattern. However, this knowledge might not be sufficient when analyzing sequential data. In sequential data analysis, context information is used to support the pattern classification or pattern analysis task. Human speech analysis is such an example where context information like grammatical rules or language models can support the pattern analysis. Different techniques to analyze sequential data exist [Diet 02]. This section introduces one of the most popular ones, the concept of a hidden markov model (HMM) [Rabi 89]. This machine learning technique can be used for either pattern classification or pattern analysis tasks.

Traditionally, HMMs were mainly applied to speech processing applications. This fact is underlined by selected applications for speech recognition in a groundwork on HMM [Rabi 89]. More recently, Dadashi et al. used HMMs for the analysis of kinematic swimming data [Dada 13a].

In their research, the repetitive movement of breaststroke swimming was analyzed with regards to kinematic phase lengths.

Generally, HMMs combine the information of the sensed signal (observation) with the sequence information (model state). Thereby, the transition to the next state is a stochastic process and depends on the current observation and the current state. This dependency is called markov property. Furthermore, the observation is a probabilistic function of the model state. Thus, two underlying stochastic processes are considered: the relation of an observation to its corresponding model state (observation distribution) and the relation of two subsequent model states (state transitions distribution) as illustrated in Fig. 2.3. As the underlying process of mapping observations to models states is not directly observable, it is called hidden.

A HMM is defined as $\lambda = (\mathbf{A}, \mathbf{B}, \Pi)$ with $\mathbf{A}$ being the state transition distribution matrix and $\mathbf{B}$ being the observation distribution information. The probability of each state being the initial state of the sequence is encoded in the vector Π. The model training determines the parameters of λ according to the labeled training sequences or prior knowledge. If both are not available, a naive labeling called linear alignment can be

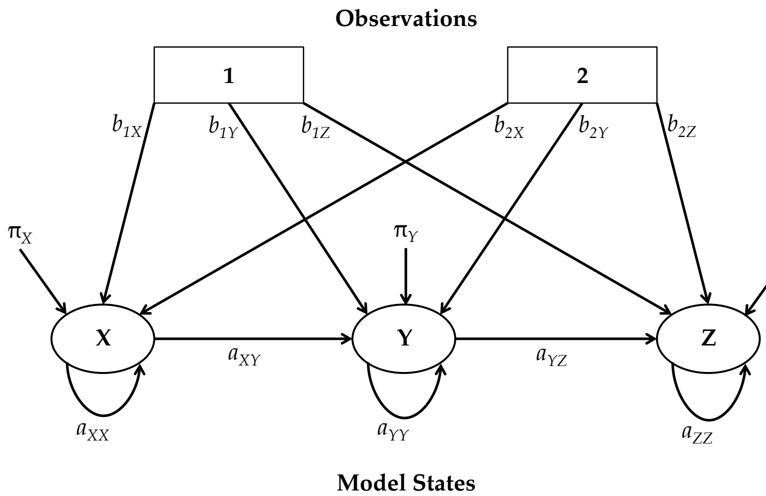

Figure 2.3: Schematic overview of a HMM $\lambda = (\mathbf{A}, \mathbf{B}, \Pi)$ with two observations (1, 2) and three model states (X, Y, Z). Annotation provides elements of state transition matrix $\mathbf{A}$ (a_{ij}), observation matrix $\mathbf{B}$ (b_{ij}) and initial state matrix Π (π_i).

used. Linear alignment creates a labeled sequence in which all states of the sequence have the same length. States are then refined according to the data characteristic. With linear alignment, training data (instead of the labeling) defines the statistical characteristics of each state.

In the working phase, observation sequences are analyzed for their most probable state sequence using the trained model. In the case of a pattern classification task, one model for each class is trained and the pattern is assigned to the class that delivers the higher probability. In the case of a pattern analysis task, the detailed structure of the pattern can be retrieved from the most probable state sequence which is the outcome of the HMM analysis.

Chapter 3

Mobile Sensing and Computing Fundamentals

This chapter gives an introduction to the paradigms and components of mobile sensing and computing systems. It is organized in a bottom up approach. First, the sensor technology behind microelectromechanical systems (MEMS) as described in [Jone 13] is introduced. Special emphasis is put on the underlying electromechanical effects, the sensor types, sensor noise and drift as well as challenges in biomechanical acceleration measurement. Second, the concept and the components of an inertial measurement unit (IMU) that consists of MEMSs is introduced. Third, the term body sensor network (BSN) is introduced. BSNs are an application area of IMUs and its components, sensor nodes and processing nodes, are described with off-the-shelf examples. Fourth, the terms embedded system and cyber-physical system (CPS) are introduced as this technology is an alternative to off-the-shelf BSN hardware. Special emphasis is put on hardware and design considerations of such systems. The section closes with a comparison between standard hardware components and custom design systems. The content of this chapter provides the technical basis for the embedded classification applications that are introduced in chapters 6 to 8.

3.1 Microelectromechanical System (MEMS)

Electromechanical transducers are devices that link electrical or magnetic forces and mechanical motion [Wils 05]. These devices work as a sensor when transforming mechanical motion into an electrical or a magnetic signal. They work as an actuator when electrical or magnetic signals are transformed into mechanical motion. The physical quantities that are measured with MEMS can be subdivided into electrical (e.g. charge, voltage, capacitance) and magnetic (e.g. magnetic flux, induced voltage, inductance) measurement. This thesis focusses on the sensing functionality of MEMS that is based on different electromechanical effects. First, these effects are described. Different sensors exist to sense various kinds of mechanical motions. Second, these sensor types are described.

3.1.1 Electromechanical Effects

Different effects like the piezoelectric, piezoresistive and the capacitive effect can be harnessed in MEMS for sensing mechanical motion [Jone 13]. The piezoelectric effect is described according to [Jone 13], the piezoresistive and the capacitive effect according to [Kraf 13].

Piezoelectric Effect

MEMS transducers can use materials that show the piezoelectric effect. This piezoelectric material can transform mechanical motion into an electrical signal which is the purpose of a MEMS sensor. The working principle is a shift of charges in the lattice structure of the material when the material undergoes strain. This shifting induces bound electric surface charge and volume-distributed dielectric polarization and occurs on tensile, compressive, or shear mechanical strain. The effect can be harnessed for MEMS sensing with crystalline, ceramic, polymer and natural materials that show the piezoelectric effect.

Piezoresistive Effect

Another possibility to sense mechanical motion is to harness the piezoresistive effect. This effect is based on a change in the resistance of piezoresistive material when its geometry is changed on mechanical stress. The working principle is a modification in the charge carrier mobility of the material. When strain is applied to the material, mean free time of carrier collisions and the carrier effective mass are changed which causes a change in the resistance. As piezoresistive material, silicon can be used.

Capacitive Effect

Mechanical motion can also be sensed using the capacitive effect. This effect exploits a change in the capacitance if the proximity of the capacitor plates changes. In such a sensing element, one capacitor plate is able to move in one direction, increasing or decreasing the distance to the second, stationary capacitor plate. Thus, the movement of the moving plate reflects the mechanical motion and can be measured as a change in capacitance.

3.1.2 Sensor Types

Different types of mechanical motion can be be sensed with MEMS. This work focusses on two inertial measurement sensor types, accelerometers and gyroscopes, that are used throughout the thesis. Recently, magnetometers supplemented inertial sensing. Their measuring principle is also briefly described.

Accelerometer

An accelerometer measures linear acceleration and its measuring principle can be described as a spring-mass-damper system (Fig. 3.1). If the housing is linearly accelerated, the inertia of the mass causes the mass to move relative to the device housing. The mass returns to its initial position when the acceleration of the housing stops [Kraf 13]. The acceleration can be calculated using Newton's law of motion

$$F = m \times a \tag{3.1}$$

where F is the force measured with the system, m is the mass of the mass element and a is the applied acceleration [Wils 05].

The dynamics of the system can be expressed with the second-order differential equation

$$m\frac{d^2 y}{dt^2} = m\frac{d^2(x-y)}{dt^2} + b\frac{d(x-y)}{dt} + k(x-y) \tag{3.2}$$

where m is the mass, $x-y$ is the displacement of the mass relative to the housing, t is the time, b is the damper constant and k is the spring stiffness [Wils 05]. All terms that are added on the right hand size are forces with unit N. The force F that is needed to calculate the applied acceleration a in Eq. 3.1 can e.g. be directly measured with the piezoelectric effect [Kraf 13]. When using the capacitive effect, F can be calculated by measuring the displacement $(x-y)$ and inserting the result into the right hand side of Eq. 3.2 [Kraf 13].

Acceleration is expressed in the unit m/s^2 (also: g) and generally comprises static acceleration and dynamic acceleration. Static acceleration

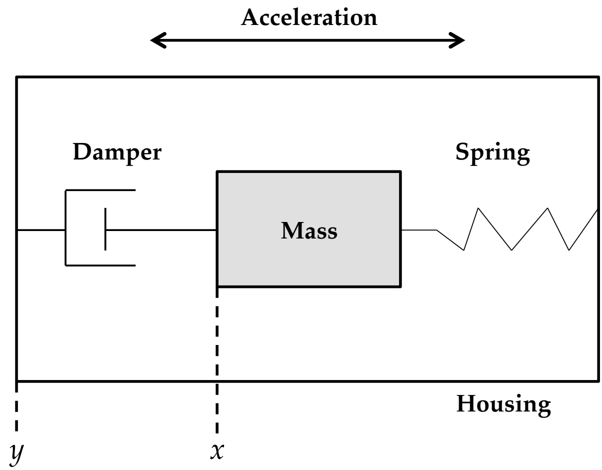

Figure 3.1: Measuring principle of an accelerometer in form of a spring-mass-damper system.

exists due to the earth's gravitational force. Dynamic acceleration occurs when sensors are physically accelerated. Three orthogonal axes are measured in state-of-the-art hardware (Tab. 3.1). The static acceleration is thereby often output in motionless state. This results in a measurement of 1 g due to earth gravity. The value of 1 g is measured on one axis when this axis points in opposite direction to the earth gravity vector and the sum of all acceleration values adds up to 1 g if the accelerometer axis does not point in the opposite direction. Using acceleration, speed and subsequently displacement can be computed with integration.

Gyroscope

A gyroscope measures angular velocity and its measuring principle can be described as a 2-D spring-mass-damper system (Fig. 3.2). Two axis, the drive axis and the sense axis, are used to determine the angular velocity of the housing rotation. To enable measurement, the mass has to oscillate at resonance and in direction of the drive axis. This oscillation is the prerequisite to be able to measure the coriolis acceleration on the sense axis when the housing is rotated [Jone 13]. This type of gyroscope is called rate vibratory gyroscope, coriolis vibratory gyroscope or vibrating structure gyroscope.

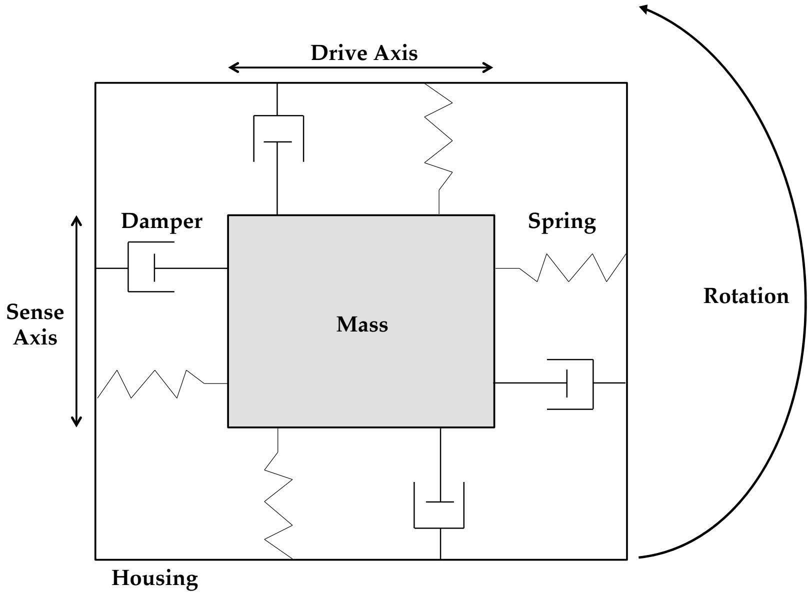

Figure 3.2: Measuring principle of a gyroscope in form of a 2-D spring-mass-damper system.

On rotation, three forces affect the movement of the mass. These are the euler force, the centrifugal force and the coriolis force. If the angular velocity is much smaller than the oscillation in drive axis direction, the euler force and the centrifugal force can be neglected [Jone 13]. The coriolis force is then the only relevant force that affects the mass movement in sense axis direction. The coriolis acceleration in sense axis direction can e.g. be measured as presented in the previous paragraph.

Two prerequisites have to be fulfilled so that the coriolis acceleration on the sense axis is directly proportional to the angular velocity of the housing on rotation [Wils 05]. First, the mass has to oscillate in drive axis direction. Second, the angular velocity of the housing rotation has to be much smaller than the drive axis oscillation. See [Wils 05] for details.

Angular velocity (also: turning rate, rotational velocity) is expressed in the unit °/s. Three orthogonal axes are measured in state-of-the-art hardware (Tab. 3.1). Using angular velocity, rotation angles can be computed with integration.

Magnetometer

A magnetometer measures magnetic fields, e.g. the vector components of the earth magnetic field. Therefore, it can be used as an electronic compass. Most of the MEMS magnetometers use the Lorentz force measuring principle [Lenz 06] which is described in the following. A conductor, carrying an electric current, experiences a force in the presence of a magnetic field. The force is called Lorentz force and causes mechanical motion (e.g. deformation, [Bero 03]) of the conductor structure. This motion can be measured with the piezoresistive effect [Bero 03].

The strength of magnetic fields is expressed in the units T (tesla) or Gs (gauss). Three vector components defining the direction of the magnetic field are measured in state-of-the-art hardware (Tab. 3.1). This information can be used to determine the orientation of the sensor with respect to the earth magnetic field.

3.1.3 Noise and Sensor Drift

The sensor characteristics like transfer function, span (valid input data range) or nonlinearity determine the behavior and precision of a sensor system. Environmental factors like aging, temperature, humidity or electromagnetic interference may affect the measurement itself but also the short and long-term stability of the sensor. The sensor signal stability is also called drift. Short-term stability describes changes of the sensor output within minutes, hours or days and can be described as ultralow-frequency noise [Frad 04].

Noise is present in all electronic sensor circuits and can be categorized in inherent noise and interference noise. Inherent noise arises in the circuit while interference noise is picked up from outside the circuit. Distortions of the sensor signal can further be categorized in systematic and stochastic. The factors causing systematic distortions (e.g. manufacturing tolerances, material quality) change slowly over time. Stochastic distortions change quickly and are often irregular and unpredictable [Frad 04].

Different sources of noise affect the measurement of MEMS sensors. One inherent noise source are the moving charges as current flow at an atomic level is very erratic. The noise level depends on the temperature since the movement of charges is also temperature dependent. Interference noise is e.g. caused by power lines and ambient temperature changes. Mechanical noise of moving sensors can cause resonant effects that alter the measurement [Frad 04]. A more detailed consideration of noise in MEMS is beyond the scope of this thesis and can be found in [Frad 04].

Manufacturer	Product	Module size [mm]	Sensor	Axes	Sensor range (maximum)	Reference
State-of-the-art						
Freescale Semiconductor Inc.	FXLS8471Q	3.0 × 3.0 × 1.0	Accelerometer	3-D	± 8 g	[Free 15b]
	FXAS21002C	4.0 × 4.0 × 1.0	Gyroscope	3-D	± 2000 °/s	[Free 15a]
	FXOS8700CQ	3.0 × 3.0 × 1.2	Accelerometer	3-D	± 8 g	[Free 15c]
			Magnetometer	3-D	± 1200 mT	
InvenSense Inc.	ITG-3701	3.0 × 3.0 × 0.75	Gyroscope	3-D	± 2000 °/s	[Inve 15c]
	MPU-6555	3.0 × 3.0 × 0.9	Accelerometer	3-D	± 16 g	[Inve 15d]
			Gyroscope	3-D	± 2000 °/s	
	MPU-9255	3.0 × 3.0 × 1.0	Accelerometer	3-D	± 16 g	[Inve 15e]
			Gyroscope	3-D	± 2000 °/s	
			Magnetometer	3-D	± 4800 mT	
STMicroelectronics N.V.	LIS2DH	2.0 × 2.0 × 1.0	Accelerometer	3-D	± 16 g	[STMi 15b]
	L3GD20H	3.0 × 3.0 × 1.0	Gyroscope	3-D	± 2000 °/s	[STMi 15a]
	LSM6DS1	3.0 × 3.0 × 0.86	Accelerometer	3-D	± 8 g	[STMi 15c]
			Gyroscope	3-D	± 2000 °/s	
	LSM9DS1	3.5 × 3.0 × 1.0	Accelerometer	3-D	± 16 g	[STMi 15d]
			Gyroscope	3-D	± 2000 °/s	
			Magnetometer	3-D	± 1600 mT	
Used in this thesis						
Freescale Semiconductor Inc.	MMA7361L	3.0 × 5.0 × 1.0	Accelerometer	3-D	± 6 g	[Free 15d]
InvenSense Inc.	IDG-500	4.0 × 5.0 × 1.2	Gyroscope	2-D	± 500 °/s	[Inve 15a]
	ISZ-500	4.0 × 5.0 × 1.2	Gyroscope	1-D	± 500 °/s	[Inve 15b]

Table 3.1: Overview of state-of-the-art in inertial measurement unit modules and sensors used in this thesis.

The measurement in MEMS sensors is affected by noise and the signal might not be stable in a short-term scale of minutes and hours. Noise and signal instability manifests in sensor drift when inertial signals are integrated over time. To calculate e.g. linear replacement from linear acceleration, the signal has to be integrated twice. The above-mentioned errors accumulate and corrupt the linear displacement value. This corruption is present for all integration operations. A correction of MEMS sensor noise is beyond the scope of this thesis. MEMSs can be used for biomechanical acceleration measurements. However, different challenges have to be considered.

3.1.4 Challenges in Biomechanical Acceleration Measurement

In biomechanics, MEMS acceleration measurements can be used to determine the ground reaction forces on the human body [Nigg 07]. Therefore, the acceleration of selected body segments is measured. As body segments consist of rigid and soft tissue, two major questions were raised in [Nigg 07]:

1. Which acceleration is measured? The acceleration of the rigid tissue part, the soft tissue part or a combination of both?
2. What is the error between measured and actual acceleration?

In general, the measured acceleration depends on bone acceleration, mounting, motion, gravity and cross-sensitivity. Light et al. conducted a study were acceleration was measured with a sensor screwed onto the bone [Ligh 80] which minimized the influence of the soft tissue. However, the sensor is usually mounted on the body surface with no direct contact to the bone. The main error when measuring segment acceleration is introduced with mounting sensors on soft tissue [Nigg 07]. Thus, this aspect will be discussed exemplarily. The remaining factors of motion, gravity and cross-sensitivity are described in [Nigg 07] in more detail.

To estimate the influence of mounting, an experiment to quantify the difference of bone and soft tissue acceleration was conducted (Fig. 3.3, [Nigg 07]). A wooden rod was equipped with an accelerometer directly mounted on the rod (bone sensor). Another accelerometer was mounted with a water bag between the sensor and the rod (soft tissue sensor) to simulate soft tissue mounting. Different experiments were conducted where the rod was dropped from different heights and on different surfaces. Furthermore, the mounting varied between light, medium and strong. In essence, three results were found:

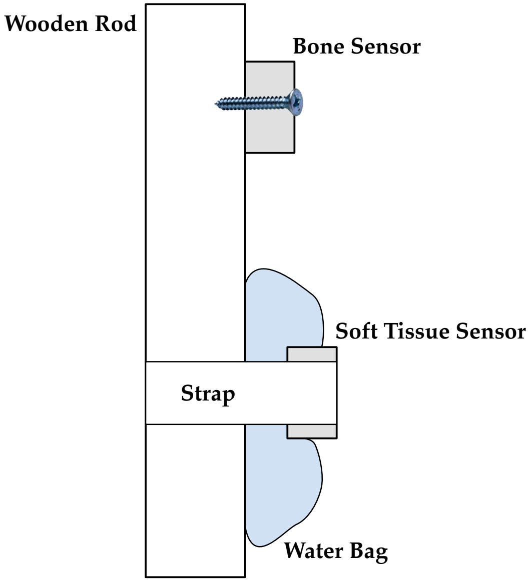

Figure 3.3: Experimental setup to estimate the influence of soft tissue mounting comparing data of a rigidly mounted sensor (bone sensor) and a soft-tissue sensor [Nigg 07].

1. The acceleration values measured with the two sensors varied in most cases and up to 700%.

2. The acceleration measured with the soft tissue sensor can be bigger or smaller than the acceleration measured with the bone sensor depending on mounting and the tightness of the mounting.

3. The hardness of the surface has an influence on whether the soft tissue mounting acceleration is bigger (soft surface) or smaller (hard surface) than the bone sensor mounting acceleration.

The presented experiment showed the challenges of accelerometer-based biomechanical measurement on the example of soft tissue sensor mounting. The presented challenge is also true for angular velocity measurements with gyroscopes as these sensors are also usually mounted on soft tissue. The result underlined that measured acceleration can in general not be directly used as biomechanical metric like segment acceleration.

3.2 Inertial Measurement Unit (IMU)

Different MEMSs can be compiled in an IMU which is the next larger component in the bottom up description of this chapter. IMUs are sensor systems that comprise different sensors and their electronics as well as hardware for data storage, processing and connectivity. They measure relative movement and normally consist of an accelerometer that senses linear acceleration and a gyroscope that senses angular velocity. These sensor units are usually denoted by 6-DOF sensor due to three perpendicular sensor axis for each sensor. Nowadays, IMUs are sometimes enhanced by a magnetometer (also: compass) that senses the earth magnetic field. Such devices are called 9-DOF inertio-magnetic measurement units. The current MEMS miniaturization is one of the driving factors to enable the wearable nature of these devices. Current state-of-the-art sensor modules from selected manufacturers as well as the modules used in this thesis are presented in Tab. 3.1. IMUs are often used in a BSN, which will be introduced in the next section.

3.3 Body Sensor Network (BSN)

A BSN is a special kind of a wireless sensor network. Chong et al. identified wireless networking and microsensing as seminal technologies for a wide range of applications [Chon 03]. The biggest advantage of this technology is its distributed nature and its flexibility due to the wireless linking. Applications in military, geoscience, traffic & positioning and automation & robotics were identified in the early stages of this technology [Chon 03].

The term BSN is used when human biosignals or movement kinematics are processed and networks are worn or positioned around the body. Different terms for this technology like (wireless) body area networks (WBAN, BAN), wireless body sensor network (WBSN) or body (area) sensor networks (BASN, BSN) were defined to describe their short-range wireless nature as well as their focus on body signal analysis.

This work uses the term BSN to describe the network technology as shown in Fig. 3.4. The term BSN signals is used as generic term for data acquired with these networks. BSN signals are further categorized in biosignals (physiological, kinematic, kinetic, ...) that are collected on the human body and kinematics that are collected on sports equipment and environment.

In this thesis, nodes that are mainly responsible for sensing, data logging and data streaming are denoted by sensor nodes. Components

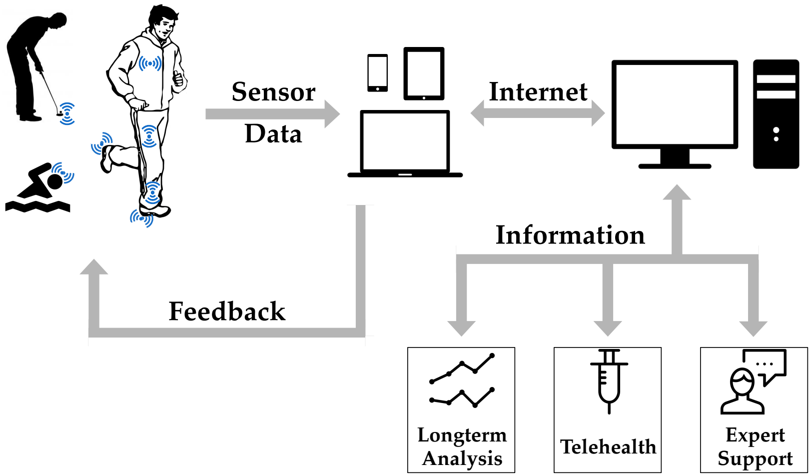

Figure 3.4: Schematic body sensor network (BSN) and its connectivity. BSN signals are sensed and transferred to processing nodes. There, data are either instantly analyzed or transferred to additional services via the internet.

that are capable of aggregating and analyzing data or providing a gateway to further information are denoted by processing nodes. Technically, the miniaturization of MEMS and the decreasing power consumption of low-power microcontrollers were the prerequisites for the success of the BSN technology as they enabled mobile components. BSNs have several technical and applicational advantages which are described next.

3.3.1 Advantages of BSNs

BSNs have different technical advantages compared to other systems that collect and process biosignals and kinematics. First, they are flexible as sensors are wirelessly connected and cabling that might restrict movement is avoided. Second, BSNs are portable or even wearable and, thus, as unobtrusive as possible [Otto 06]. Third, they are capable of providing augmented feedback during activity. These advantages address different challenges in the medical and sports domain where BSNs are mainly applied [Hans 09]. These advantages result in several application opportunities in the medical domain which are described in the next paragraph.

The wearable nature of BSNs addresses two weaknesses of patient data collection: imprecision (qualitative observation) and undersampling (infrequent assessment) [Hans 09]. In contrast, BSNs are capable of collecting multi-dimensional and multi-modal quantitative data over a longer period of time. Furthermore, data are user-centric and its analysis therefore facilitates individual care. Another advantage of these systems is their ability for telehealth applications (Fig. 3.4) and alert functionality as data can be remotely collected and digitally transmitted with a processing node like a personal computer or smartphone [Hans 09]. Additionally, if such a processing node exists, BSNs are capable of giving augmented feedback about the current health status (Fig. 3.4, [Otto 06]). The described advantages can be transferred to other domains like sports where these concepts are not as established as in the medical domain. The next paragraph summarizes the advantages of BSNs for their application in sports.

The properties of frequent assessment and quantitative data are also of great interest for movement analysis and performance assessment in sports and exercise [Hans 09]. Long-term data, collected in real-world training or competition situations, support the understanding of fundamental sport-specific biomechanics and their progress over time. Furthermore, the augmented feedback capabilities of BSNs provide an additional extrinsic data source for coaches and athletes, for instance in the motor learning and performance optimization process. Thereby, the sensor modalities and sometimes even the deduced health and movement parameters are likewise suitable for health and sports applications. An overview of ubiquitous computing in sports with a section on wearable computing is given in [Baca 09]. The advantages of BSNs as described for the medical field are also valid for the sports domain.

3.3.2 Signal Modalities

BSNs are capable of collecting different kinds of data. Modalities of interest that can be collected with BSNs are environmental, physiological, kinetic and kinematic signals [Hans 09]. Examples for relevant environmental data are temperature, pressure and location (GPS). Physiological data of interest are e.g. the electrocardiogram (ECG) and the electromyogram (EMG). Important kinetic information are ground reaction forces (GRF). Examples of relevant kinematic data are acceleration, angular velocity and orientation in relation to the earth magnetic field.

3.3.3 Components

BSNs for sports and medicine comprise different components and are connected in various ways (Fig. 3.4). BSN signals are acquired with sensors that are mostly body-worn. However they can also be integrated in equipment to measure kinematics that have their origin in human movement. Processing nodes like smartphones, tablets or personal computers act as feedback device and provide a gateway to the internet. With this connectivity, long-term data can be archived and analyzed, remote medical care can be realized as telehealth applications and expert support can be retrieved.

There is a large variety of components and technologies that were used in BSN applications. Depending on the application, systems were for instance optimized for data logging or robust data transmission. Other systems shifted the analysis part to an internet-based service. The next sections introduce the sensor and processing node hardware that was used throughout this thesis. These are generic off-the-shelf components that were successfully used for a wide range of prototype applications in sports and health. BSN applications using this hardware are presented in chapters 6 to 8.

Sensor Node (Shimmer™ platform)

Sensor nodes are mainly used to sample, store and transmit data. The Shimmer wireless sensor platform (Realtime Technologies Ltd., Dublin, Ireland) provides configurable low-power sensors for a wide range of applications [Burn 10]. The presented analyses used version 2 and 2R that consisted of a baseboard and additional daughterboards. The baseboard included a MSP430F1611 microcontroller (Texas Instruments Inc., Dallas, USA) that contains 10 KB RAM and 48 KB flash memory and can be configured to run with a processing speed of up to 8 MHz [Texa 15]. Data can be sampled up to 1 kHz depending on the sensor configuration. According to [Burn 10], the microcontroller is particularly applicable for medical sensing applications due to its low power consumption during inactivity. The sensor node also contains a wireless radio module for bluetooth streaming, SD-card slot, rechargeable battery (450 mAh) and 3-D accelerometer (Tab. 3.1) as main components. Additional sensors on the daughterboards (e.g. 3-D gyroscope, Tab. 3.1) were connected via I^2C connection. The dimensions of the sensor were 53 mm × 32 mm × 25 mm and the weight including enclosure was 22 g (Fig. 3.5). The sensor nodes were flashed

Figure 3.5: Sensor node of the Shimmer platform.

with different firmware versions to either act as a data logger or a data streaming device.

In the early stages of the product, the manufacturer mainly provided the hardware and solely software examples. These were a sensor firmware for data logging and radio transmission as well as a PC application capable of communicating with the sensors. However, the research community was eager to adopt the platform and custom connectivity to smartphone devices was developed [Kugl 11].

Due to the elevated need, professional and reliable software was later supplied by the manufacturer. Different software was available for download from the manufacturer website. These were sensor firmware to log data on SD-card or to stream data wirelessly over bluetooth radio. Further, the manufacturer provided a driver to stream data to Android platform devices and a software tool for sensor calibration based on the procedure described in [Ferr 95]. Using these provided components, the realization of a BSN with calibrated sensors and capable of streaming data wirelessly to an Android processing node was possible. The technology of the processing nodes is described next.

Processing Node (Android™ platform)

Processing nodes are commonly used as synchronous data logger or as mobile analysis and display device. For field applications, mobile, portable or even wearable devices are of main interest. These devices require a wireless radio interface to the sensor nodes, processing capabilities and components for user interaction (keypad, screen, microphone, speaker, vibration, ...). Optionally, processing nodes act as an interface to the internet to store or retrieve additional information (Fig. 3.4).

One of the most popular software platforms for mobile devices like smartphones and tablets is Android (Open Handset Alliance, Mountain View, USA). The Open Handset Alliance consists of chip makers, handset manufacturers, software developers, and service providers and was formed in 2007 to develop a joint software platform for mobile devices. The driving force in this alliance is the company Google (Google Inc., Mountain View, USA) which also provides documentation, tools, forums and the Software Development Kit (SDK) for the popular Android platform [Annu 12].

The success of Android builds upon three principles: complete, open, free. The platform is complete in a sense that the software framework is based on a secure operating system (OS) and enables rich application development. It is open as developers have unrestricted access to the device features when creating apps. And it is free as no licensing fees have to be payed for developing, distributing or selling Android applications. Additionally, development tools are available at no cost [Annu 12].

From a technical perspective, Android is a Linux OS and runs each application in its own virtual machine. These virtual machines (VM) are optimized for mobile device hardware (memory footprint, application loading and concurrency, ...) and called Dalvik VM. This managed application principle has the advantage that VMs are independent from each other and that a single app cannot cease the operation of a device. Applications are programmed in Java and register for privileges like network or camera access, user contact information or location. The Android platform comes with integrated native applications like a web browser or a contact manager. However, all applications use the same application programming interface (API) and therefore share the same privileges of using the device resources [Annu 12]. Detailed descriptions and more information about the software development for the Android platform can be found in [Annu 12, Hell 14].

3.4 Embedded System and Cyber-physical System (CPS)

An alternative to creating a BSN application with off-the-shelf components is the design of a custom embedded system or custom cyber-physical system (CPS). An embedded system is an information processing system that is embedded in an enclosing product [Marw 03]. CPSs, in contrast, describe systems where multiple embedded systems physically interact with the environment and connect to each other via networks [Lee 08a]. Generally, the design of such systems provides a more problem-specific solution to a BSN application. Thereby, a component of the BSN can be an embedded system or the complete BSN can be designed as CPS.

The general processing loop, adapted from [Marw 10], illustrates the main parts of a CPS (Fig. 3.6). Marwedel [Marw 10] showed an autonomous processing loop without human interaction. In this work, this concept is enhanced for BSN application and user interaction was introduced as additional building block. Human interaction can either target the actuator or the physical environment directly. Thus, the environment can be changed autonomously by the system loop (actuator) or by user interaction (actuator, environment). In the field of sports, an autonomous change could be an automatic adaption of equipment like the adaption of shoe cushioning or adaption of racket rigidity. A user interaction might be triggered with a system recommendation e.g. to enhance training speed or adapt the shoe cushioning manually.

In off-the-shelf BSNs, hardware is more or less predefined. In contrast, the selection of sensor, processing and actuator hardware is a key design aspect and often the driving force for a custom embedded system solution. Often, existing components cannot be integrated for wearable use, do not feature necessary actuators or consume too much energy. The actual sensor and actuator modality is highly dependent on the application. For example, a sensor and actuator system for a tennis racket rigidity adaption will differ from a system that adapts apparel according to muscle fatigue. The following considerations will therefore focus solely on processing hardware selection. Different components and technologies can be applied for data processing on an embedded system. They have different properties that need to be considered when components are selected in the system design phase. These properties are performance, programming flexibility, hardware cost, time-to-market and energy consumption [Teic 97, Gans 08]. Energy efficiency and programming flexibility were identified as the main conflicting goals [Marw 10].

- **Application-specific integrated circuits (ASICs)** implement specific functionality in a dedicated hardware circuit. ASICs are high-performance, both from a functionality and energy perspective. However, development is time-consuming, functionality cannot be changed once an ASIC is produced and costs are only affordable on high quantity.

- **Field programmable gate arrays (FPGAs)** are configurable hardware and reflect the intersection between hardware and software. Thus, they are programmable and therefore more flexible than ASICs but also more energy demanding.

- **Application-specific instruction set processors (ASIPs)** are specialized processors with a limited instruction set. This hardware component can be used in a more flexible manner with shorter time-to-market and is available for lower cost. However, ASIPs do not reach the performance of ASICs or FPGAs and consume more energy.

- **Digital signal processors (DSPs)** or special purpose microcontrollers are even less energy efficient and slower compared to the architectures above. However, their advantages are lower cost, higher flexibility (larger instruction set) and faster time-to-market.

- **Low-end microcontrollers** are bottom regarding energy efficiency and computational performance. However, they are flexible, low-cost and development cycles are short. Due to their low cost and processing power they are a valuable alternative for simple but flexible signal processing tasks.

When systems are getting more complex and/or comprise multiple components, hardware-software-codesign approaches are needed to realize the most efficient product according to the specification.

In contrast to personal computers and their operating systems, CPSs are usually real-time systems and therefore require a corresponding operating system. In such real-time operating systems (RTOS), timing behavior has to be predictable. Therefore, tasks have to be scheduled in a defined manner to ensure continuous and precise operation. The aspect which has the biggest design influence regarding real-time capability is the worst-case execution time (WCET). It describes the upper bound of the runtime of a task considering all possible inputs and initial states. As processor architectures get more and more complex, only upper bounds for the WCET can be estimated. Of course, a precise estimation is of great interest for an efficient system design.

Different design objectives have to be considered when developing custom embedded systems or CPSs. This is especially a challenge if a hardware-software-codesign is used. From a research and development perspective it is also important how systems are used.

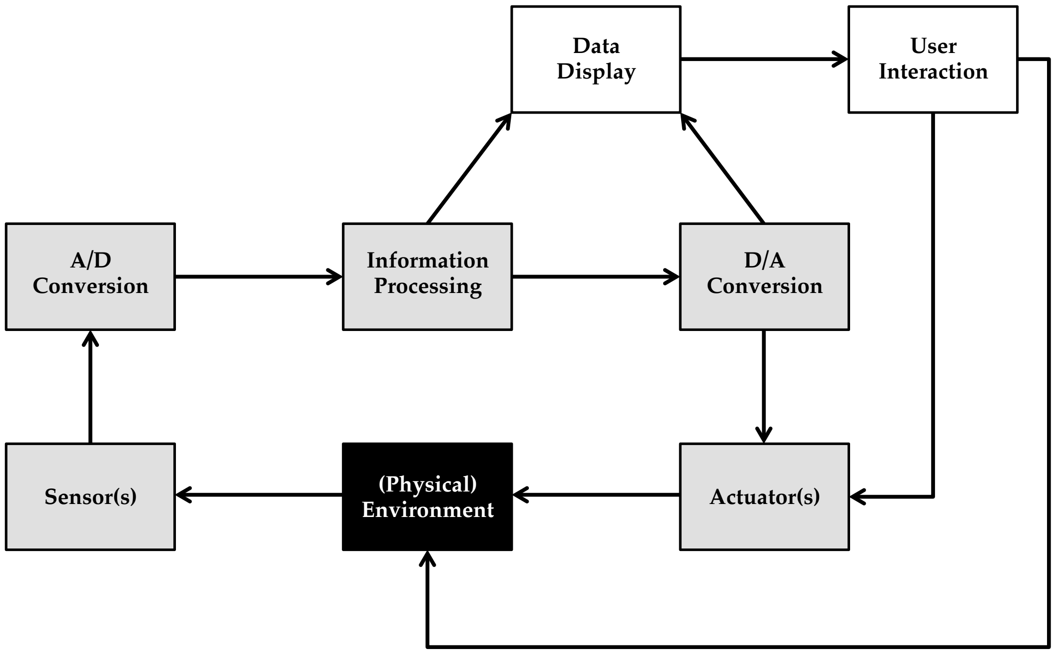

Figure 3.6: Processing loop of a cyber-physical system (CPS) (adapted from [Marw 10]). The grey boxes illustrate the hardware system and the white boxes illustrate the operator interaction.

Specific products are often optimized for maximum performance and minimum energy consumption as costs can be calculated with possibly high quantities. However, in prototyping, a lot more flexibility and multiple development iterations are needed during development. Additionally, only small quantities are needed as prototypes. Research platforms represent the use case that requires the highest amount of flexibility as hardware is potentially used in different projects and development is not based on a dedicated specification. In addition, demands on safety and reliability are not as high in research and prototyping as in final products. The following selected aspects were generally identified as important design objectives for wearable embedded systems and CPSs [Marw 10]:

- **Worst-case performance** of tasks is important to guarantee predictable scheduling in real-time processing.

- **Energy and power consumption** are crucial to make systems wearable as batteries are the most important contributor to form factor and weight.

- **Temperature and thermal behavior** have to be considered as sensor properties change when wearable system are used in different outdoor conditions like rain, frost and heat.

- **Reliability** is an important aspect in medical applications. This is especially important if actuators interact autonomously with patients.

- **Numeric precision** is an important aspect when porting prototypes to embedded implementations and depends on the signal processing algorithm. Normally, a minor loss of numerical precision can be tolerated.

- **Form factor, weight, robustness and usability** are crucial aspects in wearable systems. If a good integration in equipment, clothing or shoes cannot be guaranteed, systems won't be adapted by the users. In the medical field, users often do not want to be identified as patients making stigmatization a major issue. As systems are used during activity, an adequate level of robustness has to be maintained.

- **Cost** is of course a major driving force for product related research and development.

In essence, an important aspect in custom embedded system design is the selection of processing hardware. Processing hardware determines the key aspects of the embedded system, is highly related to the presented design objectives and can be selected according to the signal processing task at hand.

3.5 Comparison of Off-the-shelf and Custom Hardware

The previous sections introduced the technology of MEMS and BSN. Off-the-shelf components to realize such a wearable computing system were introduced. Furthermore, the underlying technology and design considerations for embedded systems and CPSs were summarized. This section compares the two development approaches of creating BSN applications based on standard components and the design of custom CPS. This differentiation is important to understand the use cases of the design consideration presented in chapter 5.

Off-the-shelf hardware as BSN component provides the advantage of short development time. Hard- and software are available and application development can be started right away. Regarding sensor nodes, platforms like Shimmer offer high flexibility in connected sensors, sampling rates and data ranges. Different BSN signal modalities (e.g. acceleration, ECG) and characteristics (e.g. low-frequency human movement, high-frequency vibration) can be recorded with the same sensor hardware. Regarding processing nodes, the popularity of the Android platform offers a vast amount of devices for data analysis and display. Additionally, connectivity to the mobile networks and the internet is integrated and the device can be programmed in the high-level language Java.

The disadvantage of such a BSN is the insufficient integration of sensors in shoes, clothing and equipment due to their versatility. This is also true for the processing nodes that provide many features that are not relevant for a specific application. The most important shortcoming of the Android platform are the missing technical real-time capabilities that guarantee a predictable timing behavior. However, in consumer applications this might not be as important if actuators are triggered by user-interaction or if information is not time-critical.

When developing algorithms running on off-the-shelf components, computational resources are predetermined. Thus, algorithms have to be specifically designed to run on the given hardware or have to be optimized for the specific hardware. In principle, off-the-shelf hardware determines the complexity of the signal processing algorithm.

Custom hardware in embedded systems realizes dedicated functionality in an efficient manner. The design methodologies for these embedded systems or CPSs support a high level of integration, low energy consumption and reliability that is beyond the range of off-the-shelf hardware. However, development is time and cost consuming and often only practicable for consumer products. From a research perspective, standard BSN components are the desired starting point for application development. For some BSN applications, a custom hardware prototype is then the next step if e.g. a high level of sensor integration is needed.

Hardware platforms can be specifically selected for an existing signal analysis algorithm. To support this selection decision, a precise estimation of computational cost and memory demand is needed to determine the most efficient solution. In principle, the complexity of the signal processing algorithm determines the selected hardware.

A possible intermediate step between these two approaches is on-node processing. The runtime of sensor nodes can be enhanced with reduced radio transmission and, as sensor nodes often feature RTOS, sensors can be programmed to meet real-time constraints. A remaining challenge is the delay and latency in wireless transmission. An on-node solution is presented in chapter 8 where swimming kinematics are processed on a Shimmer sensor node.

Chapter 4

Sport Science Fundamentals

This chapter gives an introduction to the fundamentals of human movement with reference to sports and exercise. These fundamentals provide the necessary sport-specific background for the applications presented in chapters 6 to 8. First, kinesiology basics are introduced. Second, the principles of motor control and motor learning with emphasis on augmented feedback are summarized. Third, the term digital sports as a connection of pattern recognition, mobile sensing and computing as well as sports is introduced.

4.1 Fundamentals of Kinesiology

This section summarizes the anatomic and physiological principles of human movement and describes the adaptation processes through training. As references for the complete section 4.1, [Wein 10] and [Tort 07] were used. This section provides the necessary background for the plyometric training application (chapter 6) and the swimming application (chapter 8).

4.1.1 Fundamentals of Anatomy and Physiology

Based on the description of the muscular system and the principles of muscle contraction, reflexes and muscle energy supply are introduced. Reflexes are an important aspect for plyometric training described in chapter 6. Muscle energy supply is an important aspect for the swimming kinematics application described in chapter 8. It is important to understand the short-term processes during training and the long-term effects of training to in turn be able to understand the key aspects of the plyometric training and the swimming applications.

Muscular System

Muscles are the basis for every human movement. They can be categorized in three types; cardiac muscles, smooth muscles and the skeletal muscles. These types differ in certain structural and functional characteristics. The striated skeletal muscles (Fig. 4.1) are of prior interest for human movement. A muscle consists of a bundle of myofibers (muscle fibers) which comprise myofibrils. The myofibrils are constructed out of sarcomeres that in turn consist of the smallest functional units of a muscle, actin and myosin filaments. Sarcomeres are separated by Z-lines.

Two main types of muscle fibers exist; slow twitch (ST) and fast twitch (FT) fibers. The ST fibers are characterized by a slow contraction velocity and high fatigue resistance. The FT fibers, in contrast, contract fast and develop more force but fatigue more quickly. Every muscle consists of a genetically predefined ratio of these fiber types.

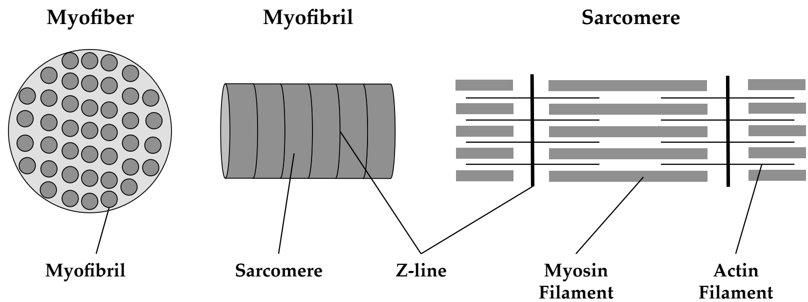

Figure 4.1: Components of a striated skeletal muscle.

Muscle Contraction

The mechanical principle of muscle contraction is the sliding of actin and myosin filaments. Thereby, the filaments telescope by a tilting movement of the myosin heads and the distance between two Z-lines decreases. As sarcomeres along the myofibrils telescope in a coordinate way, the complete muscle shortens and is therefore able to move joints.

On a chemical level, muscle contraction is triggered by the flow of calcium ions from the sarcoplasmic reticulum into the sarcomere. These calcium ions bind the protein tropomyosin and therefore uncover the binding site between actin and myosin. The decomposition of adenosine triphosphate delivers the energy for the tilting movement of the myosin heads.

From a neuromuscular perspective, muscle contraction is organized in motor units (Fig. 4.2). Motor units consist of an alpha motor neuron in the spinal cord, its axon, a motor end-plate on the myofibers and the innervated skeletal myofibers. On muscle innervation, an action potential sent by the motor neuron propagates towards the motor end-plate. The motor end-plate initiates an action potential that travels along the innervated myofibers. This action potential causes the release of calcium in the sarcoplasmic reticulum and the muscle contraction progresses as described.

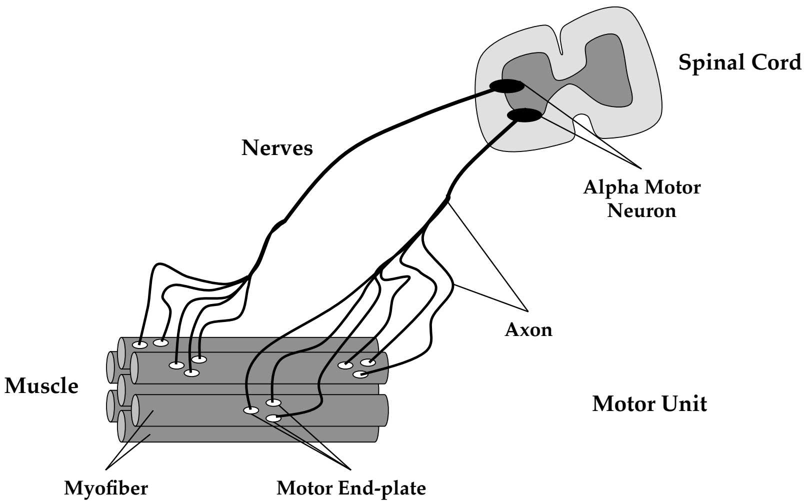

Figure 4.2: Components of a motor unit for muscle contraction.

When a motor neuron fires, all myofibers of the motor unit are innervated. The number of muscle fibers of one motor unit varies between 10 for precise movements (eye muscles) to 1600 for powerful movements (leg muscles). The movement coordination is organized according to three principles. First, fine grained movement control is realized with an increase and decrease of the firing rate of one motor unit. Second, course grained movement control is realized by means of recruiting and coordinating multiple motor units of the same muscle. Third, the maximal force of a muscle is reached when all motor units of a muscle fire at the same time (tetanic contraction). The movement speed is determined by the number of innervated myofibers and their main fiber type (ST, FT).

Reflexes

Muscle contraction is controlled by the central nervous system that consists of the brain and the spinal cord. The motor cortex in the brain is the starting point for voluntary muscle contractions (planning, initialization, control). Reflexes are handled in the spinal cord and do not require cerebral activity. Reflexes can be categorized in two groups, monosynaptic and polysynaptic reflexes.

In monosynaptic reflexes, a single synapse in the spinal cord is involved. The muscle whose receptor caused the reflex also receives the result. An example for such a reflex is the stretch reflex where a muscle reacts with contraction when quickly stretched. Thereby, muscle spindles act as sensory receptors for the length of the muscle.

In polysynaptic reflexes, multiple synapses and therefore muscles are involved. Other muscles than the one that caused the reflex are innervated. An example for such a reflex is the golgi tendon reflex that protects a muscle from high tension. Thereby, the muscle that caused the reflex lengthens and relaxes while the antagonist muscles contract. In this reflex, the golgi receptors, situated at the intersection of muscle tendon and myofibers, sense the tension of a muscle.

Muscular Energy Supply

On contraction, a muscle hydrolyzes adenosine triphosphate (ATP) to adenosine diphosphate (ADP) to gain energy to tilt the myosin heads:

$$\mathrm{ATP} + \mathrm{H_2O} \rightleftharpoons \mathrm{ADP} + \mathrm{P} + \mathrm{H^+} + \mathrm{Energy} \tag{4.1}$$

Thereby, phosphate (P) and hydrogen ions (H^+) are generated.

ATP storage in the myofiber is limited and ATP can be resynthesized with different processes. One possibility is the production with creatine kinase as catalyst:

$$\mathrm{CP} + \mathrm{ADP} \xrightleftharpoons{\text{creatine kinase}} \mathrm{ATP} + \mathrm{Creatine} \tag{4.2}$$

This process consumes creatine phosphate (CP) which is stored in the muscle cell and creates creatine.

The processes in (Eq. 4.1 and 4.2) provide energy supply for around 6-8 seconds at the beginning of muscle activity. Depending on the amount of energy consumption per time which is in turn determined by the movement intensity, additional ways of providing ATP exist. These are consuming glucose using anaerobic processes (without oxygen) or aerobic processes (with oxygen).

The anaerobic energy supply

$$\text{Glucose} \rightleftharpoons 3\,\text{ATP} + \text{Lactate} \tag{4.3}$$

is fast and the amount of energy per time is high. However, as a high concentration of lactate blocks this process, it can only be used for a limited amount of time (20-30 s). Furthermore, the process delivers a low amount of ATP. The anaerobic oxidation is suitable for short intensive activity.

The aerobic energy supply

$$\text{Glucose} + 6\,\text{O}_2 + 31\,\text{ADP} + 31\,\text{P} \rightleftharpoons 31\,\text{ATP} + 6\,\text{CO}_2 + 6\,\text{H}_2\text{O} \tag{4.4}$$

is slow and the amount of energy per time is low. However, a high amount of ATP is created. As an alternative to glucose, the aerobic energy supply is capable of using fats and proteins to synthesize ATP. The aerobic oxidation is suitable for long activity with low intensity.

The energy supply during activity is always a mixture of these processes. However, depending on the intensity, one kind of energy supply dominates.

4.1.2 Training and Physiological Adaptation

Athletes train to reach an increased performance level. The training stimulus thereby causes physiological, anatomical and other adaptations that represent an enhanced performance level. The main principle is called supercompensation (Fig. 4.3). Supercompensation describes the process where, after fatigue due to a training stimulus, a recovery stage leads to an increased performance potential level. Continuous repetition of appropriate stimuli lead to physiological adaptations.

The physiological adaptions are e.g. determined by the frequency, type and intensity of the training stimuli. The optimal timing for the following training stimulus would be the peak of the increased performance potential (Fig. 4.3). The type of adaptation determines the type of stimulus that is needed for a corresponding performance gain. Untrained athletes, of

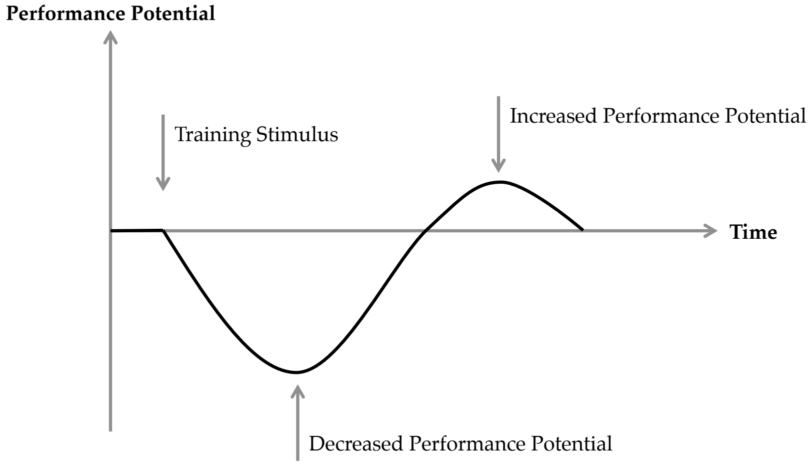

Figure 4.3: The principle of supercompensation through training. The performance potential level of an athlete decreases by a training stimulus and reaches an increased level due to physiological adaptation. Adopted from [Wein 10].

course, need lower stimulus intensities for a performance gain compared to well trained athletes.

The next section describes aspects of muscle physiology adaption and energy supply adaption through athletic training. Muscle physiology adaption is an important aspect of the plyometric training application described in chapter 6. Energy supply adaption is an important aspect for the swimming application described in chapter 6.

Muscle Physiology Adaptation

The adaptation of the neuromuscular system consists of three components. These are hypertrophy, intramuscular coordination and intermuscular coordination. Hypertrophy describes an increase in the myofibril size and therefore an increase in muscle mass. An increase in intramuscular coordination means that more motor units can be recruited and coordinated. Thus, the contraction of a single muscle involves more motor units and is more precisely coordinated. Intermuscular coordination, in contrast, describes the coordination of different muscles. An increase in this ability therefore describes the improved coordination of motor units of different muscles.

Another aspect of muscle physiology adaption regards the myofiber types. Endurance training influences the fiber type as FT fibers change their shape to ST fibers. Speed training, in contrast, stimulates the hypertrophy of FT fibers. Consequently, the muscle mass ratio shifts towards the FT fiber type.

Energy Supply Adaptation

The main adaptation regarding the cellular muscle supply is the increase of energy sources. In endurance training, glucose in the muscle cells and the liver as well as intracellular fat storage increases. Strength and speed training stimulates the storage of glucose and CP in the muscle cell. The increased energy storage potential is accompanied by an improved enzyme activity. Additionally, all processes that create ATP are optimized according to the predominant type of training.

Another aspect of energy supply adaption is the increased ability to eliminate lactate. ST fibers can adapt to remove lactate faster and can therefore maintain the anaerobic energy supply (Eq. 4.3) for a longer amount of time.

4.2 Motor Learning and Control

This section describes the fundamentals of motor learning and control. First, terms are defined. Second, an introduction on motor control is given. Third, aspects of motor learning are summarized. Fourth, the aspect of augmented feedback in motor learning and control is discussed. If not indicated differently, [Magi 11] was used as reference throughout section 4.2.

The motor learning and control fundamentals described in this section provide the necessary background for the golf putting application described in chapter 7. Special emphasis will be put on the aspect of augmented feedback. The possibility of augmented feedback is an important feature in many BSN applications. Therefore, principles and challenges of augmented feedback applications are described in more details. This description provides the necessary background to understand the potential of BSN applications regarding augmented feedback training.

4.2.1 Terms and Definitions

Voluntary movement activities of the body, limbs or head with a specific goal are defined as motor skills (also: actions). We will focus on motor skills that have to be learned or relearned (e.g. walking, ball throwing). Motor skills comprise movements (body, limbs, head) as component parts and can be categorized in:

- Muscle size involvement
  - Gross: Primarily large muscles required
  - Fine: Primarily small muscles required
- Specificity of action start and end
  - Discrete: Simple movement, distinct start and end
  - Continuous: Repetitive movement, arbitrary start and end
  - Serial: A series of discrete skills
- Stability of environmental context (surface, objects, people)
  - Closed (also: self-paced):
    Environmental objects are stationary
  - Open (also: externally-paced):
    Environmental objects are moving

Motor skills are studied as motor control and motor learning. Motor control investigates how the neuromuscular system functions to perform motor skills. The principles of motor control are essential to develop strategies for motor learning which is defined as the acquisition and performance enhancement of motor skills. For motor learning, the behavioral and neurological changes as well as the variables that influence these changes are of interest. Motor skills can be studied by investigating how movements are controlled (motor control studies) and how humans learn movements (motor learning studies).

4.2.2 Motor Control

Motor control contains two essential issues; coordination and solving the degree-of-freedom problem. Coordination means the simultaneous movement of head, body and limbs to reach the movement goal. This patterning needs to be relative to environmental objects and events. The degree-of-freedom problem described the fact that independent elements (joints, limbs) can act in a number of ways and that these components have to be adequately controlled in a complex task [Bern 67]. A solution to this problem results in a specific movement.

As an analogy to mechanical engineering, motor control can be categorized in open-loop and closed-loop systems (Fig. 4.4). They are general theories of how the central and peripheral nervous system (movement control center) initiates and controls action of the limb, head and body muscles (movement effectors). Both systems differ in the feedback branch, which includes proprioceptive as well as visual and auditory internal feedback. In an open-loop system, the movement instructions contain all necessary information for movement execution. Feedback is not needed, not used or not available. In a closed-loop system, feedback is transmitted to the movement control center to constantly update or change movement instructions.

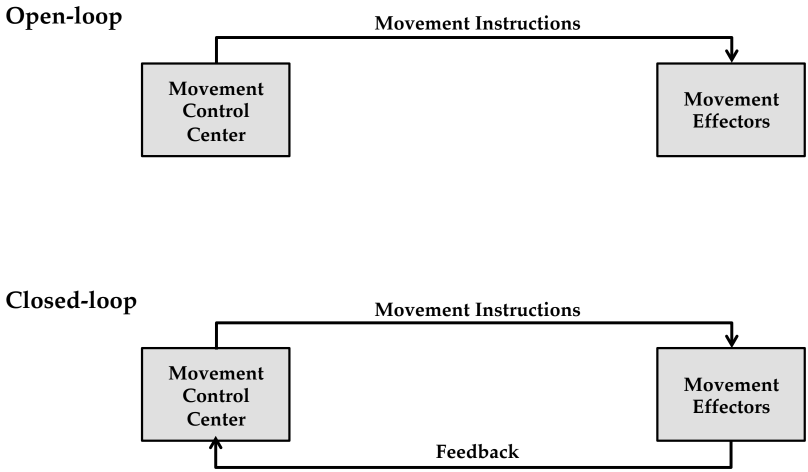

Figure 4.4: The open-loop and the closed-loop control system for motor control. Adapted from [Magi 11].

There are two main types of motor control theories; motor approaches and action approaches. The following paragraphs describe the Generalized Motor Program as an example of the motor approach and the Dynamic Pattern Theory as an example of the action approach.

Generalized Motor Program (GMP)

The general idea of this theory is that a GMP controls a class of motor skills. All motor skills of the same class share invariant features that form the fundamental pattern of the class of actions. The invariant features are consistent from one performance to the other. A specific action of this class is based on the invariant features but enhanced with movement-specific parameters. These parameters are responsible for the desired variation in the different instances of the same class of motor skills. Using the example of walking with different speeds, a GMP exists that defines the fundamental pattern of the motor skill. This fundamental pattern, e.g. the percentage of each walking phase, is constant throughout all speeds. For different speeds, however, the actual timing of the walking phase changes. Thus, movement-specific parameters change with the instance while the fundamental patterns remain. With increasing walking speed, the motor skill changes to running at some stage. In the GMP theory, the two fundamentally different motor skills of walking and running are explained with two different motor programs for walking and running.

The Schema Theory, an abstract representation of rules, explains how a GMP can be used to control coordinated movement. It consists of two parts; the GMP and the Motor Response Schema. The GMP thereby controls the coordination patterns using the invariant features. It is used in movement initiation and is an open-loop system. The Motor Response Schema, in contrast, consist of rules governing the performance of the action using the parameters. It is used to control the movement in a closed-loop system.

Dynamic Pattern Theory (DPT)

The DPT (also: Dynamic Systems Theory) regards motor control from a nonlinear dynamics perspective where physical and mathematical laws govern the behavior of abrupt system changes. These changes are neither continuous nor linear but triggered if a control parameter (e.g. speed) changes. These dynamics are reflected in the concept of stability. Stability describes a stable state of behavior to which a slightly perturbed system will return. Such stable states in human motor control are denoted by attractors (also: attractor state) and characterized by minimal behavioral variance and high energy efficiency. The overall system behavior is defined by order parameters that enable a coordinated pattern of movement and distinguish patterns from each other. Order parameters are changed by varying control parameters. In the DPT, different walking speeds and the transition to running are explained with the two stable phases of walking and running. The control parameter speed triggers an abrupt system change, a change of the order parameters and the transition from the walking to the running state.

Two main concepts support the DPT; self-organization and coordinative structures. Self-organization means that a stable pattern of behavior emerges due to certain conditions. In contrast to a motor program organizing the behavior, the behavior evolves according to the conditions. Coordinative structures act as an ensemble in a way that the functional collection instead of its parts (e.g. joints, muscles) is controlled. Thus, the degrees of freedom are reduced and the action can still be completed if parts of the ensemble are inactive.

4.2.3 Motor Learning

The assessment of learning is essential for every motor learning process and the questions whether something was learned and what was learned are of main interest. However, learning cannot be observed directly but has to be inferred from performance. The term performance describes an observable execution of a skill at a specific time and in a specific situation. A specific performance might be a temporary snapshot that is not achieved through skill learning. Learning, however, describes the permanent change of a person's potential to perform a skill due to practice or experience. Performance, a single instance of skill execution, is influenced by performance variables like alertness, fatigue as well as the environmental setting and does therefore not directly reflect learning. There are five characteristics that reveal skill learning:

- Improvement: Higher level of skill after practice
- Consistency: Smaller variation in performance
- Stability: Less influence of perturbations (e.g. stress, wind) on performance
- Persistence: Performance potential lasts over time
- Adaptability: More successful performance in changing circumstances

Assessment Tests

The assessment of learning during practice might be misleading as performance variables and the existence of feedback (or a different kind of feedback) might influence performance. Especially the controlled environment in a lab or gym can influence performance and therefore alter the learning assessment. Therefore, different tests like the retention test and the transfer test are executed. The retention test, a test after practice has ceased, assesses the persistence of the performance level achieved during practice. The transfer test, on the other hand, investigates the performance under different conditions and assesses the adaptability aspect of learning.

Learning Curves and Plateaus

The learning process can be expressed as curve where the performance is shown over time. Different types of curves exist and the negatively accelerated curve (Fig. 4.5) is accepted as the most common one. It describes a strong improvement at the beginning of the learning process and a smaller amount of improvement later on. A general explanation for this kind of curve might be that a lot of errors occur in an early learning stage. However, they can be corrected quite easily. As the error rate decreases, an improvement is harder to achieve.

Figure 4.5: A negatively accelerated learning curve.

Learning curves also illustrate plateaus where a learner experiences no improvement between two periods of improvement. One reason for this phenomenon might be the transition between learning phases or different aspects of the skill. Other reasons are poor motivation, muscular fatigue or lack of attention. Additionally, the performance measure might cause ceiling or flooring effects as it does not reflect further progress.

Stages of Learning

Learners go through distinct stages when learning a skill. The transitions are gradual from one stage to the other and the progress depends on various factors like the nature of the skill, the practice conditions and the character of the learner. This paragraph describes two different models of learning stages.

The Fitts & Posner model [Fitt 67] describes three stages. The first one is the cognitive stage in which the learner focuses on cognitively oriented problems. Examples are the correct position of body parts and desired goal of the skill. There is a high cognitive load as the learner has to listen to instructions and receive feedback from the instructor. This phase is characterized by a large amount of errors and high variation. The second stage is the associative stage where the learner has learned to associate environmental cues with the movements. Basic fundamentals of the skill are acquired and the skill is refined in this stage. Skills are performed more consistently and variability decreases. The third stage in this model is called autonomous stage where the skill is performed in an autonomous or habitual manner. Skills are performed without conscious thought and together with other tasks at the same time.

The Gentile model [Gent 72] describes two stages, the initial stage of learning and a later stage of learning. In the initial stage, the learner tries to achieve two main goals. First, the acquisition of the movement coordination pattern. Second, learning to distinguish between regulatory and non-regulatory environmental conditions. Thereby, non-regulatory conditions do not influence the execution of the skill. The phase is characterized by exploring movement possibilities e.g. with trial and error and cognitive problem-solving activity. At the end of the stage, the learner has developed a movement coordination pattern that is neither consistent nor efficient. In the later stage of learning, the learner seeks for adapting the skill to specific demands, increasing the consistency and performing the skill with an economy of effort. For closed skills, the main goal is fixation so that the movement pattern is performed correctly, consistently and efficiently in each trial. For open skills, the main goal is diversification of the skill to be able to successfully modify the movement pattern according to the environment at hand.

4.2.4 Augmented Feedback

In motor learning and control, task-intrinsic feedback (also: internal feedback) is always available through sensory perception. Humans use different sensory systems consisting of receptors, afferences and corresponding brain regions to process the sensed information. In movement, the most important sensory modalities are visual, auditory, tactile, kinesthetic and vestibular. Kinesthetic information refers to the perception of the pose of the body and vestibular information to the body balance information [Wein 10, Sigr 13].

According to [Sigr 13], augmented feedback is given by an external source like a trainer or a display. In this context, a display describes all possible feedback modalities like screens, speakers or robots. It provides information that the learner is e.g. not able to measure with his own senses or that he is not able to process simultaneously. Augmented (also: external, task-extrinsic) feedback can either be given during motor task execution (concurrent, online, real-time) or after task completion (terminal) [Sigr 13]. In this work, the term augmented feedback as well as the terms concurrent feedback and terminal feedback will be used.

According to the definitions in [Magi 11, Schm 08], augmented feedback can also be classified by the type of information that is given. It is distinguished between “knowledge of performances” (KP) and “knowledge of result” (KR). KP feedback contains information about the quality and details of the movement that lead to the performance outcome. KR feedback, in contrast, presents the outcome of performing the skill or whether the goal of the skill was achieved or not.

Augmented feedback plays a major role in the motor learning process. One aspect is that augmented feedback provides information of whether the movement is appropriate for achieving the goal of the skill. This information can accelerate and simplify the learning process. Another aspect is that augmented feedback can motivate the learner to adhere to training or rehabilitation programs and continue trying to achieve their goal.

A comprehensive state-of-the-art review is given in [Sigr 13] and will be presented in the following paragraphs. Sigrist et al. [Sigr 13] are the reference for the remaining part of chapter 4.2.4. Selected topics of augmented feedback will be described in more detail. Thereby, the term haptic describes tactile and kinesthetic perception.

Task Complexity

Different aspects influence the effectiveness of augmented feedback for simple and complex motor learning tasks. Simple tasks are characterized by a single degree of freedom, can be mastered in a single session, are mostly artificial and characterize learning in the associative stage of the Fitts & Posner model. Complex tasks comprise several degrees of freedom and cannot be mastered in a single session. They characterize learning in the cognitive stage of the Fitts & Posner model.

Concurrent Feedback

It has been found that concurrent feedback in simple tasks leads to superior performance but the improvements are not sustainable. Thus, they are lost in retention tests. The reason might be that subjects strongly rely on the feedback information and reduce the use of task-intrinsic feedback. In contrast, subjects can highly benefit from concurrent feedback when performing complex tasks in the cognitive phase of the Fitts & Posner model. It is believed that concurrent feedback makes the task more manageable and therefore avoids cognitive overload.

Feedback Frequency

In the associative and autonomous phase (Fitts & Posner model) of complex task learning, less frequent concurrent feedback or terminal feedback achieved superior results. Trials without feedback are needed to develop a persistent internal movement representation. A general guideline for the frequency of feedback states that beginners need higher frequencies than experts. Fading feedback, the constant reduction of trials where feedback is given, seems to be very promising. However, the fading rate is hard to determine so that self-controlled feedback was proposed as promising solution. It was also found that terminal feedback is most efficient when it was accessed after trials where subjects believed that they have performed very well.

Terminal Feedback

Terminal feedback should be delayed a couple of seconds to enable the evaluation of task-intrinsic feedback before the augmented feedback is received. Furthermore, the feedback should be prescriptive rather than descriptive. Meaning, it should describe how to improve rather than describing errors.

Visual Feedback

Visual augmented feedback is considered to be the most important feedback modality. However, the task complexity, skill level and feedback presentation determine its effectiveness. The feedback should be designed to enable a parallel processing of visual and intrinsic information. This is specifically important when designing sophisticated visualizations that - if inappropriate - can even cause negative effects. The actual design strongly depends on the type of movement and number of important variables. Generally, a small number and only the most important parameters should be considered for feedback and presented in an abstract and unambiguous way.

Auditory Feedback

Auditory augmented feedback has the advantage of not overwhelming the user when the task itself needs strong visual attention. Thus, it might be a good augmented feedback modality as additional visual feedback overloads cognitive processing. Auditory feedback can be classified in alarms, sonification of movement variables and sonification of movement errors. Alarms are easy to interpret but the user cannot recognize to what extend the movement should be altered. Sonification of movement variables describes the change of sound parameters with the change of data values. To be useful, a variable must be suitable for sonification and must be linked to an adequate movement representation. A combination of these approaches is the sonification of movement errors where the deviation of the target movement is encoded in sound. Generally, auditory feedback is more abstract and appropriate designs to avoid misinterpretation have to be chosen.

Haptic Feedback

Haptic augmented feedback was mainly investigated for simple tasks with the goal of providing position control. Thereby, the user is directed to a predefined movement e.g. by a robot. The robot guides the learner by restricting the target movement trajectory. With this kind of feedback the user “feels” how the specific movement should be performed correctly. The concept is most interesting for novices without a good understanding of the target movement, thus without an established motor program. However, when learning progresses, the low variation in a robot movement might hinder the subject from learning. Beside robotic applications, vibrotactile displays have been introduced for different sports applications. They represent an alternative modality to visual and auditory feedback but are not as straightforward to use. Their effectiveness needs to be further evaluated.

Multimodal Feedback

Multimodal augmented feedback applications (e.g. audiovisual or visuo-haptic) incorporate more than one feedback modality. Their advantage is that different aspects of a movement can be addressed simultaneously. This is beneficial in complex motor tasks where this kind of feedback is believed to reach a workload reduction for the learner. Especially the spatial and temporal aspects of a movement can be trained simultaneously with visual and auditory feedback.

Key Findings

The literature review revealed several key findings for the future design of feedback applications for motor learning.

- Learning of complex tasks in the cognitive stage (Fitts & Posner model) can be well supported with concurrent feedback.
- When learning progresses (associative stage of Fitts &Posner model), feedback should be reduced (fading frequency) or terminal feedback should be used.
- Self-controlled feedback frequency highly motivates the learner and adapts to the current skill level.
- Adaptive feedback monitors the skill level and selects movement features and errors accordingly.
- Feedback should force to correct the most relevant errors and should be prescriptive.

Chapter 4.2.4 provided the necessary background for the design of augmented feedback applications. Many aspects like the kind of feedback display have to be taken into account in an early phase of development.

4.2.5 Biofeedback Training

The kind of feedback that has previously been described as augmented feedback is sometimes misleadingly denoted by the term biofeedback. This is especially the case if physiological, kinematic or kinetic data of a user are used to generate feedback. This work uses the term biofeedback as defined in [Basm 81, Magi 11] who restrict the biofeedback to the use and altering of physiological data. To understand the differences and similarities between augmented feedback and biofeedback the latter will be shortly summarized.

Biofeedback training is a method to learn the manipulation of physiological processes [Basm 81]. Thereby, some kind of feedback (e.g. visual, acoustic, tactile) of the physiological parameter to be altered is employed. The goal is to make unconscious physiological activity accessible for the athlete. Having this feedback, the athlete trains conscious control over the physiological processes. A simple example would be the controlled decrease of the heart rate using a pulse monitor. Early research used EMG, EEG and cardiovascular signals for biofeedback training and introduced the treatment of neurologically and orthopedically impaired patients as main application [Basm 81].

Still, rehabilitation of neurologically and orthopedically impaired patients is a main application of biofeedback training. A lot of research has been conducted on stroke rehabilitation, for instance to treat upper extremity spasticity, train hand function or the ability to perform activities of daily living [Doga 12]. Another application of biofeedback training is the support of headache patients to manage pain conditions [Nest 08]. Often, the training with biosignals like peripheral skin temperature, blood-volume-pulse and electromyography is combined with relaxation and stress management techniques. Another application of biofeedback training is the prevention and treatment of psychological and physiological illnesses like depression or cardiovascular diseases. One cause for these diseases is an increased stress response level caused by the disfunction of the heart rate variability and baroreflex system. Specific breathing and electromyography training programs reached encouraging results, but have to be investigated for long-term effects [Whea 10].

In the sports science field, biofeedback training is used to enhance focus and reduce undesired movements in sports like archery, curling and golf [Wein 10]. The reduction of the yips-phenomenom occurring in golf putting and chipping is one reported application. Athletes with yips show involuntary muscle activity under stress with a large effect on the result of the stroke. The psychophysiologic method of biofeedback training is often combined with pure psychological training approaches like mind training and autogenic training [Wein 10]. Current trends in sports training and research are summarized in [Blum 02]. There, specific programs for biofeedback training are presented and the need for mobile biofeedback devices is mentioned as training is still mostly performed in a lab environment.

Biofeedback training can be classified as a subcategory of augmented feedback. Biofeedback applications are often implemented as BSN. This work will focus on augmented feedback in general.

4.3 Digital Sports

The preceding sections introduced the fundamentals of kinesiology and motor learning. A desired anatomical and physiological adaption can only be reached if the right training stimulus is triggered. This is the case for short-term and long-term training control. It is therefore of great interest for athletes and coaches to assess BSN signals to support correct and precise training. In motor learning, the importance of augmented feedback was identified as a valuable source of information. However, the kind, timing and modality of augmented feedback is crucial for the resulting effects.

BSN technology is a promising approach to support the optimization of physical training and motor learning. Currently, the use of such systems gains popularity on all performance levels, from professional to recreational. Performance can, in theory and depending on the application, be assessed in the field, on a long-term scale, with unobtrusive equipment and in real-time. Due to the relatively young technology of BSNs, the aforementioned aspects (field data, long-term assessment, unobtrusive recording and real-time analysis) need improvement. They are subject to active research. A term that describes this field of research is digital sports.

According to [Mend 92], important aspects in the context of wearable digital sports applications are parameter precision, relevance of information, timing and availability of results as well as the influence of the technical system on performance. Information is only useful in a training intervention if correct and relevant results are available at the right time. Further, the technical system should not influence the athlete in his performance in any way. According to Wagner [Wagn 06], two main requirements for measurement and information systems in the sports domain exist. These are the measurement of relevant parameters with sufficient precision and the minimal influence of the athlete's movement. These requirements are also valid for wearable digital sports applications. Liebermann et al. [Lieb 02] reviewed different kinds of digital sports applications and specifically their use as augmented feedback source. They concluded that the introduction of information technology in sports performance enhancement appears to be beneficial for effective and efficient learning. Further, augmented feedback is of major importance for motor learning and should be considered in the normal practice scheme. Although BSN technology was not considered in [Lieb 02], it is assumed that the above mentioned conclusions are also valid for BSN applications.

The wearable nature of BSNs leads to multiple challenges when developing mobile applications in the sports domain. Often, these challenges cannot be assigned to the technology or the application side but have to be tackled simultaneously from both perspectives. One example is a feedback display that is restricted by power consumption or price (technology) and restricted by optimal feedback presentation and real-time capability (application). Practical solutions require that both, technology and application, are considered in the design of BSN applications.

This following chapter focusses on the data processing and analysis at the above mentioned interface of technology and application. The technological focus will be on the design and implementation of classification algorithms for low-power microcontrollers. Two underlying use cases were considered. First, technology is predefined and data analysis algorithms have to match these technical requirements. Second, data analysis algorithms are defined and technology matching the algorithm demand needs to be identified. This work provides a practical methodology to tackle these use cases for embedded classification (chapter 5). Examples of different BSN applications that use these design considerations are presented in chapters 6 to 8.

Chapter 5

Design Considerations for Embedded Classification Systems

The previous chapters introduced fundamentals of pattern recognition, mobile sensing and computing as well as sports science. A specific application and the combination of these fields, embedded classification of inertial sensor data for athlete support, will from here on be considered in more detail. This chapter presents design considerations to develop such embedded classification systems and starts with an overview of the embedded classification development workflow. The remainder is organized in two parts.

The first part of this chapter uses the development overview to describe three design considerations in more detail. These are a general methodology to tackle the accuracy-cost tradeoff, best practices to estimate classifier accuracy with feature selection and considerations regarding energy-efficient BSN signal classification. The application in chapter 8 considered these three aspects in a swimming application.

The second part of this chapter focuses on sequential data analysis. More specifically, best practices for HMM modeling for signal analysis are described. The application presented in chapter 6 considered an accurate modeling of HMMs for a plyometric training application.

The methodology to tackle the accuracy-cost tradeoff is the most comprehensive consideration described in this chapter. It will be laid out in detail as it is relevant for a wide range of applications and handles its underlying challenge in a generic approach. The remaining considerations examine more specific topics and are therefore described briefly. Their key aspects, however, can also be transferred to other applications and problems.

This chapter is organized as follows: first, the development workflow of embedded classification systems in presented. Second, a methodology to support the solution of the accuracy-cost tradeoff is shown. Third, the procedure of classifier accuracy estimation with feature selection is reviewed and best practices are deduced. Fourth, aspects of energy-efficient BSN signal classification are summarized. Finally, accurate and thus powerful HMM modeling for the analysis of BSN signals is described.

5.1 Development Workflow

This section introduces a development workflow to design embedded classification systems (Fig. 5.1). The workflow starts with a data collection and labeling step. In this step, data for training the classification system are collected. The outcome of this step is a sufficient training sample.

The second step is the algorithm design phase to determine the system with the highest accuracy. According to the No Free Lunch theorem, this is an iterative process of comparing different classification pipeline settings. The algorithm design phase (also: offline phase) is often performed on high-performance multi-purpose computers using prototyping tools like MATLAB. The outcome of this phase is the selected classification algorithm.

After having determined the target algorithm in the design phase, this system has to be implemented, tested and optimized. Therefore, the classification working phase (also: online phase) has to be implemented on target hardware in the final step. This implementation includes a trained classifier. The outcome of this phase is a source code implementation for embedded hardware.

A crucial step in this workflow is the algorithm selection before implementation. Two use cases illustrate this statement. In the first example, use case A, the target hardware is not predefined. It has to be selected according to the classification system that reached the desired accuracy during design. In the second example, use case B, the target hardware is predefined. Constraints might be the price, size or availability.

Therefore, the task is to identify the most accurate classification system that is capable of running on the predefined hardware. An economic choice of hardware is especially important for wearable systems as they are constrained in weight, form factor and batterie capacity.

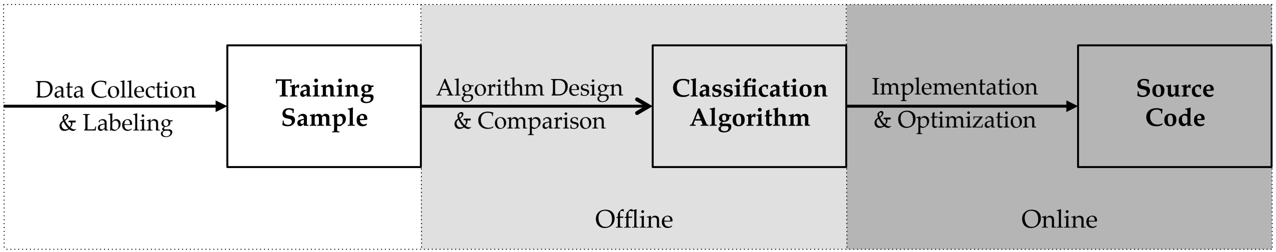

Figure 5.1: Embedded classification workflow example consisting of data collection, algorithm design and implementation.

These use cases illustrate the need for an embedded classification design tool. The following prerequisites were identified: first, as hardware might not be selected yet, the analysis level has to be target independent. Second, the measures that the design decision is based on have to be practical in order to support the implementation decision. Third, the methodology has to be applicable to all algorithms that are compared in the design phase. Fourth, working phase cost estimation is of major interest as the training of the system can be performed offline and only the working phase algorithms have to be implemented on the target hardware. The literature research presented in chapter 1.2.1 revealed that such a methodology does currently not exist and will therefore be presented in chapter 5.2.

Additional aspects require consideration in the algorithm design phase. For decision making in the latter, precise estimates of the algorithm accuracies are required. A designer of such a system is mainly interested in the expected accuracy when the system classifies previously unseen data. For decision making in the design phase, the designer of such a system needs to have an understanding of how specific processing steps influence the energy consumption of the final system. These two aspects are considered in chapters 5.3 and 5.4.

5.2 Tackling the Accuracy-cost Tradeoff

The aspect described in this chapter considers the accuracy-cost tradeoff in embedded classification system design. The literature review in chapter 1.2.1 revealed that classifier accuracy and computational cost are two competing goals in the design of embedded classification systems. However, no general approach to support the solution to this accuracy-cost tradeoff exists. This work describes a methodology that determines computational cost and memory demand in the embedded classification design phase. It can be used to support decisions in the system design phase and is the most important aspect in this chapter. It will therefore be described extensively.

First, the analysis methodology is presented in chapters 5.2.1 to 5.2.5. Second, a software package implementing the methodology is described in chapter 5.2.6 and appendix A. The content of this section was published in similar form in [Jens 16b].

5.2.1 Classification Cost Measures

Appropriate cost measures are required to assess the working phase cost of an embedded classification system. The measures we propose are chosen according to [Duda 01]. There, the number of basic mathematical operations and the memory needed on a computer were identified as suitable computational cost measures.

As basic mathematical operations, arithmetic operations ($+, -, \times, \div$), comparisons ($<, >, \leq, \geq$) and function evaluations ($\sqrt{x}, e^x$) were considered. We grouped additions and subtractions as well as all types of comparisons for simplification and assumed that all operations are performed using the same data type (e.g. float).

The number and type of mathematical operations can be identified in all considered pattern recognition algorithms. These results can be used for a hardware-dependent analysis, the estimation of the number of machine cycles for a specific microcontroller. This additional step is presented in [Jens 16b] and beyond the scope of this thesis.

As memory measures, we considered the number and type of precomputed constants required for the final classification system. We categorized them in integer (INT) and floating-point (FLT) constants. These constants can be stored in read-only memory. We omitted main memory consumption for intermediate variables.

The computational cost and the memory demand need to be estimated for the working phase of the classification pipeline (Fig. 2.1). Algorithms of the feature extraction, feature preprocessing and classification step were considered and the nomenclature introduced in chapter 2 was used. An overview of the deduced cost measures is compiled at the end of the chapter (Tab. 5.1).

The reference implementation for the algorithms was the apache common math library [Comm 13] for the feature extraction algorithms and the weka data mining software library [Hall 09, Witt 11] for the feature preprocessing algorithms as well as all classifiers except support vector machine (SVM). The reference implementation for the SVM was the libSVM library [Chan 11].

For many algorithms, different alternatives for implementation are available. Comparing all implementations for the same algorithm is beyond the scope of this thesis. Furthermore, some alternatives are related to specific hardware architectures. We used the above-mentioned reference implementations and simplified the algorithms as described.

5.2.2 Feature Extraction

The implemented feature set consists of heuristic features in time domain. As features are often application dependent, we chose a set of generic features so that this set can be applied to a wide range of classification problems. The presented feature set was partly described in [Ciac 93] and showed encouraging results in a pilot study where BSN signals were analyzed [Jens 12b].

The feature extraction of the presented methodology uses the popular sliding window technique of analyzing sequential data [Diet 02]. The feature extraction step transformed the input h_t^s to a feature vector $\mathbf{x}_i$ (Fig. 2.1). Our analysis assumed a fixed pattern length T (data space). This constant (INT) has to be saved on the target system.

We assumed that only one sensor dimension contributed to a feature value. The signal energy will be used as example to clarify this assumption. When processing a 6-D data window, the signal energy can be computed from six different axes. If more sensor dimensions contributed to the same feature (e.g. all dimensions), less features could be extracted for one window. Thus, all features introduced in this chapter are extracted using one single axis.

We further assumed that each feature value was extracted independently. The standard deviation will be used as an example to clarify this assumption. The standard deviation is computed from the square root of the variance. Our computational cost estimation assumes that standard deviation and variance are computed in two different steps. We chose this setup as feature selection algorithm might select these mathematically connected features. Thus, the computational cost of each feature is treated independently although mathematical operations might be saved due to intermediate results.

We also assumed that the pattern length T (data space) has to be saved for each feature value and that therefore the memory demand for each feature was one constant (INT). The memory demand and the computational cost of the feature extraction are summarized in Tab. 5.1.

Signal Energy

The signal energy is the sum of absolute values in an interval of length T (Eq. 5.1).

$$E = \sum_{t=1}^{T} |h_t^s| \tag{5.1}$$

The computational cost for E was straightforward and determined with $T-1$ additions. We omitted the absolute value operation as it can be implemented with computationally cheap shifting operations.

Minimum

The minimum feature was computed according to Alg. 1. The computational cost for min was determined with $T-1$ comparisons.

Algorithm 1: Computes the minimum and the maximum.

$min \leftarrow h^s_{t_1}$
$max \leftarrow h^s_{t_1}$
for $t = 2$ **to** $t = T$ **do**
    if $h^s_t < min$ **then**
        $min \leftarrow h^s_t$
    end
    if $h^s_t > max$ **then**
        $max \leftarrow h^s_t$
    end
end

Maximum

The maximum feature was computed according to Alg. 1. The computational cost for max was determined with $T-1$ comparisons.

Mean Value

The mean value computation (Eq. 5.2) consisted of additions to sum all values of an interval and a final division by the interval size.

$$M = \frac{1}{T}\sum_{t=1}^{T} h^s_t \tag{5.2}$$

The computational cost for M was $T-1$ additions and a single division. The division could as well be mapped to a multiplication as $\frac{1}{T}$ can be precomputed and saved on the target system as an additional constant.

Variance

The variance computation was based on the mean calculation and therefore included the cost of M. Our analysis used the unbiased variance computation with the square operation mapped to a multiplication (Eq. 5.3).

$$\sigma^2 = \frac{1}{T-1}\sum_{t=1}^{T}(h_t^s - M)(h_t^s - M) \tag{5.3}$$

The computational cost for σ^2 was the cost for M and additional cost. The additional cost was T multiplications and T subtractions in the sum. Additionally, the summation was $T-1$ additions. For the final result, one division and one subtraction were also needed for the term outside the sum.

Standard Deviation

The standard deviation can be computed from the variance σ^2 (Eq. 5.4). Thereby, the cost for the variance and the mean contribute to the computational cost of the standard deviation.

$$\sigma = \sqrt{\sigma^2} \tag{5.4}$$

The computational cost for σ was the cost for σ^2 as well as one square root operation.

Skewness

The skewness was computed using standard deviation, variance and mean. Based on the unbiased version of the skewness (Eq. 5.5), we adapted the expression so that less computations are performed inside the sum and that exponential operations were mapped to multiplications.

$$V_{\text{basic}} = \frac{T}{(T-1)(T-2)}\sum_{t=1}^{T}\left(\frac{h_t^s - M}{\sigma}\right)^3 \tag{5.5}$$

In the simplified version (Eq. 5.6), the denominator was removed from the repetitive computation in the sum and the power operation was mapped to multiplications.

$$V = \frac{T}{(T-1)(T-2)\sigma\sigma^2}\sum_{t=1}^{T}(h_t^s - M)(h_t^s - M)(h_t^s - M) \tag{5.6}$$

The computational cost for V consisted of the cost for the standard deviation and its intermediate results (σ^2, M) that were available without additional cost. Additionally, T subtractions and $T-1$ additions as well as $2T$ multiplications were required in the sum. Outside the sum, two subtractions, four multiplications (denominator and sum) and one division were required.

Kurtosis

The kurtosis was computed using the variance and the mean. Based on the unbiased version of the kurtosis (Eq. 5.7), we adapted the expression so that less computations are performed inside the sum and that exponential operations were mapped to multiplications.

$$W_{\text{basic}} = \left[\frac{T(T+1)}{(T-1)(T-2)(T-3)} \sum_{t=1}^{T} \left(\frac{h_t^s - M}{\sigma} \right)^4 \right] - \frac{3(T-1)^2}{(T-2)(T-3)} \tag{5.7}$$

In the simplified version (Eq. 5.8), the denominator was removed from the repetitive computation in the sum and the power operations were mapped to multiplications.

$$W = \left[\frac{T(T+1)}{(T-1)(T-2)(T-3)\sigma^2\sigma^2} \sum_{t=1}^{T} (h_t^s - M)(h_t^s - M)(h_t^s - M)(h_t^s - M) \right] - \frac{3(T-1)(T-1)}{(T-2)(T-3)} \tag{5.8}$$

The computational cost for W consisted of the cost for the variance and its intermediate result M that was available without additional cost. Additionally, T subtractions and $T-1$ additions as well as $3T$ multiplications were performed in the sum. Outside the sum, five additions/subtractions, nine multiplications and two divisions were required.

5.2.3 Feature Preprocessing

Feature preprocessing is a transformation step to create more suitable patterns for classification. This work focuses on two aspects, the removal of erroneous sensor recordings and the normalization of feature value ranges. The first aspect can be tackled with outlier removal and the second aspect with data normalization. Feature preprocessing transforms a vector $\mathbf{x}_i$ to a vector $\mathbf{x}_i$ (Fig. 2.1). The memory demand and the computational cost of the feature preprocessing are summarized in Tab. 5.1.

Outlier Removal

Sensor malfunction or errors in the measurement process can lead to patterns that are inconsistent with the remainder of the data. It is often desired to remove them from further processing. Even when no error occurred, recordings that occur very infrequently might bias the underlying data distribution and a more realistic sample can be achieved by removing these data [Barn 87]. Our analysis considers outlier removal based on quartiles [Laur 00].

A quartile p divides the data so that $p \times 100\%$ of the values are smaller and $(1-p) \times 100\%$ of the values are bigger than the quartile value. In the training phase, the presented outlier removal algorithm determined the 0.25-quartile $q^d_{0.25}$ and 0.75-quartile $q^d_{0.75}$ for each dimension D.

The interquartile range iqr^d was computed for further processing:

$$iqr^d = q^d_{0.75} - q^d_{0.25} \tag{5.9}$$

In the working phase, a feature vector $\mathbf{x}_i$ was further processed if the following equation was fulfilled in all dimensions and otherwise declared as outlier:

$$q^d_{0.25} - f_{\text{out}} \times iqr^d \leq \hat{x}^d_i \leq q^d_{0.75} + f_{\text{out}} \times iqr^d \tag{5.10}$$

Thereby, the factor f_{out} sets the sensitivity of the removal according to the interquartile range and is determined in training phase. Thus, in worst case, all feature dimensions have to be checked to decide whether the pattern is an outlier or not. Two bounds for each dimension d have to be saved on the target system to detect outliers in the working phase :

$$lower^d = q^d_{0.25} - f_{\text{out}} \times iqr^d \tag{5.11}$$

$$upper^d = q^d_{0.75} + f_{\text{out}} \times iqr^d \tag{5.12}$$

The computational cost for the described outlier removal was $2D$ comparisons. The memory demand was $2D$ FLT parameters for the two bounds of each dimension.

Normalization

Normalization maps numeric features into the same value range [Niem 90] and therefore equalizes the influence of all features. This is especially important for classifiers that use distance computations and also avoids numerical problems like data overflow. Our analysis considered a feature-wise normalization.

In the training phase, the extremum values min^d and max^d for each feature dimension d have to be determined. Additionally, two factors have to be set, the scale a and the translation b. These factors represent the desired value range that is identical for all features. The values $a = 2$ and $b = 1$ for example would normalize the features to the range of $[-1, 1]$.

Using these values, a feature vector $\hat{\mathbf{x}}_i$ was normalized to $\mathbf{x}_i$ in the working phase using the individual min^d and max^d values of each dimension (Eq. 5.13).

$$x_i^d = a \frac{\hat{x}_i^d - min^d}{max^d - min^d} + b \tag{5.13}$$

The computational cost deduced from Eq. 5.13 was $3D$ additions/subtractions, D divisions and D multiplications. The memory demand was $2D$ FLT parameters for the min^d and max^d values and two FLT parameters for a and b that were the same for all dimensions.

5.2.4 Classification

Classification is the final step in the processing pipeline. In this step, a pattern is categorized in one of the predefined classes. Thus, a preprocessed feature vector $\mathbf{x}_i$ is the input for a decision function f($\mathbf{x}_i$) whose output is a class index (Fig. 2.1).

We selected eight classification algorithms for our analysis. These algorithm represent different classification paradigms and are among the most popular pattern classification algorithms. In each paragraph, a short description of the classifier paradigm will be given before the decision making is considered in more detail. The description also contains a list of parameters that were either free or determined during training.

The general problem-dependent parameters of all classifiers were:

- C: Number of classes
- D: Dimensionality of the data

The memory demand and the computational cost of the classification are summarized in Tab. 5.1.

AdaBoost (AB)

The paradigm of AdaBoost [Freu 97] is to combine several simple (weak) classifiers to a single complex (strong) classifier. This procedure is called boosting [Scha 90]. During training, the iterative classifier construction is steered by previous misclassifications that result from the simple nature of the weak classifiers. Polikar [Poli 07] gave an overview of the topic and potential application fields of AdaBoost beyond pattern classification which are incremental learning, data fusion and missing feature analysis.

The analysis presented in this work considered decision stumps as weak classifiers. Decision stumps consist of a decision rule (Eq. 5.14) and an associated weight w as well as class indices c_{false} and c_{true}.

$$x_i^d \leq k \tag{5.14}$$

Thereby, x_i^d denoted the value of feature dimension d of pattern i and k a threshold value. According to the result of the decision rule evaluation, class index c_{false} (false) or c_{true} (true) are incremented with weight w.

For each decision stump, the feature index and the threshold had to be known to evaluate the decision rule. Additionally, the class indices that corresponded to the result (true, false) need to be known. The weak classifier decisions were then weighted in their contribution to the final classification. Thus, each rule has a corresponding weight that reflects its importance. After evaluating the weak classifiers, a score was accumulated for each class and the class with the highest score was selected.

The parameters of this classifier were:

- B: Number of weak classifiers (or iterations)

The computational cost was one comparison for the decision stump evaluation and one addition to update the class weight sum for each of the B decision stumps. For the final class decision, the maximum class score was determined with $C-1$ comparisons. The memory demand was three INT parameters for the feature and class indices and two FLT parameters for the threshold and the weight for each weak classifier.

Artificial Neural Network (ANN)

The idea of this classifier originated in analogy with the structure of the human brain. Our analysis considered a multilayer perceptron. According to [Duda 01], its structure is organized as a connected graph that is arranged in several layers (Fig. 5.2). The input layer represents the feature vector while the output layer represents the classes. One or more hidden layers with intermediate nodes n_l with $l = 1, \ldots, L$ connect the input and output nodes. During training, the weights of the links (w_i with $i = 1, \ldots, W$) connecting the nodes are adjusted.

The input layer nodes forwarded the feature value and did not perform any computation. The remaining nodes, those in the hidden layers and output layer, performed a two step computation. First, a linear model of the input was evaluated:

$$\mathrm{f}(\mathbf{x}_i) = \mathbf{w}^{\mathrm{T}} \mathbf{x}_i + w_0 \tag{5.15}$$

The vector $\mathbf{w}$ consisted of the link weights and $\mathbf{x}_i$ consisted of the output values of the connected nodes. These were the feature values in the first hidden layer. The parameter w_0 was a constant determined during training. Second, the result of the linear model evaluation was fed into a sigmoid function to compute the node output $\mathrm{g}(\mathbf{x}_i)$ (Eq. 5.16).

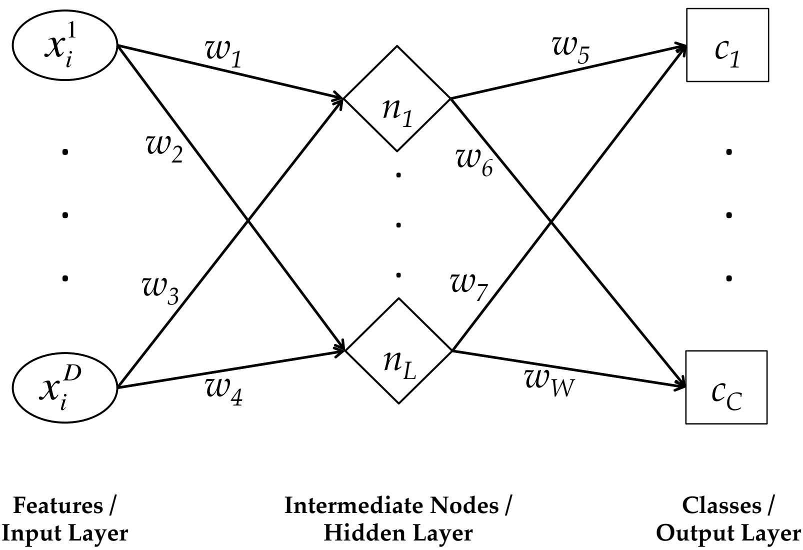

Figure 5.2: Structure of a multilayer perceptron with three layers featuring D features, L intermediate nodes, W links and C classes. Features are denoted by $x_i^1, \ldots, x_i^D$, hidden layer nodes with $n_1, \ldots, n_L$, links with $w_1, \ldots, w_W$ and classes with $c_1, \ldots, c_C$ (Fig. 2.1).

$$g(\mathbf{x}_i) = \frac{1}{1 + e^{-f(\mathbf{x}_i)}} \tag{5.16}$$

The decision rule was to choose the class whose output node computed the maximum value.

The parameters of this classifier were:

- W: Overall number of links
- $H = L + C$: Number of intermediate and output nodes

The computational cost comprised of linear model evaluation (Eq. 5.15) and sigmoid function evaluation (Eq. 5.16) of each node.

The linear model evaluation is considered first. Each link in W contributed to the computation in the target node. Thus, overall W multiplications were performed in the dot products of all nodes. Additionally, each cross product computation comprised $W-1$ additions and the weight addition required one addition for the constant w_0. Overall, W multiplications and W additions for the linear model evaluation were needed.

The sigmoid function evaluation comprised H additions, H divisions and H exponential function evaluations. For the class decision, $C-1$ comparisons were used to determine the output node with the maximum value.

The memory demand was W FLT parameters for each link weight and H FLT parameters for the w_0 constant.

C4.5

This classifier discriminates classes using a decision tree. Tree-based classifiers mainly differ in the way they build up and prune the decision tree. Important aspects are the node type (binary, multiple), the criterion to choose and stop splitting nodes and the pruning of subtrees [Duda 01]. These aspects influence the training complexity, classification performance and generalization capabilities. Our analysis considered the C4.5 algorithm [Quin 93].

In the working phase, the learned decision tree was descended from root to leaf for decision making (Fig. 5.3). Each leaf contained a class index c_i with $i = 1, \ldots, L$) which represented the final classification decision. Each inner node n_i with $i = 1, \ldots, I$ contained a decision rule like Eq. 5.14 and two node indices that determined how to proceed in descending the tree. The two indices, r_0 (false) and r_1 (true), determined the next node according to the rule result.

Depending on the tree structure, the paths from root to leaf had different lengths. We decided to provide an analysis of the longest possible path according to the worst case execution time often used in embedded system design. Our analysis therefore considered the cost of descending from root to leaf on the longest path (computational cost) and the size of the trained tree (memory demand).

The parameters of this classifier were:

- U: Number of inner nodes on the longest path through the tree
- I: Number of inner nodes
- L: Number of leaves

The computational cost was U comparisons to descend the tree from root to node. The memory demand was three INT parameters and one

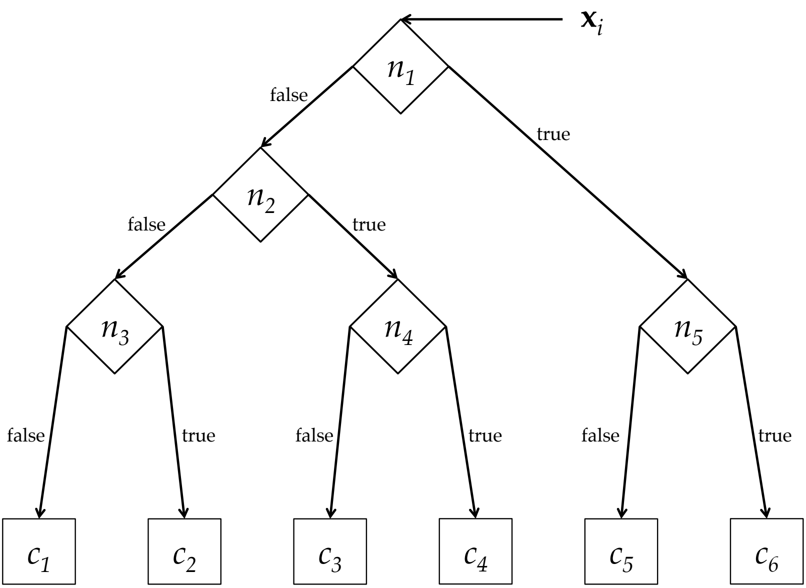

Figure 5.3: Example of a classification tree for classification with feature vector $\mathbf{x}_i$, five inner nodes (n_1 to n_5), six leaves (c_1 to c_6) and a longest path of three nodes (e.g. n_1, n_2, n_3).

FLT parameter for each inner node I. These parameters were needed to define the threshold, the feature index and r_0 & r_1. Each leaf L was represented by a single INT parameter, the class index.

Classification via logistic regression (CLR)

The idea of this classifier is to create a linear regression model [Hast 08] for each class in the training phase and decide for the class that corresponds to the best fitting model in the working phase. The akaike information criterion [Akai 73] was used to create the individual class models $\mathrm{f}_c(\mathbf{x}_i)$ in the training phase (Eq. 5.15).

Two steps were required during working phase; the evaluation of the individual class models and the transformation of the model results to class probabilities. The computations in the model evaluation can be deduced from Eq. 5.15 and were conducted for each class. The transformation to class probabilities was performed in two steps [Hall 09]. First, values bigger than one were set to one and negative values were set to zero. Second, all values were scaled to sum up to one. The class with the highest probability (the best fitting model) was chosen as classification decision.

The computational cost was CD multiplications and CD additions for the class model evaluations that consisted of a dot product and the addition of the constant. Additionally, the transformation used $2C$ comparisons, $C-1$ additions and C divisions. The cost for the final class decision was $C-1$ comparisons. The memory demand was $C(D+1)$ FLT parameters for the linear class models.

Naive Bayes (NB)

This classification paradigm is based on the statistical properties of the feature distributions and decides for the class with the highest posterior probability $P(c|\mathbf{x}_i)$ according to the bayes rule (see chapter 2.2.5).

In the naive bayes approach as it is explained in [Theo 09], the class-conditional pdf is assumed to be normally distributed. Furthermore, it is estimated using the product of feature pdfs instead of an estimation of one high-dimensional pdf:

$$P(c|\mathbf{x}_i) = P(c)\prod_{d=1}^{D} p(x_i^d|c). \tag{5.17}$$

Thereby, it is assumed that the features are conditionally independent. For each class and dimension, one normally distributed pdf had to be estimated with:

$$p(x_i^d|c) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(\mathbf{x}-\mu)^2}{2\sigma^2}} \tag{5.18}$$

Thereby, the mean is denoted by μ and the standard deviation is denoted by σ. Together with the prior class probability $P(c)$, the highest posterior probability $P(c|\mathbf{x}_i)$ among all classes was determined and the corresponding class was selected.

The computational cost for Eq. 5.18 was one square root, one exponential function evaluation, one subtraction and two divisions for each dimension of each class. Two simplifications were performed. First, we mapped the squaring operation to a multiplication. Second, instead of saving π and σ as constants, we precomputed 2π and σ^2 and saved these values as constant. Overall, five multiplications were determined for each class. Additionally, D multiplications for each class in Eq. 5.17 and $C-1$ comparisons to determine the maximum posterior probability were determined.

The memory demand was C FLT parameters for the prior class probabilities and $2CD$ FLT parameters for the pdf means and standard deviations of each class C and dimension D.

Nearest Neighbor (NN)

The nearest neighbor classifier is a non-parametric technique as the class-conditional pdf is unknown and not estimated. No classification model is learned during training. In the working phase, an unknown pattern is assigned to the class of the closest neighboring training pattern [Duda 01]. This simple concept can be enhanced by considering the k nearest neighbors and choosing the class that occurs most often.

Our analysis considered a version with k nearest neighbors and the euclidean distance measure. We assumed that the complete training set was available for the working phase. To classify a pattern, the k nearest neighbors had to be found within the complete training set. Thus, the euclidean distance between an unknown pattern $\mathbf{x}_i$ and each training pattern $\mathbf{t}_p$ with $p = 1, \ldots, P$ was computed (Eq. 5.19). Thereby, P denoted the number of training patterns.

$$\left\|\mathbf{x}_i - \mathbf{t}_p\right\| = \sqrt{\sum_{d=1}^{D}(x_i^d - t_p^d)(x_i^d - t_p^d)} \tag{5.19}$$

During working phase, a list of the k shortest distances had to be maintained. Therefore, this list was first filled with the first k distances and updated subsequently. For the $P-k$ remaining patterns, a worst case of k comparisons was performed in the update step. After distance computation, the class occurrence was determined. This required k additions and a final search along all classes that required $C-1$ comparisons.

The parameters of this classifier were:

- P: Number of training patterns
- k: Number of neighbors

The computational cost was PD subtractions, $P(D-1)$ additions, PD multiplications and P square root operations for the distance computations. Additionally, $(P-k)k$ comparisons were performed to update the list of distances and k additions as well as $C-1$ comparisons for the final class decision.

The memory demand was PD FLT parameters to save the training patterns and P INT parameters to save the class label. Additionally, one INT parameter (k) had to be saved.

PART

This algorithm belongs to the class of tree-based classifiers. Based on a decision tree, these classifiers often create a set of rules that represent a simplified and transformed version of the original tree. Compared to the algorithms C4.5 and RIPPER, PART differs in the way this rule set is generated and avoids global optimization [Fran 98]. PART operates in a repeated divide-and-conquer approach where only parts of the training data are considered. The specific difference of PART is that the algorithm creates partial decision trees that contain branches to undefined subtrees. These trees avoid extensive creation and pruning operations like they are performed in alternative algorithms. Furthermore, these partial decision trees are optimized for rule deduction.

PART outputs a set of decision rules as classifier model. The rules consist of one or more comparisons against thresholds and a class label. Multiple comparisons are concatenated with logical operands. On positive evaluation of a rule, the pattern is classified according to the class label of this rule. Our analysis considered the worst case where all rules and, within all rules, all comparisons had to be considered to classify an unknown pattern.

The parameters of this classifier were:

- R: Number of rules
- Q: Number of all comparisons in all rules

The computational cost was Q comparisons when evaluating all rules. The memory demand for the comparisons was $2Q$ INT parameters for the feature index and the type of comparison which were both encoded in an INT parameter. Additionally, Q FLT parameters for the thresholds and R class indices were determined.

Support Vector Machine (SVM)

SVM is a powerful and widely-used classification method [Vapn 95, Scho 01]. In the training phase, an affine function is estimated that generates the maximum margin of class separation:

$$\mathrm{f}(\mathbf{x}_\mathrm{i}) = \mathrm{sgn}\left(\sum_{j=1}^{SV} y_j \alpha_j \,\mathrm{K}(\mathbf{x}_j, \mathbf{x}_i) + \alpha_0\right) \tag{5.20}$$

Thereby, SV denotes the number of support vectors, y_j the class label and α_j the weight of support vector $\mathbf{x}_j$. The variable a_0 is a constant and $\mathrm{K}(\mathbf{x}_j, \mathbf{x}_i)$ the kernel function of the support vector and the input pattern.

SVM is basically a linear classifier. However, the kernel trick enables the classification of non-linearly separable data with linear methods. As patterns are represented as dot products in the algorithm, kernel functions can be applied to transform the patterns to a higher dimensional space. The target space is chosen so that the data is linearly separable.The SVM framework contains extensions for the case that data is not linearly separable. So called slack variables are introduced to allow misclassifications in training phase.

Our analysis considered a simplified linear SVM

$$\mathrm{f}_\mathrm{lin}(\mathbf{x}_i) = \mathrm{sgn}(\boldsymbol{\beta}^\mathrm{T}\mathbf{x}_i + \alpha_0) \tag{5.21}$$

and kernel SVMs (Eq. 5.20) with polynomial kernel

$$\mathrm{K}(\mathbf{x}_j, \mathbf{x}_i) = (\gamma_0 \mathbf{x}_j^\mathrm{T}\mathbf{x}_i + \gamma_1)^{\gamma_2} \tag{5.22}$$

as well as radial basis function kernel

$$\mathrm{K}(\mathbf{x}_j, \mathbf{x}_i) = e^{-\gamma\|\mathbf{x}_j - \mathbf{x}_i\|^2}. \tag{5.23}$$

For the computation of the γ_2-th power, the square-and-multiply technique [Knut 98] that requires $2\lfloor \log_2 \gamma_2 \rfloor$ multiplications was used.

The generic multi-class case was analyzed. There, the class with most individual votes in the one on one decisions is selected as classification decision [Kres 98].

The parameters of this classifier were:

- N: Number of support vectors in all SVMs
- S: Number of SVMs with $S = C(C-1)/2$ (multi class) and $S = 1$ (two classes)

The computational cost and memory demand are split for the sake of readability. The overview provides the cost for a classification with either a simplified linear kernel $\mathrm{f_{lin}}(\mathbf{x}_i)$ (Eg. 5.21) or kernel SVM. The cost measures for the kernel SVM consisted of the basic measures (Eq. 5.20) plus the measures for the corresponding kernel.

Linear SVM The computational cost according to Eq. 5.21 was D multiplications and D additions and one comparison (sgn) for each of the S SVMs. Additionally, the class occurrence was updated with S additions and the final class decision was determined with $C-1$ comparisons. The memory demand was D+1 FLT parameters for β and α_0 for each of the S SVMs.

Kernel SVM The computational cost according to Eq. 5.20 was $2N$ multiplications, N additions and S comparisons (sgn). Additionally, the class occurrence was updated with S additions and the final class decision was determined with $C-1$ comparisons. The memory demand was ND FLT parameters for the support vectors $\mathbf{x}_j$, N FLT parameters for the the weights α_j, N INT parameters for the class labels y_i and S INT parameters for the constant α_0 for each SVM.

Polynomial kernel The computational cost according to Eq. 5.22 together with the simplified power computation was $D+1+2\lfloor \log_2 \gamma_2 \rfloor$ multiplications and D additions for each of the N support vectors. The memory demand was three FLT parameter for γ_0, γ_1 and γ_2.

Rbf kernel The computational cost according to Eq. 5.23 was $D+1$ multiplications, $2D-1$ additions/subtractions and one exponential function evaluation for each of the N support vectors. The memory demand was one FLT parameter (γ).

5.2.5 Numeric Cost Measures

The previous sections introduced a generic algorithm cost analysis subject to variables. To directly compare algorithms, these variables have to be resolved to specific numerical cost measures. Due to the above-mentioned No Free Lunch theorem, algorithms have to be compared to find fitting solutions. The following paragraphs describe how numerical cost values are deduced in the presented methodology and how algorithm accuracy might be influenced by these variables.

The presented methodology tackles the accuracy-cost tradeoff with the simultaneous consideration of classification accuracy and computational cost. The analysis in chapters 5.2.2 to 5.2.4 revealed that the proposed cost measures depend on different variables. We also identified factors and variables that affect the accuracy estimation. The variables and factors that influence the estimated accuracy and the cost measures can be grouped by:

- Data and problem definition
  - T: Interval size
  - P: Number of training patterns
  - C: Number of classes
  - ...
- Classifier training output
  - W: Number of overall links (ANN)
  - U: Number of inner nodes on the longest path through the tree (C4.5)
  - N: Overall number of support vectors (SVM)
  - ...
- Free parameters
  - f_{out}: Sensitivity factor in outlier removal
  - B: Number of iterations for classifier AB
  - k: Number of neighbors in classifier NN
  - ...

Algorithm	**Computational Cost**						**Memory Demand**	
	$+,-$	$\times$	$\div$	$<$	$\sqrt{x}$	e^x	INT	FLT
Signal Energy	$T-1$	–	–	–	–	–	1	–
Minimum	–	–	–	$T-1$	–	–	1	–
Maximum	–	–	–	$T-1$	–	–	1	–
Mean	$T-1$	–	1	–	–	–	1	–
Variance	$3T-1$	T	2	–	–	–	1	–
Standard Deviation	$3T-1$	T	2	–	1	–	1	–
Skewness	$5T$	$2T+4$	3	–	1	–	1	–
Kurtosis	$5T+3$	$3T+9$	4	–	–	–	1	–
Outlier Removal	–	–	–	$2D$	–	–	–	$2D$
Normalization	$3D$	D	D	–	–	–	–	$2D+2$
AdaBoost (AB)	B	–	–	$B+C-1$	–	–	$3B$	$2B$
Artificial Neural Network (ANN)	$H+W$	W	H	$C-1$	–	H	–	$W+H$
C4.5	–	–	–	U	–	–	$3I+3L$	I
Classification via log. regression	$CD+C-1$	CD	C	$2C+C-1$	–	–	–	$C(D+1)$
Naive Bayes	CD	$6CD$	$2CD$	$C-1$	CD	CD	–	$C+2CD$
Nearest Neighbor (NN)	$PD+P(D-1)+k$	DP	–	$k(P-k)+(C-1)$	P	–	$P+1$	PD
PART	–	–	–	Q	–	–	$2Q+R$	Q
Linear SVM	$SD+S$	SD	–	$S+C-1$	–	–	–	$S(D+1)$
Kernel SVM	$S+N$	$2N$	–	$S+C-1$	–	–	N	$S+N+ND$
+ polynomial kernel	DN	$(D+1+2\lfloor\log_2\gamma_2\rfloor)N$	–	–	–	–	–	3
+ radial basis kernel	$(2D-1)N$	$(D+1)N$	–	–	–	N	–	1

For feature computation, the variable T denotes the length of the segmented input signal. Note that the cost is only given for the computation of the feature along one single feature dimension. For the preprocessing and classification algorithms, the variable D denotes the dimensionality of the data, i.e. the number of used features after feature selection. The variable C denotes the number of classes. The remaining variables are algorithm dependent: For AB, B is the number of weak classifiers. For ANN, W is the overall number of links and H is the sum of intermediate and output nodes. For C4.5, U represents the number of inner nodes on the longest path through the tree, L is the number of leaves nodes and I is the number of inner nodes. For NN, k is the number of neighbors considered, while P is the number of training patterns. For PART, the variable R represents the number of rules, while Q is the number of all comparisons in all rules. In the SVM case, N is the number of support vectors in all SVMs, while S is the number of trained SVMs. For the kernel SVM, the total cost consists of the SVM costs plus the costs of the selected kernel.

Table 5.1: Cost overview for the presented working phase algorithms.

- Algorithm choice
  - Evaluation method: Training-testing, cross-validation (see chapter 5.2.6)
  - Dimensionality reduction through feature selection
  - Performance gain through feature preprocessing
  - ...

The variables from the categories “data and problem definition” and “classifier training output” are automatically resolved for a specific data sample and analysis. Thus, classifier training results in specific values for these parameters so that numerical measures can directly be deduced.

The variables from the categories “free parameters” and “algorithm choice” depend on user choice and configurations. They can either be set on specific problem domain knowledge and experience or evaluated in a broad experimental setup. The latter brute force approach on user configuration parameters coincides with the brute force approach on classifier selection according to the above-mentioned No Free Lunch theorem. The next sections introduce a software package that allows an extensive comparison of different classification pipeline configuration including different classifiers and algorithm parameters. It implements the methodology described above.

5.2.6 Embedded Classification Software Toolbox (ECST)

The embedded classification software toolbox (ECST) was specifically developed to provide a support tool to tackle the accuracy-cost tradeoff according to the above-mentioned methodology. It provided a toolbox of common classification algorithms and implemented the practical cost measures that were introduced in the previous sections. These cost measures reflect the computational cost and memory demand of the classifier in the working phase. However, as training phase algorithms heavily influence the accuracy and the cost of a classifier model, these algorithms had to be integrated in such a classification system design tool. See also appendix A for ECST user interface details and explanations.

Training Phase Algorithms

The ECST contained several training phase algorithms for feature selection and classifier evaluation. These algorithms were not part of the cost analysis, but influenced the resulting cost analysis and accuracy estimation.

A reduced feature set e.g. changes the dimensionality of the classification problem and therefore influences the cost measures. Furthermore, feature selection is often conducted to reach higher classification accuracy.

Three feature selection algorithms were part of the ECST: best first, branch-and-bound and weighted wearch. **Best first search** as described in [Theo 09] starts with an empty (complete) feature set and sequentially adds (removes) a feature for the highest accuracy improvement. A backtracking functionality allows to consider alternatives in the search process. The ECST used the implementation from [Hall 09]. **Branch-and-bound search** determines the optimal feature set for a given subset size. However, it does not exhaustively evaluate all feature sets. The ECST contained the native implementation of the branch-and-bound algorithm from [Chen 03] that used the Bhattacharyya distance as monotonic distance measure. **Weighted search** is an algorithm for cost aware feature selection in the context of embedded classification. It was not published previously and was specifically developed in the ECST to support the presented methodology. It simultaneously optimizes cost and accuracy in a feature selection algorithm. Therefore, a cost weight can be assigned to every feature and a maximum total weight can be defined as user input. The algorithm then performed a search for the most accurate feature set while considering the weight constraints defined before. The search was either sequential (forward or backward) or exhaustive.

Three classifier evaluation algorithms were part of the ECST: training-testing, cross-validation and bootstrapping. **Training-testing** evaluation uses one data set for classifier training and a disjoint set for classifier testing [Theo 09]. **Cross-validation** splits the available data set in n partitions and subsequently trains a classifier on $n-1$ partitions. Testing is performed on the remaining partition. The overall accuracy is averaged over all intermediate results. When data are partitioned by subjects, the procedure is called leave-n-subject-out cross-validation [Duda 01]. **Bootstrapping** evaluation repeatedly creates training and testing sets using random selection with replacement. The difference to cross-validation is the way how data are partitioned. In Bootstrapping, the same pattern might be part of the testing in more than one evaluation step. However, testing and training set are always disjoint [Poli 07].

Together with these training phase algorithms, a user was able to compare the accuracy and cost of different classification pipeline configurations. The next section describes the workflow of such an analysis and the different functional parts of the ECST.

Analysis Workflow

The ECST workflow consisted of input, output, integrated libraries and the functionality described in chapters 5.2.1 to 5.2.5 (Fig. 5.4 and appendix A).

Analysis **input** (Fig. 5.4, left and Fig. A.1, left) comprised training data loading as either raw data (e.g. comma-separated values) or as feature vectors. Thus, the described generic features or precomputed custom features can be used. Certainly, feature extraction cost can only be computed if the features included in the ECST are used.

The second **input** was the pipeline settings. The algorithms to be used for comparison can either be specified in the GUI (Fig. A.1 to A.3) or loaded from file. Storing and loading the pipeline from file made experiments repeatable and enabled the storage of well-proven settings. Pipeline settings are stored as XML file in a machine readable format.

The ECST used **external libraries** as computational core. These were the apache common math library (Version 3.3.0, [Comm 13]), the weka data mining software library (Version 3.6.6, [Hall 09, Witt 11]) and the libSVM library (Version 3.11, [Chan 11]).

The **functionality** of the ECST consisted of four main steps.

1. The first step was the configuration of the classification pipeline. It was either loaded from file or manually configured in a graphical user interface (GUI). In this step, the user selected algorithms and settings for every classification pipeline step (Fig.A.1 to A.3). Furthermore, the user can label and partition data according to the classification problem (e.g. 3 class problem) and evaluation strategy (e.g. cross-validation). The ECST computed results for each combination of pipeline steps.

2. The second step was the accuracy estimation of the classification systems. According to the chosen evaluation method, the classification rate of each pipeline configuration was estimated. See chapter 5.3 for a more detailed description of the necessary steps in cross-validation evaluation with feature selection. The output was the expected classifier accuracy for previously unseen data.

3. The third step created the final classification system using the complete input data in a single training step. We assumed that the complete input data were meant to be used to train a model for the implementation on embedded hardware. The output was a single trained classifier for implementation.

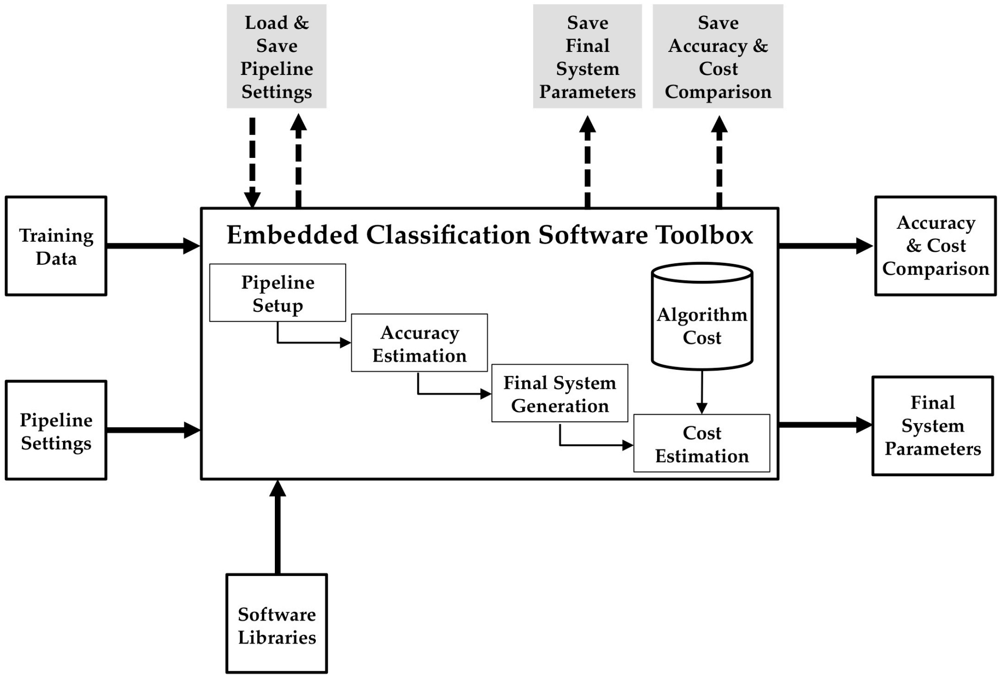

Figure 5.4: Embedded classification software toolbox (ECST) overview and workflow showing analysis input (left), output (right) and integrated libraries (bottom). The main functional parts of the ECST are shown in the center. The file interfaces for saving and loading are shown on top.

4. The fourth step was the cost estimation of the final classification system. Therefore, the algorithm analysis of computational cost and memory demand described in chapters 5.2.2 to 5.2.4 was used. The output was a cost report for the trained classifier determined in the third step.

The **output** of the ECST was an accuracy-cost comparison (Fig. A.3, right) and the parameters of the final classification system. Both results can be stored as XML file in a machine readable format.

The accuracy-cost comparison compares accuracy (e.g. classification rate) and cost (e.g. number of multiplications) of a specific pipeline setting to supports the solution of the accuracy-cost tradeoff. It provides a clear numeric overview of different algorithms to compare them directly.

The trained classifier can be implemented on target hardware using the final system parameters and the working phase algorithm description. Using the example of the SVM classifier (Eq. 5.20), the final system parameters are the support vectors SV_j consisting of y_j, α_j and $\mathbf{x}_j$ as well as the constant a_0. The working phase algorithm description is Eq. 5.20 that determines the class decision. The output of the ECST is a XML file that contains the final system parameters.

Software Architecture

The ECST is a graphical software toolbox that was developed in the Java programming language and is freely available from the project website[1]. Being written in Java, it is platform independent and capable of running on every system with a Java Virtual Machine. The ECST was designed to be extended and customized according to the user needs.

Two steps are needed to extend the ECST with new working phase algorithms. First, the algorithm has to be implemented and integrated according to the software interface of the ECST. The software interfaces of the ECST are clearly visible from the source code which is available on request. Their description is beyond the scope of this thesis. Second, the computational cost and memory demand analysis of the algorithms has to be provided. This analysis corresponds to a new row in Tab. 5.1. Technically, Tab. 5.1 is a machine readable configuration file (XML) that has to be extended with the corresponding entries. The precise description of this configuration file is beyond the scope of this thesis. However, its extension

[1] http://www.tinyurl.com/ecstproject

is straight forward once an algorithm is analyzed as described in chapters 5.2.2 to 5.2.4. Extending the ECST with new training phase algorithms requires the implementation and integration of these algorithms.

5.3 Classifier Accuracy with Feature Selection

The aspect described in this chapter is the accurate estimation of the algorithm accuracy in the offline phase of the embedded system development (Fig. 5.1). This consideration addresses an important question within the design of an embedded classification system: What is the expected accuracy of the system when confronted with previously unseen data? This chapter specifically considers the experimental setup where data are analyzed using cross-validation and feature selection. The correct partitioning of training and test data in the course of such experiments is crucial as result can differ considerably between correct and incorrect setup, see for example [Jens 12a]. Therefore, the next section describes the correct embedding of feature selection with a wrapper approach into a cross-validation experiment. Parts of this contribution were previously published in [Jens 12a] and the correct partitioning was used in all experiments of this thesis.

5.3.1 Classifier Accuracy Estimation

According to [Poli 07], the expected accuracy of a classifier suffers from two errors which Polikar denotes as bias and variance. The bias is the offset between estimated accuracy and actual accuracy on previously unseen data. The estimated accuracy is biased and most likely overestimated when the same data are used for training and testing. However, the variation when training and testing with the same data is minimal as only one classifier is trained and subsequently tested in one round of evaluation. The estimated accuracy is always identical on the same dataset.

The bias can be reduced when data are split into disjoint sets for training and testing. However, the classifier accuracy estimation varies when data are partitioned differently. This means that accuracy estimates differ when the testing and training sets change. This difference between results is called variance.

The evaluation techniques of cross-validation and bootstrapping avoid this bias, providing an unbiased estimate of the actual accuracy. Bootstrapping, in addition, also reduces the variance. Both techniques are more complex as the accuracy is estimated in an iterative process. Thereby, different data partitions are repeatedly used as training and testing sets as described in chapter 5.2.6.

In essence, the unbiased accuracy of a classifier can only be estimated if training and testing data are separated in the classifier evaluation. If this procedure is followed, classification accuracy of previously unseen data can be accurately estimated. The principle of separating training and testing data is valid for different aspects of classifier training like data partitioning, free parameter optimization and feature selection. The latter will be described in more detail in a cross-validation procedure.

5.3.2 Feature Selection and Cross-validation

Feature selection is a common approach to enhance the classification accuracy (chapter 2.2.6) and to reduce the computational cost. The latter aspect is underlined by the direct dependency of computational cost on feature dimension D (Tab. 5.1, e.g. naive bayes, classification via logistic regression). In addition, the cost of features extraction is reduced if less features are used for classification. These aspects make feature selection a valuable step in embedded classification system design.

Unbiased accuracy estimates require disjoint training and testing sets. This disjoint partitioning is also required in the feature selection step. The feature selection decision has to be based solely on the training set. The desired separation of training and testing can be visualized with the situation when the classifier is applied on previously unseen data. Classifier training as well as feature selection are based on the training data set and the previously unseen data are only available for evaluation with the final trained classifier.

The correct embedding of feature selection is shown using the example of a dataset with five subjects, leave-one-subject-out cross-validation and a wrapper approach for feature selection (Fig. 5.5). Features are selected with an inner leave-one-subject-out cross-validation loop. The classifier accuracy is determined with an outer classification cross-validation loop using the feature set determined with the feature selection loop. It has to be mentioned that this procedure creates one selected feature set for each classification cross-validation. Thus, the outcome of one classification loop evaluation is twofold. First, an estimated classification accuracy. Second, a selected feature set that achieved this classification accuracy. The overall accuracy can then be computed with averaging the results for all cross-validation steps. No single feature set can be linked to this overall estimated classification accuracy.

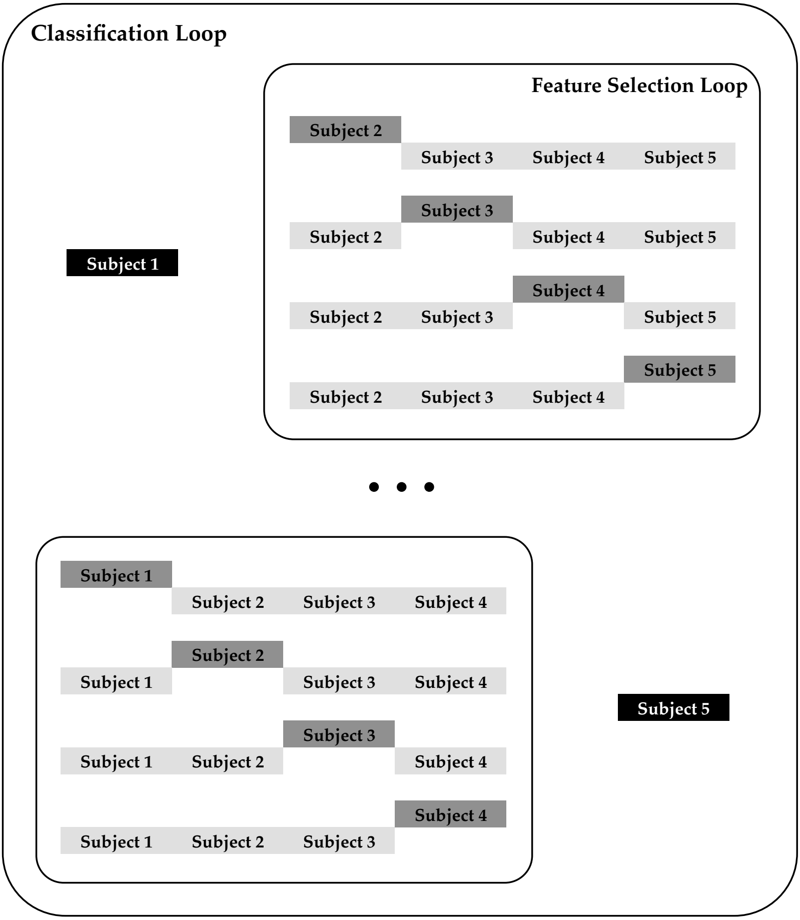

Figure 5.5: Embedding of feature selection in cross-validation for classifier accuracy estimation. Subjects with black background are test subjects for classification. Subjects with dark and light grey background are training subjects for classification. Subjects with dark grey are test subjects for feature selection. Subjects with light grey background are training subjects for feature selection.

The procedure described in Fig. 5.5 shows the embedding of feature selection using a wrapper approach (chapter 2.2.6) and cross-validation. When classifiers are compared, the procedure must be repeated for every classifier which is evaluated. The correct and incorrect embedding of feature selection using a wrapper approach was compared in a digital sports application for golf putt analysis [Jens 12a]. The results showed that the estimated classifier accuracy highly differs between correct and incorrect embedding. These results underline the importance of this aspect.

The described analysis sequence changes if a filter approach (chapter 2.2.6) is used. Filter approaches are classifier independent and, thus, selected features can be used across classifiers. A filter approach would result in a two step procedure. First, a single feature selection step for each subject is required. Second, the classifier accuracies are estimated for each classifier and each subject using the feature set determined in step one.

This sequence also changes if an embedded feature selection approach (chapter 2.2.6) is used. Using this feature selection that is embedded in classifier training, no additional feature selection step or loop is required. The procedure follows the normal cross-validation approach as it would be performed without feature selection.

5.4 Energy-efficient BSN Signal Classification

The aspect considered in this chapter is the energy-efficiency during the online phase of embedded system development (Fig. 5.1). Algorithms have to be designed and optimized to run on embedded hardware incorporating microcontrollers as processing units.

As described in chapter 1, batteries are often the largest components of sensor nodes. A reduction in energy consumption therefore facilitates smaller nodes or longer runtimes. Subsequently, sensor nodes are either less obtrusive or more convenient due to longer battery lifetime. Different aspects regarding energy-efficiency have to be already considered in algorithm design and comparison and affect the final embedded implementation.

In this section, four aspects are discussed. First, the aspect of power saving through hardware sleep mode is considered. Second, the influence of sampling rate reduction is discussed. Third, the technique of mapping floating-point operations to fixed-point operations is reviewed. Fourth, techniques to avoid processing steps or perform pre computations in the offline phase are described.

5.4.1 Power Saving through Hardware Low-power Mode

Energy-efficiency can be increased by reducing the power consumption of different hardware modules. An important aspect for optimization is the energy consumption during inactivity. The BSN sensor node hardware used for this thesis (chapter 3.3.3) contained a microcontroller whose CPU is disabled during inactivity and re-enabled when necessary [Burn 10]. Overall, five low-power modes can be set [Texa 15]. Other components (accelerometer, microSD card, Bluetooth radio module) can be powered off by the firmware when not in use. Four power modes of the sensor node are listed along with their respective power consumption (Tab. 5.2).

Power mode	Transmission	Reception	Idle state	Deep sleep
Power consumption	60 mA	40 mA	1.4 mA	50 μA

Table 5.2: Different power modes of a BSN sensor node and their corresponding power consumption [Burn 10].

These numbers highlight the benefits of periods of idle state or even deep sleep. These states can be reached when no computation is performed on the microcontroller and data are neither sensed nor transmitted. The swimming application described in chapter 8 gives further insights into the influence of power modes on energy consumption. In many cases, the frequency of entering a power-saving mode depends on the sampling rate. Therefore, the aspect of sampling rate reduction is discussed next.

5.4.2 Sampling Rate Reduction

The rate of data sampling influences the computational cost and memory demand of a BSN application. Higher sampling rates result in more collected data and usually also result in a higher computational cost for data processing. The memory demand is neglected in this section.

From a computational cost perspective, three benefits can be gained from lower sampling rates. These are less computational cost for sampling, less samples for the same processing interval and longer low power intervals. Fig. 5.6 illustrates these benefits of sampling at different rates (1 Hz, 2 Hz) and also illustrates that a processing interval of 5 s results in a different number of samples per interval (T, Tab. 5.1). In summary, the number of samples per interval is proportional to the sampling rate and therefore doubles the cost for sampling when the sampling rate is doubled. Additionally, double the amount of data must be processed in further steps. The time interval between sampling is larger for lower sampling rates and low-power modes could be entered for a longer period of time.

The sampling rate is an important aspect for the accuracy-cost tradeoff introduced in chapter 5.2. The number of samples directly influences the computational cost for feature extraction (Tab. 5.1). The sampling rate further determines the maximum frequency content that can be represented by the data. In essence, the Nyquist-Shannon sampling theorem states that frequencies up to half of the sampling frequency can be measured in the sampling process [Shan 49]. Thus, cost and accuracy are influenced by the sampling rate which must be considered when designing embedded classification algorithms.

A sample rate reduction for energy-efficient signal processing was proposed for swimming exercise tracking [Siir 11] and will be discussed in more detail with a similar application which is presented in chapter 8.

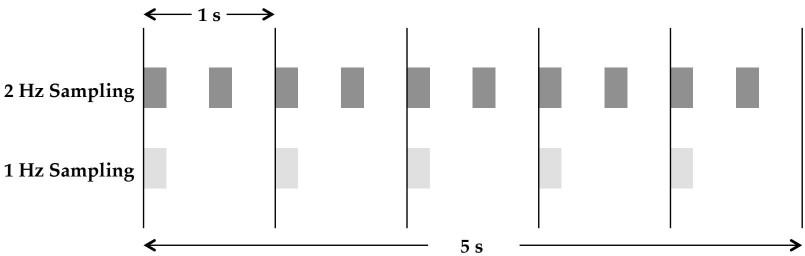

Figure 5.6: Influence of sampling rate reduction on number of samples and time between samples. Each sampling event is shown as grey box and the influence of a higher sampling rate is illustrated with double the number of dark boxes than light boxes.

5.4.3 Floating-point versus Fixed-point Arithmetic

This chapter considers the challenge of implementing floating-point calculations on fixed-point hardware. The equations in chapter 5.2 showed that classification algorithms usually contain mathematical operations with floating-point arithmetic. However, only a few microcontrollers directly support floating-point arithmetic in their instruction set [Gord 98]. Usually, the compiler maps floating-point operations to fixed-point operations. This mapping increases the number of operations when floating-point arithmetic is used [Smit 11]. The additional cost depends on the specific microcontroller and the type of operation. A general rule when optimizing the number of required operations on microcontrollers is to avoid floating-point operations wherever possible. This work briefly describes the aspect of converting floating-point to fixed-point arithmetic when implementing embedded classification algorithms. The topic is discussed in more detail by Gordon in [Gord 98].

Algorithm 2: Maps floating-point numbers to fixed-point numbers and back.

$exponent \leftarrow 10$
$factor \leftarrow 2^{exponent}$
$floatInput \leftarrow 3917.7251$
$fixedInput \leftarrow floatInput << exponent$
$fixedInput = 4011750$
Embedded Classification Algorithm
$fixedOutput \leftarrow 4011750$
$floatOutput \leftarrow fixedOutput >> exponent$
$floatOutput = 3917.7246$

Floating-point numbers can be converted to fixed-point numbers with a loss of precision (Alg. 2). This mapping is achieved with multiplication and truncation of the fractional part. The multiplication factor determines the precision leading to higher factors for increased precision. In order to map the fixed-point number back to floating-point, it must be divided by the multiplication factor. Usually, the mapping to fixed-point is performed before extensive calculations are performed. The mapping back to floating-point can sometimes be omitted. To additionally enhance the speed of the mapping, multiplication and division can be converted to shift operations if the factor is a power of 2. The procedure is summarized in Alg. 2 which shows the required operations and visualizes the loss of precision through the mapping operation. The swimming kinematics application described in chapter 8 provides an example where floating-point operations are mapped to fixed-point arithmetics. Making required operations as computationally cheap as possible is one possible optimization. Avoiding operations on embedded hardware is another option that is considered in the following section.

5.4.4 Processing Step Avoidance and Offline Processing

Computations on an embedded system are costly as processing resources (CPU, RAM) and battery capacity are limited. One consideration when optimizing the implementation of classification systems on embedded hardware is to avoid as many operations as possible. This can be achieved for instance by avoiding processing steps and moving processing to the offline phase (Fig. 5.1). These two considerations will be shown using the example of BSN signal classification. The provided examples could provide a basis for other signal processing techniques.

The feature set presented in this work (chapter 5.2.2) consists of signal characteristics and statistical measures. These metrics can be computed on different data characteristics like raw analog to digital (AD) converter output on the one hand, and acceleration and angular velocity on the other hand. One possibility to avoid a processing step is the signal processing in AD output domain. Omitting the calibration from AD values to acceleration or angular velocity units has two benefits: first, the computations performed for calibration can be avoided. Second, signals are present in fixed-point (see chapter 5.4.3). However, avoiding this calibration comes with several drawbacks. Firstly, differences in sensor calibration could bias the signal classification and subsequently reduce the classification accuracy. Secondly, features are harder to interpret when computed from raw AD values compared to them being computed from acceleration and angular velocity units. Third, prior knowledge regarding possible or relevant data values for acceleration and angular velocity cannot be integrated if analysis is performed in AD output space. Nevertheless, omitting signal calibration could make this type of signal processing more energy-efficient.

Omitting processing steps has to be carefully reviewed as these steps might be crucial for classification accuracy. One candidate for removal is feature normalization (chapter 5.2.3). Normalization is a crucial step for classifiers that use the distance of patterns in the feature space. Examples of these classifiers are SVM and NN (chapter 5.2.4). We speculate that other classifiers, which are based on feature value thresholds, show similar accuracy on non-normalized feature sets. Examples of these classifiers are AB and PART (chapter 5.2.4). Another aspect can influence the need for feature normalization. When features are already roughly normalized, an additional normalization step can be avoided. An example of such features is maximum, minimum and mean value (chapter 5.2.2). The influence of processing steps on the accuracy and cost of a specific classification problem can be experimentally compared with the ECST (chapter 5.2.6).

Another possibility to reduce the processing load of the embedded system is to move computations to the offline phase. One example is the simplification of the SVM classification step for a linear kernel (chapter 5.2.4). The equation for SVM classification (chapter 5.2.4, Eq. 5.20 & 5.21) with the main computations

$$\sum_{j=1}^{SV} y_j \alpha_j \, \mathrm{K}(\mathbf{x}_j, \mathbf{x}_i) + \alpha_0 \tag{5.24}$$

can be simplified to

$$\boldsymbol{\beta}^{\mathrm{T}} \mathbf{x}_i + \alpha_0 \tag{5.25}$$

if a linear kernel ist used. Thereby, SV denotes the number of support vectors, y_j the class label and α_j the weight of support vector $\mathbf{x}_j$. The variable a_0 is a constant and $\mathrm{K}(\mathbf{x}_j, \mathbf{x}_i)$ the kernel function of the support vector $\mathbf{x}_j$ and the input pattern $\mathbf{x}_i$. The variable $\boldsymbol{\beta}^{\mathrm{T}}$ denotes the preprocessed support vectors in the linear kernel version and can be computed in the offline phase. In essence, the simplified version (Eq. 5.25) is an implementation alternative that shifts computations to the offline phase through classification model simplification. The reduced amount of operations on the embedded system can be achieved without a loss of precision.

The swimming application described in chapter 8 uses processing in AD space and the above-mentioned SVM offline pre computation.

5.5 Accurate HMM Modeling for BSN Signal Analysis

The HMM technique is a powerful approach for sequential BSN signal analysis. See chapter 2.3 for an introduction to HMMs. It is important to consider how observable temporal phases of BSN signals are modeled in order to use the complete power of the HMM methodology. The modeling of temporal phases occurs in the design phase of an HMM analysis algorithm and the presented design considerations can support the development of HMMs with higher analytical precision. These considerations are used in the development of the plyometric training application that is described in chapter 6.

5.5.1 HMM Modeling with Temporal Phases

Different applications use HMMs for BSN signal analysis to deduce temporal phases of interest. Examples are ground contact time determination in jumping sequences [Jens 14b] and kinematic phase determination in breaststroke swimming [Dada 13a]. Temporal phases of interest were directly modeled as a single HMM state. This is an unnecessary restriction of the HMM model as it directly maps observations (BSN signals) to model states (temporal phases). However, the idea behind HMMs is that this mapping is hidden as mapping observations to model states and mapping one model state to another are both probabilistic processes (chapter 2.3). Thus, it should be modeled as such in order to unfold the full potential of the HMM methodology. Such a modeling is introduced in the following sections.

5.5.2 Accurate HMM Modeling with Hidden States

The main idea of accurate HMM modeling which is described in this work is that temporal phases of interest are modeled with multiple states (Fig. 5.7). This enhancement introduces the hidden component; the model receives the freedom to represent temporal phases with multiple states according to the characteristics of the data of that phase. The assumption behind this modeling approach is that temporal phases comprise of subphases of different lengths and with different data characteristics. In the context of HMMs, the temporal phases are denoted by emission states and comprise several model states.

HMMs are capable of solving supervised classification tasks as well as unsupervised classification tasks (chapter 2.1). The labeling procedure is described for the temporal phase determination example.

In the supervised case, model states are labeled with linear alignment. This means that all model states for the same temporal phase have equal length as initial condition for model training. In Fig. 5.7, the phases 1 to 4 would have the same length. During training, lengths are adjusted according to the training data. Estimated temporal phases can then be calculated during the working phase by summing the subsequent model state lengths.

In the unsupervised case, the complete BSN signal sequence is modeled with linear alignment. Again, model training adjusts the model state lengths according to the data. Temporal phases can then be deduced from the resulting model phases.

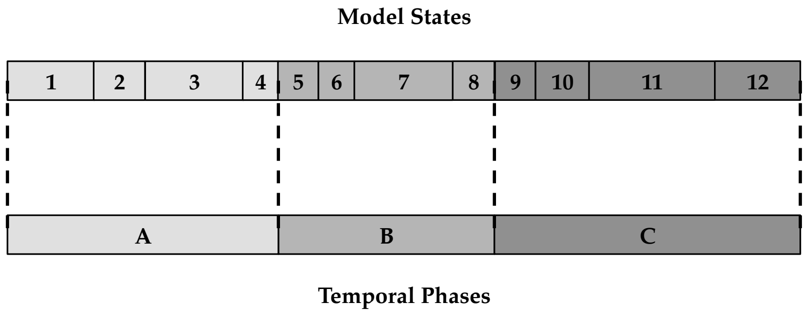

Figure 5.7: Accurate temporal phase modeling with trained HMM model states. Each temporal phase (A,B,C) is modeled with four model states (1–12).

The concept of accurate modeling using HMMs can be further enhanced. One enhancement is the introduction of meta states. Meta states combine multiple temporal phases and can be used to model movement events. Therefore, the HMM is enhanced with an event detection functionality. Events and their corresponding phases can be detected in a time series of data containing events and idle phases between these events. Different idle states can be modeled in more detail but it is often sufficient to model all non-event data as undefined state. The meta state analysis layer can be modeled in a supervised or unsupervised manner. Thus, a HMM with meta states can perform two analysis steps: event detection and temporal phase determination. This is a valuable concept when the temporal phases of specific events are of interest and when these events occur only once in a while. This accurate HMM modeling will be used in an application for plyometric training analysis described in chapter 6.

Chapter 6

Plyometric Training Application

This chapter presents the first of three BSN applications that are described in this thesis. The presented application uses BSN signals to determine temporal phase lengths in plyometric training. Therefore, an accurate HMM for sequential data analysis is used (see chapter 5.5).

Plyometric training for the lower body includes several body weight jumping-type exercises like drop jumps, countermovement jumps, alternate leg bounding and hopping [Flec 04]. This work focuses on a mobile BSN application for the drop jump exercise. Measuring the foot-wise ground contact time (GCT) with a mobile system can be used to determine and monitor the most effective training setup like drop height and number of repetitions according to predefined exercise goals. A wearable system to determine the GCT would be a valuable support tool for coaches and athletes (see chapter 1.2.3). Parts of this section were published in [Jens 14b].

First, the data collection is summarized. Second, the data analysis with a HMM is described. Third, the implementation and evaluation of the data analysis is introduced. Fourth, the results are summarized and discussed.

6.1 Data Collection

Two data collection studies were conducted. In the first study (UNIFORM), we collected data of a homogeneous group of trained athletes. High-speed video recordings were used as reference. We conducted the second study (DIVERSE) with a diverse group of subjects containing trained athletes as well as untrained subjects. In DIVERSE, a force plate was used as reference for the GCT validation. All subjects signed an informed consent form.

In both studies, subjects were equipped with an instrumented sports shoe on each foot (Fig. 6.1). Each shoe (left & right) comprised a Shimmer platform IMU as described in chapter 3 that was mounted on the heel (laterally) using a plastic clip. A wide range of shoe sizes were available and chosen according to the subjects foot size. The sensor node comprised a 3-D accelerometer (range ± 6 g) and a 3-D gyroscope (range ± 500 °/s). Data were sampled with 512 Hz.

Figure 6.1: Instrumented sports shoe with a Shimmer platform IMU. The IMU was mounted with a plastic clip.

6.1.1 UNIFORM

We recruited eight male subjects (age [years]: 24.4 ± 2.3, height [cm]: 185.4 ± 6.5, weight [kg]: 83.5 ± 8.2) for data collection. They were trained athletes and had prior experience in plyometric training. A minimum of 50 cm jump height in a jump and reach test [Viit 88] was the criterion to participate in the study.

For data collection, data were wirelessly transmitted to a recording PC. The sensor nodes were equipped with the BTStream firmware (Version 1.2, provided by manufacturer) and calibrated with the 9-DOF calibration tool (Version 2.0, provided by manufacturer). As reference, a Casio EX-FC100 high-speed camera (Casio, Tokio, Japan) recorded the experiment with 1000 fps. Both sensors and the video recording were synchronized with a specific movement pattern in front of the camera. This movement was a strong vertical acceleration that could easily be identified in the IMU raw data and the video recording.

In order to collect drop jump data, a box (drop height: 30 cm) and a hurdle (height: 50 cm) with 70 cm space in between were set up (Fig. 6.2).

Subjects were instructed to drop off from the box, jump over the hurdle and walk back to the drop platform to perform the next jump. Overall, eight experiments comprising a sequence of five drop jumps were conducted. Further plyometric exercises recorded in the same session were not used in this thesis.

We recorded simultaneous data from the left and the right instrumented shoes as well as the reference video. The video was analyzed frame-by-frame to extract the reference information of drop jump GCT, flight time and landing. Thus, 80 labeled drop jumps were used for further analysis.

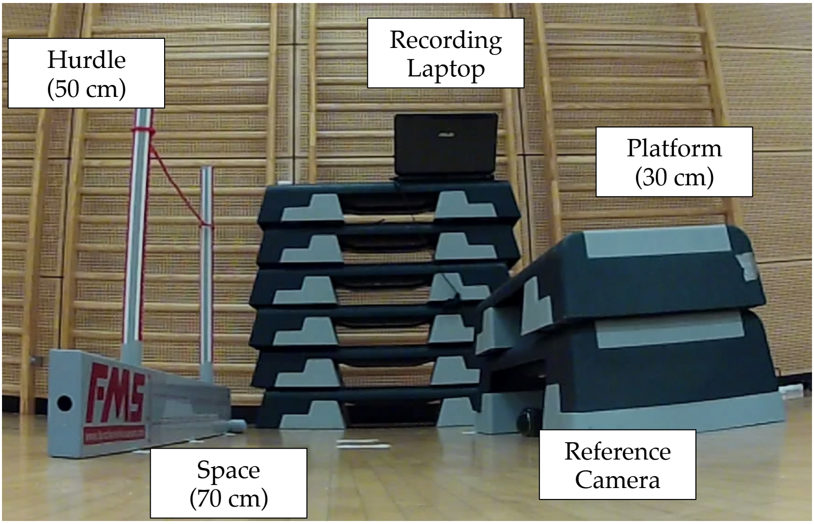

Figure 6.2: Data collection setup for the recording of the drop jump exercise in the UNIFORM study.

6.1.2 DIVERSE

This dataset comprised 9 subjects (4 female subjects, age [years]: 24.1 ± 2.3, height [cm]: 180.6 ± 10.4, weight [kg]: 76.1 ± 12.2) of varying training levels. There was no inclusion criterion and the population was intentionally selected to be diverse. In DIVERSE, a different reference modality was chosen as video processing in UNIFORM turned out to be labor intensive.

For data collection, data were logged on the sensor node as radio interferences with the reference system were detected in a pilot data collection.

The sensor nodes were equipped with the SDLog firmware (Version 1.4.0, provided by manufacturer) and calibrated with the 9-DOF calibration tool (Version 2.0, provided by manufacturer). As a reference, a fully instrumented treadmill (Bertec Corp., Columbus, USA) was used. This treadmill contained two force plates that recorded ground reaction forces with 1000 Hz. Experiments were additionally recorded with a Casio EX-FC100 video camera (Casio, Tokio, Japan) with a frame rate of 15 Hz to get an experiment overview. Sensors and camera were synchronized with a specific movement pattern in front of the camera. This movement was a strong vertical acceleration that could easily be identified in the IMU raw data and the video recording. The treadmill was not synchronized with the video and sensors as the jumping event was easily visible in the ground reaction force raw data.

A table was used as drop platform providing an elevation of 30 cm above the instrumented treadmill (Fig. 6.3). After performing the drop jump, subjects had to walk and step back on the drop platform to perform the next jump. Overall, nine experiments of a sequence comprising six drop jumps were conducted. Further plyometric exercises recorded in the same session were not used in this thesis.

We recorded simultaneous data from the left and the right instrumented shoes and the video recording. The video was used to identify the drop jump events but not for precise labeling. The precise labeling of the drop jump's temporal phases was based on two steps: visual inspection of the IMU data and data from the treadmill reference.

First, the impact of the drop jump was identified by visual inspection of the data. Data from the UNIFORM study revealed characteristic IMU signals at the beginning of the drop jump. This signal was used to identify the impact of the drop jump. Second, the reference information of drop jump GCT, flight time and landing was obtained from the force plate analysis. These temporal phase lengths were identified in the treadmill recordings. To identify these phases, the force plate noise without any load was measured. We considered values above the noise level as ground contact. Overall, 108 labeled drop jumps were used for further analysis.

6.2 Data Analysis

Both data collections setups allowed to measure the GCT for each foot individually. We present a methodology that uses the same data analysis model for both feet, but the analysis of each foot was performed

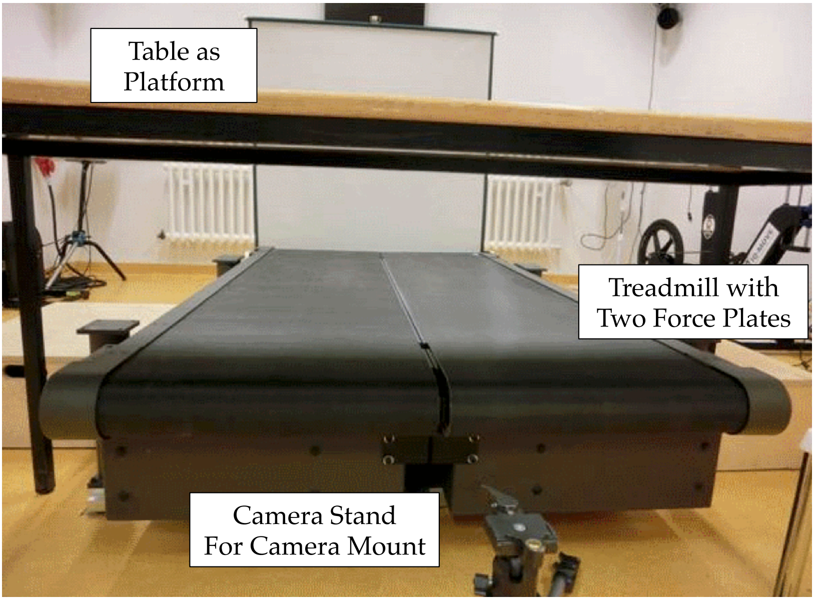

Figure 6.3: Data collection setup for the recording of the drop jump exercise in the DIVERSE study.

individually. Analysis was performed on the complete labeled sequences comprising standing, walking, and drop jumps.

6.2.1 Preprocessing

Five preprocessing steps were conducted before data were analyzed in more detail. The preprocessing resulted in 12-D data in feature space (also: observation) comprising 6-D raw data and 6-D first order derivatives.

1. Data were calibrated with the manufacturer's calibration software (see chapter 3).
2. Data of the right and the left shoe were transformed to the same coordinate system to be able to process both sides with the same data analysis algorithm. We adapted the signs of the accelerometer axis in the transversal plane to account for the different lateral movement directions of the right and the left foot. We inverted the gyroscope axes to harmonize supination and pronation movements of the right and the left foot.

3. Data packets lost during wireless transmission were reconstructed for precise temporal phase determination. Therefore, we inspected the timestamps and interpolated missing data linearly. Data loss was not registered during jump events and, thus, did not influence the further processing. The number of data loss events was so low that it was not evaluated any further.

4. The 6-D input data were enhanced with the first order derivatives estimated with a regression polynomial. This step was performed to add context information and is described in more detail in [Schu 95] (Eq. 3.65). In the following sections, the resulting 12-D data will be referred to as "features".

5. Features were normalized to uniform mean and variance. Therefore, we assumed that each feature was distributed according to the Gaussian distribution. We computed the parameters for the normalization with the maximum likelihood algorithm [Theo 09].

6.2.2 Pattern Analysis

The sequence of drop jumps was modeled as an HMM and temporal phase lengths were determined using this model. The strength of HMMs is the combination of probabilistic data (observation) and sequence (state) information [Rabi 89]. Both the collected experiments as well as the drop jump exercise comprised a consecutive sequence. The experiments consisted of five (UNIFORM) and six (DIVERSE) drop jumps respectively, surrounded by arbitrary movements (step, standing). The drop jump movement itself was comprised of the three strictly consecutive phases representing the foot GCT, the flight phase and the subsequent landing.

The model setup will be described from the lowest level to the highest (Fig 6.4) and follows the design consideration described in chapter 5.5. On the lowest level, five hidden states were used to model each emission state that coincided with physically observable phases.

The emission states were:

- GROUND: Phase of ground contact after dropping from platform
- FLIGHT: Flight phase of drop jump after take-off
- LANDING: Landing phase of drop jump
- UNDEF: Accumulative phase for standing, walking and arbitrary movements which coincides with a meta state as described below

While labeling was available for the emission states, the hidden states were labeled with linear alignment (chapter 5.5). Overall, the model contained 20 hidden states as each emission state was modeled with five hidden states.

The three emission states of the drop jump (GROUND, FLIGHT, LANDING) were combined to the meta state denoted by JUMP. The UNDEF emission state represented its own meta state. Using these meta states and the experiment information, the analysis was restricted to the sequence of five or six jumps with undefined movements in between. The number of jumps in a sequence were used for a forced alignment in the decoding phase where testing data were processed. This procedure was the event alignment step. The number of jumps in a sequence for the forced alignment can be easily provided by the coach or the athlete. The result of the HMM analysis was a sequence of emission states. The automatically determined GCT based on the IMU measurements was the length of the GROUND phase.

6.3 Implementation and Evaluation

We used the Java Speech Toolkit (JSTK), an open-source speech toolkit implementation [Univ 14], for implementation and evaluation. The observation distribution of each state, encoded in matrix **B**, was modeled with Gaussian Mixture Models (GMM) with ten components [Theo 09]. The distributions were trained with 25 iterations of the Expectation Maximization (EM) algorithm [Theo 09].

Matrix **A**, encoding the state transition information, was trained with 20 iterations of the Viterbi algorithm [Rabi 89]. The Viterbi beam search algorithm was used to determine the most probable state sequence from which the temporal phase information of an experiment was deduced. The analysis was conducted in a leave-one-subject-out cross-validation procedure as described in chapter 5.2.6.

We conducted three experiments. Each dataset (UNIFORM, DIVERSE) was evaluated individually and subsequently combined (COMBINED). Each foot (left and right) was evaluated individually. We used three measures to compare the performance.

$$\text{Absolute error} = \frac{|\,\text{GCT}_{\text{reference}} - \text{GCT}_{\text{analysis}}\,|}{\#\text{JUMP}_{\text{detected}}} \tag{6.1}$$

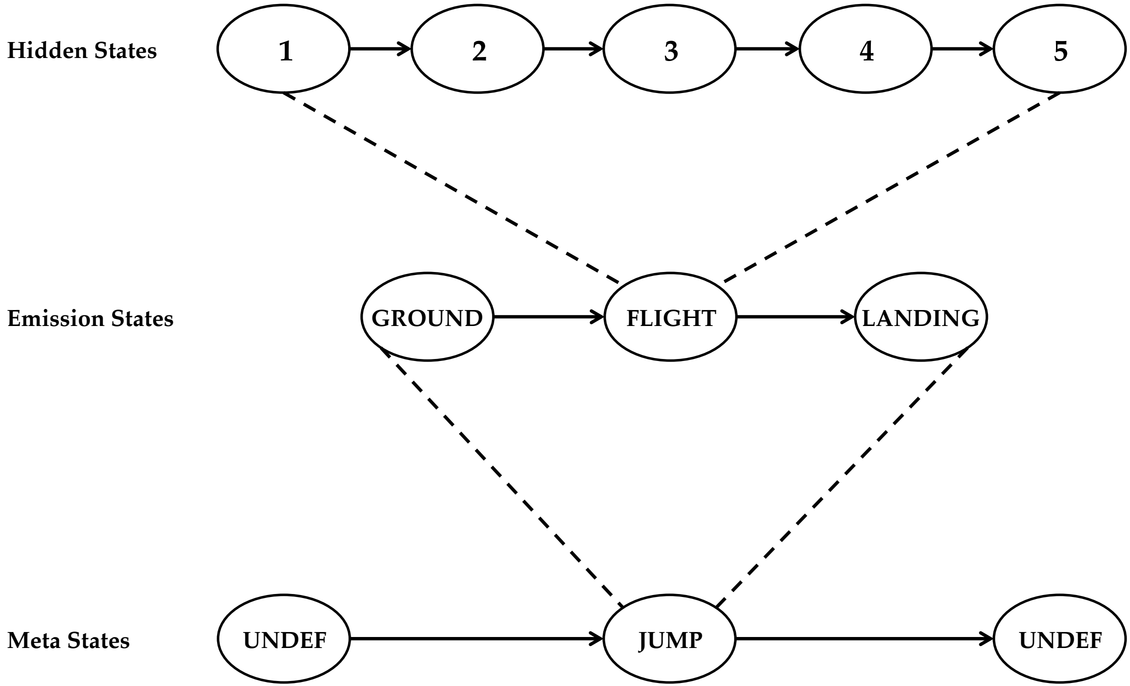

Figure 6.4: Hierarchy of the HMM model for the ground contact time calculation during the drop jump exercise. The different levels of hidden states, emission states and meta states are shown as an example.

$$\text{Detection rate} = \frac{\#\text{JUMP}_{\text{detected}}}{\#\text{JUMP}_{\text{performed}}} \times 100 \tag{6.2}$$

$$\text{Mean error} = \frac{\text{GCT}_{\text{reference}} - \text{GCT}_{\text{analysis}}}{\#\text{JUMP}_{\text{detected}}} \tag{6.3}$$

Thereby, $\text{GCT}_{\text{reference}}$ denotes the GCT of the corresponding reference (video, force plate) and $\text{GCT}_{\text{analysis}}$ denotes the GCT of the HMM analysis. The variable $\#\text{JUMP}_{\text{detected}}$ denotes the number of jumps that were correctly detected and $\#\text{JUMP}_{\text{performed}}$ denotes the number of jumps that were actually performed.

6.4 Results

The results revealed that all analyses achieved an absolute error of 10.8 ms or lower and a detection rate of 92.6% and above (Tab. 6.1). In absolute error and detection rate, UNIFORM outperformed COMBINED. DIVERSE showed the worst performance. The mean error was positive and 1.7 ms or below in all experiments.

Dataset	Absolute error [ms]	Detection rate [%]	Mean error [ms]
UNIFORM	4.3	95.0	1.7
DIVERSE	10.8	92.6	0.9
COMBINED	8.2	93.6	0.8

Table 6.1: Result overview of the ground contact time measurement based on inertial data. Results are shown for each experiment and averaged over all correctly detected jumps.

6.5 Discussion

The presented results cannot be directly compared to the prior art of Patterson et al. [Patt 10] as the reactive strength index was used to compare IMU and reference results in their work. The presented results can be compared to those from Jaitner et al. [Jait 14] and Jensen et al. [Jens 14b]. A comparison with Jaitner et al. revealed that the analysis of the UNIFORM dataset (Tab. 6.1) achieved similar results (absolute error: 3.4 ms, detection rate: 94% [Jait 14]). The results on the UNIFORM dataset and the results reported by Jaitner et al. were both achieved with data of experienced athletes. However, Jaitner et al. did not fully describe the experimental setup and the underlying data analysis algorithms. Therefore, a more detailed comparison could not be performed. Their analysis was implemented on the sensor node and integrated in a BSN application comprising of multiple sensor nodes and a smartphone. For the presented analysis, these steps have to be performed in further research. A comparison with Jensen et al. revealed that their investigation achieved lower results with an analysis of the UNIFORM dataset. The presented analysis of Jensen et al. was not based on an accurate modeling of HMM (chapter 5.5). An absolute error of 12.3 ms for the ground contact time and a detection rate of 98.8% were reported by this previous investigation. This result highlights the benefits of the accurate modeling of HMMs as superior results in the absolute error of GCT were achieved. We assume that the absolute error of 10.8 ms or lower is sufficient for most training applications that demand a specific training zone. However, the presented results might be further improved with different boosting techniques like signal-shift boosting [Jens 14b] or feature boosting [Dada 13a].

The analysis performed better on the UNIFORM dataset. Thus, the analysis of a homogeneous population with athletes having experience in plyometric training achieved more accurate results. We assume that this is due to the more controlled movement execution and lower variation between subjects.

The provided analysis processed both sides separately, but with the same analysis model. The methodology can therefore also be used for single-leg jumps which are quite common in plyometric training. Furthermore, landing symmetry can be assessed as GCT data for each side are available. Landing symmetry describes the time between the first and the second foot touching the ground in a two-legged jump. A long time difference may cause or contribute to a muscle imbalance [Ball 10] which will have negative long-term training effects [Hay 06].

The results showed a detection rate of above 92%. A more detailed analysis revealed that misdetections coincided with steps when walking and climbing back onto the drop platform. These misdetections might be excluded by combining data or jump detection from both feet. A good fusion strategy would be to combine the outcome of the jump detection of each foot. This enhancement could differentiate the alternating movement of walking and the simultaneous movement of two-legged jumps.

The positive mean error showed that the GCT was overestimated when averaged over all experiments. A more detailed analysis revealed different error situations. Some subjects showed either a positive or a negative mean error for both feet. We assume that this error is due to their specific jumping technique. In contrast, some subjects showed a positive (negative) mean error for one foot and a negative (positive) mean error for the other foot. We assume that this error is due to landing symmetry differences.

A more detailed analysis of the absolute error revealed that the majority of values were below 20 ms. The highest registered error was 127 ms. We could not identify subjects where analysis generally fails and we therefore speculate that the results will become more stable with more data. Further analysis has to be conducted to explain the influence of landing symmetry, jumping technique and the edge cases where precise estimation fails.

The analysis relied on the prior knowledge of how many jumps are expected in one exercise. In the presented datasets, experiments that included series of five (UNIFORM) or six (DIVERSE) drop jumps were analyzed. In training practice, this information can be easily given as the structure of the plyometric exercise is normally known. However, the HMM analysis framework allows an analysis without the prior knowledge of the number of expected jumps. Thus, the analysis can be enhanced to detect an unknown number of jumping events and analyze the temporal phases of these events.

The presented HMM provides a more advanced analysis than the previously described threshold approach [Patt 10]. We believe that the increased modeling power of HMMs is more suitable than thresholding to precisely detect the take-off which was identified as the main source of error (chapter 1.2.3). The take-off is not a clear or visible signal characteristic in the raw inertial sensor data and therefore sophisticated methods are needed for identification.

The presented tool for GCT estimation based on inertial sensor data has the advantage of being user-centric and has the potential of being mobile. The user-centric nature enables coaches to track a crucial training parameter of an individual athlete. Thus, training parameters can be individualized for the athletes (e.g. drop height, number of repetitions). Furthermore, the system has the potential to identify individual fatigue if this is reflected in longer GCT times. The mobile nature enables coaches to use the system in training practice without the need of lab equipment or the restriction of a lab environment. Longterm data analysis and the collection of a high number of trials is also possible. However, real-time analysis on a mobile device (sensor node or processing node) has not been implemented so far.

The greatest technical challenge for a mobile implementation of the system is the high sampling rate. In the current setup, data of two sensor nodes are simultaneously streamed with 512 Hz. In a multi-user scenario, even more data would have to be transferred via wireless radio. To facilitate a mobile multi-user system with 512 Hz sampling, we propose an on-node implementation of the jump detection and GCT estimation to reduce the wireless transmission effort.

Chapter 7

Golf Putt Application

This chapter presents the second of three BSN applications that are described in this thesis. A BSN comprising a sensor node mounted on the golf club head and a processing node were used (see chapter 3.3.3). The application detected golf putt events and determined kinematic parameters of the putt that were analyzed to give an insight into the training progress. The analysis is based on a combination of pattern recognition methods (see chapter 2).

The literature review in chapter 1.2.4 introduced different systems that analyze inertial data for kinematic golf putt analysis and revealed the need for augmented feedback applications for training in the field. Besides, the need to investigate the learning path during golf putting was mentioned. The contribution of this thesis is a BSN that is capable of providing augmented feedback as well as an analysis of the training progress during golf putting. The presented methodology can be transferred to other sports or data analysis applications.

The main application features were an automatic putt detection with machine learning methods and a real-time parameter calculation in the club coordinate system. Furthermore, a progress data analysis methodology that reflects a learning path is introduced. Parts of this section have previously been published in [Jens 11, Jens 12a, Jens 14a, Jens 15b].

The chapter is organized in five parts. First, system hardware and collected data is introduced. Second, data analysis is described. Third, experimental evaluation is summarized. Fourth, results are presented. Finally, results are discussed.

7.1 Data Collection

This section introduces the system hardware and the collected data. We conducted two research studies. Study one collected data for algorithm development and model creation and study two was used to evaluate the automatic putt detection performance on a disjoint population. The data of study two also served as a basis for the training progress analysis. All subjects signed an informed consent form.

7.1.1 System Hardware

The BSN hardware comprised a single sensor node (Shimmer platform) and a processing node (Android platform) as described in chapter 3.3.3. The sensor nodes were equipped with the BTStream firmware (Version 1.2, provided by manufacturer), calibrated with the provided 9-DOF calibration tool (Version 2.0) and configured to collect 3-D accelerometer and 3-D gyroscope data. The accelerometer range was ± 1.5 g and the gyroscope range was ± 500 °/s. Data were sampled with 256 Hz and wirelessly transmitted to the processing node which was an Asus Nexus 7C (Asus Inc., Taipeh, Taiwan). The processing node ran a custom application based on the Shimmer Android driver library (Version 1.3, provided by manufacturer) which performed the calibration and the data processing. In general, the analysis application is runnable on every Android device. The sensor node was mounted on the upper side of the club head using adhesive tape. It collected 6-D kinematic data of the club head for real-time analysis on the processing node (Fig. 7.1).

The application provided two modes; data collection and data processing. Both modes needed the transformation functionality (chapter 7.2.1) to be run before collecting data. The data collection mode logged 6-D raw data to a file. The data processing mode ran the data processing as described below (chapters 7.2.2 & 7.2.3) and saved the extracted parameters to a file. File names were encoded with meta data like the subject ID and the training session number that were collected prior to starting the data collection (Fig. 7.1).

7.1.2 Model Creation Study (MODEL)

Data to develop and train the analysis algorithm were collected at Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU). 15 subjects took part in this study and were completely unexperienced golfers. Subjects received a basic introduction to the putting movement (grip positioning, putt phases,

Figure 7.1: Mobile golf putt analysis system comprising a sensor node mounted on the club head and a processing node. The club head coordinate system with accelerometer (straight arrow) and gyroscope (curved arrow) axes are shown. The application screen shows the data processing mode.

pendulum movement) but no coaching. We chose this kind of population as we intended to create a model based on the putt kinematics of beginners. Due to the introduction to the movement, the subjects had a basic understanding of the pendulum putting technique but were still in an early phase of learning.

The study used the data collection mode of the mobile application. The protocol comprised three putts from three different distances (1.5 m, 3 m, 5 m) with two different putters. The clubs were a TaylorMade™ Manta (TaylorMade Inc., Carlsbad, USA) and a Pro Ace™ 20704 (Pro Ace Ltd., London, UK). Data were collected on an artificial putting green and subjects performed one training swing before each putt. One subject performed two putts less and one subject performed four more putts than specified in the protocol. Overall, 272 putts and the same amount of training swings were collected. The 6-D raw IMU data were logged on the analysis device. The collected data were used to train the model for putt detection and putt candidate classification (chapter 7.2.2).

Week	Name	Putt Count	Collected Data
1	Pre Test	10	Kinematic, hit count, distance
2	Training 1	36	Kinematic
2	Training 2	36	Kinematic
3	Training 3	36	Kinematic
3	Training 4	36	Kinematic
4	Training 5	36	Kinematic
4	Training 6	36	Kinematic
5	Training 7	36	Kinematic
5	Training 8	36	Kinematic
6	Post Test	10	Kinematic, hit count, distance
6	Transfer Test 1	10	Kinematic, hit count, distance
7	Retention Test 1	10	Kinematic, hit count, distance
7	Transfer Test 2	10	Kinematic, hit count, distance
9	Retention Test 2	10	Kinematic, hit count, distance
9	Transfer Test 3	10	Kinematic, hit count, distance

Table 7.1: Overview of the data collection protocol in the TRAINING study. All putts were performed from 3 m distance to the hole. Transfer tests were performed on a different surface.

7.1.3 Evaluation and Training Study (TRAINING)

The system was used in a study at TU Dortmund University to assess putt detection performance and the effects of motor learning in repetitive training. The study used the data processing mode of the mobile application. Therefore, 11 subjects who were completely unexperienced golfers were recruited to perform putts on an artificial putting green. For training, subjects were instructed to repetitively perform putts in two sessions a week. Subjects received a short introduction before pre test but no additional coaching afterwards during repetitive training. They were instructed to try holing as many putts as possible.

The putt length was 3 m and a TaylorMade™ Ghost Spider (TaylorMade Inc., Carlsbad, USA) was used. The protocol consisted of pre test, training sessions, post test, retention tests and transfer tests (Tab. 7.1). The pre and post test assessed the performance right before and after training intervention. The training sessions comprised repetitive training twice a week from the same distance. Putts in the training sessions were performed in blocks of six. The transfer tests were performed after completion of training intervention on a different surface to assess transfer capabilities. Retention tests were conducted after completion of the training intervention to test the persistence of motor learning. Subjects were free to perform training swings prior to the putt.

For each detected putt, the automatically computed kinematic parameters were logged. If a putt was not detected, the study advisor logged a corresponding entry. Undetected putts were not repeated. The advisor logged misdetected putts. This was done each time the system displayed a detected putt although the subject did not perform a stroke. In addition to the kinematic parameters, the ratio of holed putts and the distance from hole after each putt (not used in this thesis) were collected in each test session. Subjects were also videotaped. Data collection comprised 358 putts for each subject. Subject 5 missed training session 8 due to illness. Thus, 3902 putts were performed throughout the TRAINING study.

7.2 Data Analysis

The data analysis section introduces the different functional parts of transformation, putt detection, parameter extraction and data logging.

7.2.1 Transformation

We transformed the calibrated sensor data to the club head coordinate system to assess the club head kinematics. Thus, the sensor can be mounted in any desired position (e.g. according to putter geometry, upside down, sideways, ...) or removed for charging. The transformation, however, has to be performed after each repositioning. Thus, the procedure was integrated in the analysis application and simplified as much as possible. First, the steps to acquire the necessary data for transformation are described. Second, the mathematical operations of the transformation are summarized. These two steps have to be repeated every time the sensor position changes.

Data Acquisition

The following steps were performed to acquire the vectors $\mathbf{z}_{\mathrm{REC}}$ and $\mathbf{x}_{\mathrm{REC}}$ for further processing:

1. A bull's eye spirit level was mounted on the upper side of the club head.
2. Using the spirit level, the club was positioned so that the z-axis (Fig. 7.1) was pointing vertically upwards.
3. IMU sensor data was recorded for 2 s, averaged and saved as $\mathbf{z}_{\mathrm{REC}}$ for further processing.
4. A bull's eye spirit level was mounted on the club face.
5. Using the spirit level, the club was positioned so that the perpendicular of the club face was pointing vertically upwards.
6. IMU sensor data was recorded for 2 s, averaged and saved as $\mathbf{x}_{\mathrm{REC}}$ for further processing.

Data Transformation

The application used the direction cosine matrix (DCM) as described in [Kuip 02]. The DCM denoted by $\mathbf{T}_{\mathrm{DCM}}$ transformed the IMU data vector $\mathbf{v}_{\mathrm{IMU}}$ (sensor coordinate system) to $\mathbf{v}_{\mathrm{CLUB}}$ (club coordinate system) with

$$\mathbf{v}_{\mathrm{CLUB}} = \mathbf{T}_{\mathrm{DCM}} \cdot \mathbf{v}_{\mathrm{IMU}}. \tag{7.1}$$

Thereby, $\mathbf{v}_{\mathrm{IMU}}$ denoted the raw sensor recording as received from the sensor node during putting. The variable $\mathbf{v}_{\mathrm{CLUB}}$ denoted the same recording transformed to the club head coordinate system as shown in Fig. 7.1 for further processing. Eq. 7.1 had to be performed for every raw sensor recording. The variable $\mathbf{T}_{\mathrm{DCM}}$ was determined in the data transformation step and stored for further processing. The following paragraphs describe the steps to calculate $\mathbf{T}_{\mathrm{DCM}}$.

The transformation matrix $\mathbf{T}_{\mathrm{DCM}}$ was computed with

$$\mathbf{T}_{\mathrm{DCM}} = \begin{pmatrix} \mathbf{x}_{\mathrm{IMU}} \cdot \mathbf{x}_{\mathrm{DEF}} & \mathbf{y}_{\mathrm{IMU}} \cdot \mathbf{x}_{\mathrm{DEF}} & \mathbf{z}_{\mathrm{IMU}} \cdot \mathbf{x}_{\mathrm{DEF}} \\ \mathbf{x}_{\mathrm{IMU}} \cdot \mathbf{y}_{\mathrm{DEF}} & \mathbf{y}_{\mathrm{IMU}} \cdot \mathbf{y}_{\mathrm{DEF}} & \mathbf{z}_{\mathrm{IMU}} \cdot \mathbf{y}_{\mathrm{DEF}} \\ \mathbf{x}_{\mathrm{IMU}} \cdot \mathbf{z}_{\mathrm{DEF}} & \mathbf{y}_{\mathrm{IMU}} \cdot \mathbf{z}_{\mathrm{DEF}} & \mathbf{z}_{\mathrm{IMU}} \cdot \mathbf{z}_{\mathrm{DEF}} \end{pmatrix}. \tag{7.2}$$

Thereby, the club coordinate system (Fig. 7.1, base vectors: $\mathbf{x}_{\mathrm{DEF}}$, $\mathbf{y}_{\mathrm{DEF}}$, $\mathbf{z}_{\mathrm{DEF}}$) and the sensor coordinate system in relation to the club coordinate system (base vectors: $\mathbf{x}_{\mathrm{IMU}}$, $\mathbf{y}_{\mathrm{IMU}}$, $\mathbf{z}_{\mathrm{IMU}}$) were used.

We defined the club head coordinate system with the x-axis pointing in playing direction, the y-axis pointing to the player (right-handed) and the z-axis pointing vertically upwards (Fig. 7.1). The corresponding vectors were

$$\mathbf{x}_{\mathrm{DEF}} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \ \mathbf{y}_{\mathrm{DEF}} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \ \mathbf{z}_{\mathrm{DEF}} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}. \tag{7.3}$$

The sensor coordinate system in relation to the club coordinate system was determined using the vectors $\mathbf{z}_{\mathrm{REC}}$ and $\mathbf{x}_{\mathrm{REC}}$ from the data acquisition step. The 3-D recording of $\mathbf{z}_{\mathrm{REC}}$ was directly used as base vector $\mathbf{z}_{\mathrm{IMU}}$ with

$$\mathbf{z}_{\mathrm{IMU}} = \mathbf{z}_{\mathrm{REC}}. \tag{7.4}$$

As the upper side of the club head and the club face were not necessarily perpendicular, we used $\mathbf{x}_{\mathrm{REC}}$ to define the base vector $\mathbf{y}_{\mathrm{IMU}}$ (Eq. 7.5).

$$\mathbf{y}_{\mathrm{IMU}} = \mathbf{z}_{\mathrm{REC}} \times \mathbf{x}_{\mathrm{REC}} \tag{7.5}$$

The remaining axis $\mathbf{x}_{\mathrm{IMU}}$ was computed as cross product of the previously defined sensor coordinate base vectors (Eq. 7.6).

$$\mathbf{x}_{\mathrm{IMU}} = \mathbf{y}_{\mathrm{IMU}} \times \mathbf{z}_{\mathrm{IMU}} \tag{7.6}$$

7.2.2 Putt Detection

The putt detection was a two-step procedure, comprising the identification of putt candidates with a HMM (see chapter 2.3) and the exclusion of misdetections with an AB classifier (see chapter 5.2.4).

Putt Candidate Detection

The purpose of this step was event detection, the extraction of putting events from a club movement data stream. During golf training, different movements like training swings, actual putts and arbitrary movements are performed. The analysis targets the reliable automatic detection of actual golf putts from input IMU data. This process has not been described in literature previously.

We used a HMM to model the phase sequence of a golf putt and extracted the corresponding phase lengths for final event classification. Our model comprised the four putt phases backswing (BS), foreswing (FS, also: forward swing or downswing), impact (IM) and follow-through (FT) (Fig. 7.2). Additionally, we modeled arbitrary movements of the club head with an idle and a noise phase. The additional phases were chosen to detect putts during free training without a triggering user interaction. The HMM sequence comprised a state transition from FS to FT as well as a transition from FS to IM. Thus, actual putts containing an impact and training swings (no impact) were modeled. Idle and noise states could be reached from BS or FT to restrict the model for improved detection performance.

Data were processed in a sliding window approach. We set the window size to 500 samples which is equivalent to 1.95 s to cover one stroke in a single window and used 50% overlap. The model was trained on labelled data with the Baum-Welch-Algorithm [Rabi 89] that determined the matrices **A** and **B** of the trained HMM (chapter 2.3). We followed the model described in [Burc 10, Kooy 13] for the putt phases labeling (BS, FS, IM and FT). The remaining data were labeled as idle and noise. Data were labeled as idle if the main rotational axis (gyroscope y-axis) showed values of below a threshold of 10 °/s. Data were labeled as noise if the values were above the threshold. All phases were modeled as multivariate Gaussian distributions. The evaluation was performed with the Viterbi-Algorithm [Rabi 89]. Each sequence that comprised an IM phase was considered as putt candidate. If the impact event was at the border of the window, we repositioned the window so that the impact was at 70% of the window and reran the state sequence determination. Finally, we computed the duration of the putt phases (BS, FS, IM, FT) and the overall putt length for final event detection.

Putt Candidate Classification

A putt candidate, found with the HMM, was either an actual putt (true positive) or a misdetected putt (false positive). These misdetections were e.g. training swings with a misdetected impact or arbitrary movements. We used a classification algorithm to exclude these misdetections from further processing.

Two classes, actual putts (PUTT) and other detections (OTHER), were used. The patterns for classification were HMM putt candidates represented with five features. These were the putt phase lengths (BS, FS, IM, FT) and the overall length (BS + FS + IM + FT) that were computed from the HMM output. Our analysis used an AB classifier. It was trained using the ECST (chapter 5.2.6). The result of the putt candidate classification decided whether the data were further processed (PUTT) or excluded from further processing (OTHER).

7.2.3 Parameter Extraction

Detected putts that were confirmed by classification were further processed. As described below, we segmented the movement in putt phases, filtered the data and extracted kinematic parameters. The determination of the lie and loft angle is thereby explained in more detail.

Putt Phase Determination and Filtering

The putt phase determination was initialized at the impact phase determined by the HMM analysis. However, the remaining HMM phase information was discarded as it was found to be error prone. The HMM result was only used for the detection of the putt event. The putt phase determination was a three step process:

1. We segmented the impact phase with a threshold approach. Therefore, we computed the squared differences of two subsequent values of the gyroscope y-axis of the labeled impact phases in the model creation set. The minimum of these values was used as threshold and all values above the threshold were assigned to the impact phase in detection mode.

2. For further processing after impact detection, data of all phases were filtered with a moving average filter of order five to remove high-frequency noise in the kinematic data.

3. Prior knowledge of the putt movement was used for further processing. Therefore, we used a putt model based on zero crossings of the main rotational axis (gyroscope y-axis) [Burc 10, Kooy 13].

Parameter Extraction

We selected 31 kinematic parameters that can be extracted from the 6-D IMU data (Tab. 7.2) and were previously described [Dela 97, Para 00, King 08, Burc 10, Sim 10].

The selection criterion was that parameters could be implemented with the use of 6-D club head motion data in one integration step at most. Integration of IMU data are affected by drift and error accumulates in repeated integration (see chapter 3.1.3). Due to this fact, we extracted angles and linear velocities (single integration) and omitted linear displacements (double integration). Errors are expected to be low due to short integration times as putts are typically shorter than 2 s [Burc 10]. Parameters were grouped in five categories (Tab. 7.3).

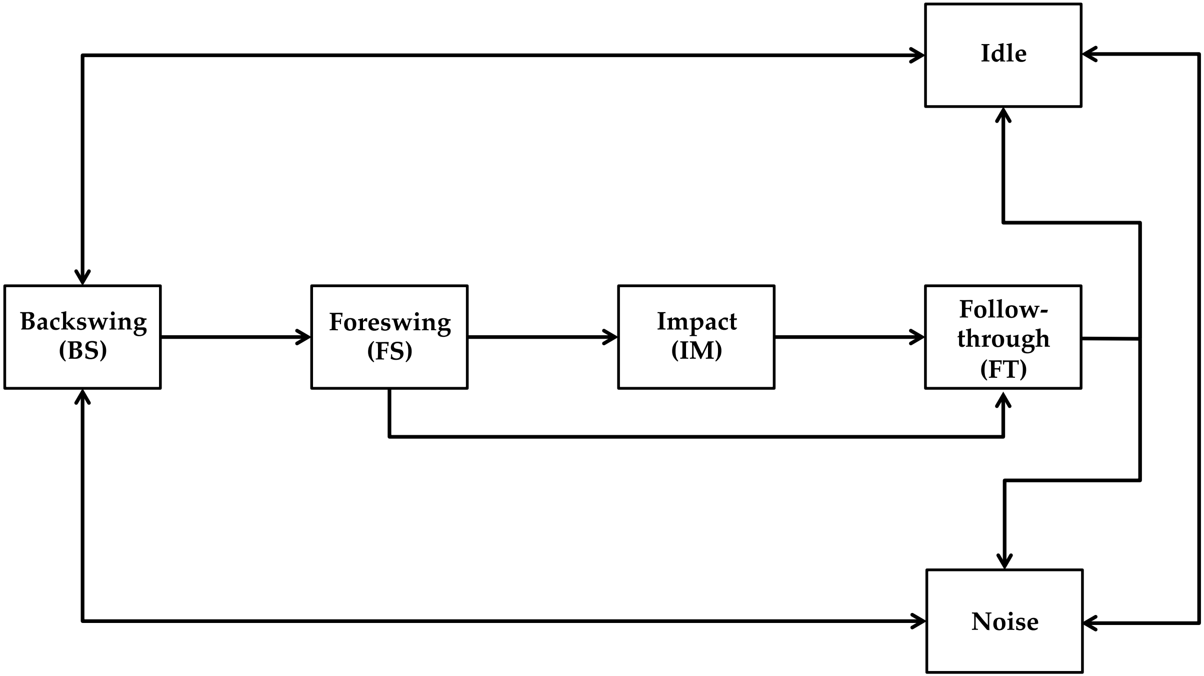

Figure 7.2: Simplified HMM model to detect putt candidates in sequential IMU data. This model incorporates training swings and actual putts with a transition from foreswing to impact and foreswing to follow-through. Transitions to the same state are omitted for enhanced readability.

Lie and Loft Angle Computation

The lie (P9) and the loft (P10) angle could not be determined with gyroscope integration and will therefore be described in more detail. These angles described the club orientation at ball impact. On impact, a lie angle other than zero reflects a rotation around the x-axis and a nonzero loft angle a rotation around the y-axis. As the club coordinate system was

Nr.	Axes	Description	Unit
P1	G-Y	Duration of putt	s
P2-P5	G-Y	Duration of BS, FS, IM and FT	s
P6	G-Y	Duration of swing (BS + FS)	s
P7	G-Y	Ratio of BS duration to FS duration	AU
P8	G-Y	Ratio of FS duration to FT duration	AU
P9- P10	A-X, A-Y, A-Z	Lie and loft angle at impact Avg. of 5 samples (~19.5 ms) before impact	°
P11- P13	G-X, G-Y, G-Z	Summed angle of BS and FS Computed for each axis separately	°
P14- P16	G-X, G-Y, G-Z	Summed angle of phase before IM Phase length 20 samples (~78 ms)	°
P17- P19	G-X, G-Y, G-Z	Summed angle of phase after IM Phase length 20 samples (~78 ms)	°
P20-P22	G-Y	BS, FS and FT angle	°
P23	G-Y	Ratio of FS angle to FT angle	AU
P24	A-X	Linear velocity at impact	m/s
P25	G-Y	Angular velocity at impact	°/s
P26	A-X	Summed acceleration of phase before IM Phase length 20 samples (~78 ms)	m/s^2
P27	A-X	Summed acceleration of phase after IM Phase length 20 samples (~78 ms)	m/s^2
P28	A-X	Maximum acceleration value in FS	m/s^2
P29	A-X	Maximum acceleration position in FS	% of FS
P30	A-X	Maximum velocity value in FS	m/s
P31	A-X	Maximum velocity position in FS	% of FS

Table 7.2: List and description of the computed putt parameters. Accelerometer axes are abbreviated with A-X, A-Y and A-Z. Gyroscope axes are abbreviated with G-X, G-Y and G-Z. Swing phases are abbreviated with BS (backswing), FS (foreswing), IM (impact) and FT (follow through). Parameters measured in arbitrary units are marked with AU.

created according to a zero lie and zero loft angle orientation of the club head, we used the base vectors and according planes to define the lie and loft angle. The angles were computed as projections in the y-z plane (lie angle) and x-z plane (loft angle). We averaged the angles using five values (19.5 ms) before impact.

The measured acceleration for lie and loft angle calculation is the combination of gravity and movement acceleration. We assumed that the movement acceleration is very low right before ball impact. Therefore, we neglected the movement acceleration by normalizing the measurements to 1 g and assumed that the measurements originated from gravity only.

Category	Description	Parameters	Number
1	Temporal	P1 - P8	8
2	Angles	P9 - P23	15
3	Velocities	P24 - P25	2
4	Acceleration around impact	P26 - P27	2
5	Foreswing profile	P28 - P31	4

Table 7.3: Putt parameter categories.

7.3 Evaluation

We evaluated the system regarding putt detection and analyzed the training progress during repetitive training. Putt detection results were drawn from the MODEL and TRAINING dataset. The training progress analysis is based on the TRAINING dataset.

7.3.1 Putt Detection

The putt detection was evaluated with the detection rate and false positive rate. The detection rate was calculated as

$$\mathrm{DR}\,[\%] = \frac{N_d}{N_p} \times 100 \tag{7.7}$$

Thereby, N_d denotes the number of correctly detected putts and N_p the number of performed putts.

The false positive rate reflects the performance regarding misdetections and was calculated as

$$\text{FPR}\,[\%] = \frac{N_m}{N_m + N_p} \times 100 \tag{7.8}$$

Thereby, N_m denotes the number of misdetected putts and N_p the number of performed putts.

The MODEL dataset was evaluated with leave-one-subject-out cross-validation (chapter 5.2.6). The TRAINING dataset was treated as disjoint testing set and an overall detection rate as well as the individual rate for each subject was calculated.

7.3.2 Training Progress

We analyzed the eight training sessions of the TRAINING dataset to provide an insight into the training progress of novices. We were interested in

1. The change of putting performance
2. Kinematic parameters representing training progress
3. The change of these parameters during training

We therefore analyzed the change of putting performance with a hit count analysis, ranked kinematic parameters according to their relevance in the training progress and analyzed the change of the most relevant parameters.

Putting Performance Analysis

The intervention in the TRAINING study was repetitive training without specific coaching or feedback. The target for the subjects was to improve their hit count that is measured as

$$\text{HC}\,[\%] = \frac{N_h}{N_p} \times 100 \tag{7.9}$$

Thereby, N_h denotes the number of holed putts and N_p the number of performed putts. Data from all subjects was combined. The hit count was computed in the pre test and in the post test.

Kinematic Parameter Ranking

In terms of knowledge of results (chapter 4.2.4), the subjects were able to see whether they holed the putt or not. Our analysis intended to reveal the presence and the type of the kinematic parameter change with repetitive training. Thus, we were not primarily interested in the actual parameter values but their progression over time. The training progress was not influenced by a coach. The presented analysis therefore observed the training progress instead of evaluating it. We chose a data driven evaluation to describe the change of kinematic parameters and therefore the training progress. This means that we selected and analyzed the change of the parameters without taking into account expected or predefined training outcome regarding kinematic changes.

The training progress was analyzed with data from training session 1 to training session 8 (Tab. 7.1). Pre, post, retention and transfer tests were intentionally excluded from the training progress analysis to ensure that the same amount of data was available for each analysis session (see putt count in Tab. 7.1).

The parameter ranking was conducted to reveal the relevant kinematic parameters that change with training. From a machine learning perspective, the identification of relevant parameters is a feature selection task (chapter 2.2.6). We used the information-theoretical concept of entropy to rank parameters according to their relevance for discriminating the training sessions [Witt 11]. The assumption that supports this approach is that parameters with high information gain contain more information for assigning putts to the correct training session than those with low information gain [Yu 03]. Therefore, these parameters are well suited to discriminate training sessions and subsequently reflect the strongest change during training.

According to [Yu 03], the information gain IG of the parameters Y and Z determines the amount of additional information about Y provided by Z and is computed with:

$$IG(Y|Z) = H(Y) - H(Y|Z) \tag{7.10}$$

In this equation, $H(Y)$ denotes the entropy of a parameter Y (regarded as random variable) and $H(Y|Z)$ denotes the entropy of Y after observing Z:

$$H(Y) = -\sum_i P(y_i)\log_2(P(y_i)) \tag{7.11}$$

$$H(Y|Z) = -\sum_j P(z_j)\sum_i P(y_i|z_j)\log_2(P(y_i|z_j)) \tag{7.12}$$

In these equations, $P(y_i)$ denotes the prior probabilities for all values of Y. $P(y_i|z_i)$ denotes the posterior probabilities of Y given the values of Z.

The result of the first step was a list of parameters sorted by information gain and therefore sorted by relevance for discriminating training sessions. Using this list, we selected the highest ranked 25% of parameters (seven parameters) for further analysis. This restricts the progress analysis to the most relevant parameters according to the information gain measure. Seven relevant kinematic parameters were further analyzed.

Analysis of Relevant Kinematic Parameters

We analyzed the change of the seven kinematic parameters selected with the above-mentioned ranking step. The inter-individual progress of a specific parameter with respect to the training session number was of interest. Data were analyzed in four steps.

First, all successfully detected putts were labeled with the sequential training session number (1 to 8). Second, all putts of all subjects were grouped by training session. This step was needed to illustrate the inter-individual progress. Third, the median for each parameter and training session was computed. We used the median instead of the mean as this measure is more robust regarding outliers. Fourth, the Spearman correlation coefficient of each parameter and training session was computed. The Spearman correlation coefficient of each parameter was computed to quantify training progress.

7.4 Results

7.4.1 Putt Detection

Performing leave-one-subject-out cross-validation on the MODEL dataset, the putt detection algorithm successfully detected 261 out of 272 putts resulting in an overall detection rate of 96.0%. The number of misdetected putts was 22, which corresponds to a false positive rate of 7.5%.

The system detected 2660 out of 3902 putts from the TRAINING study data resulting in an overall detection rate of 68.2%. In total, we observed that in 97 cases during the TRAINING study, a random movement or training swing was detected as putt. In effect, we obtained a false positive rate of 2.4%.

The detection rate varied throughout subjects, ranging from 98.9% to 3.1% (Tab. 7.4). Examining each subject separately, we observed that our system was either satisfactorily detecting putts (8 subjects, detection rate ≥ 83%) or rarely detecting putts (3 subjects, detection rate < 16%).

Subject	Detection Rate [%]
S1	83.0
S2	88.8
S3	96.7
S4	91.9
S5	84.5
S6	95.8
S7	98.9
S8	15.4
S9	3.1
S10	5.6
S11	88.0

Table 7.4: Results of the putt detection for each subject of the TRAINING study.

Rank	Parameter	Information Gain	Correlation Coefficient
1	P5	0.107	0.98
2	P12	0.059	0.88
3	P22	0.054	0.95
4	P29	0.053	0.88
5	P1	0.051	0.90
6	P30	0.044	-0.91
7	P24	0.042	-0.91

Table 7.5: Result of the training progress analysis. Parameters were ranked using the information gain criterion. Quantitative parameter change is reflected in the correlation coefficient relating training session and parameter progression.

7.4.2 Training Progress

The hit count increased from 10.0% before training intervention (pre test) to 39.1% after intervention (post test).

The subsequent training progress analysis was conducted with data from the eight subjects (TRAINING) that achieved a detection rate of more than 83.0%. The three remaining subjects were excluded as only 15.4% or less putts were available (Tab. 7.4). Overall, 1946 correctly detected putts were used for the analysis.

Seven parameters were selected according to their information gain (Tab. 7.5). Five of them showed a high positive correlation with the training progress (P1, P5, P12, P22, P29). Two parameters showed a high negative correlation with advancing training (P24, P30). Exemplarily, plots of the parameters P5 (Fig. 7.3), P22 (Fig. 7.4) and P24 (Fig. 7.5) are presented to illustrate the correlation tendency. Furthermore, visual inspection revealed training session instances where parameter and training session do not correlate. These were e.g. training session 8 and P5 (Fig. 7.3), training session 6 and P22 (Fig. 7.4) and training session 7 and P24 (Fig. 7.5).

7.5 Discussion

7.5.1 Putt Detection

MODEL Dataset

The cross-validated putt detection on the MODEL dataset achieved a detection rate of 96.0%. Thus, putts from three different distances were detected with high success rate. As the system was repeatedly trained and evaluated on disjoint data, the results implied good generalization capabilities of the analysis with HMM and AB. We assume that the MODEL population was homogeneous as all subjects received the same movement instructions as introduction (chapter 7.1.2).

The low false positive rate of 7.5% (MODEL dataset) showed that actual putts can be distinguished from other movements. However, the analysis still revealed a considerable amount of misdetections. A more detailed analysis showed that misdetections occurred mostly on training swings. We assume that this is due to the fact that putts from a short distance (1 m) were part of the MODEL dataset. Due to a lower acceleration and linear impact velocity, putts from the short distance and training swings show a similar visual data characteristic.

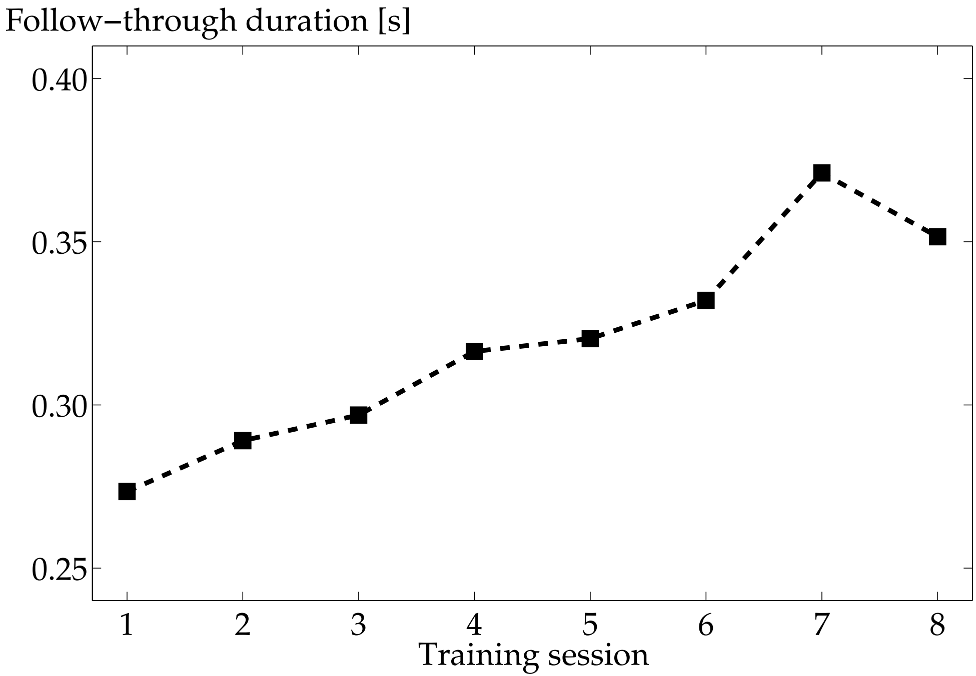

Figure 7.3: Median values of follow-through duration (P5, y-axis) for each training session (1 to 8, x-axis).

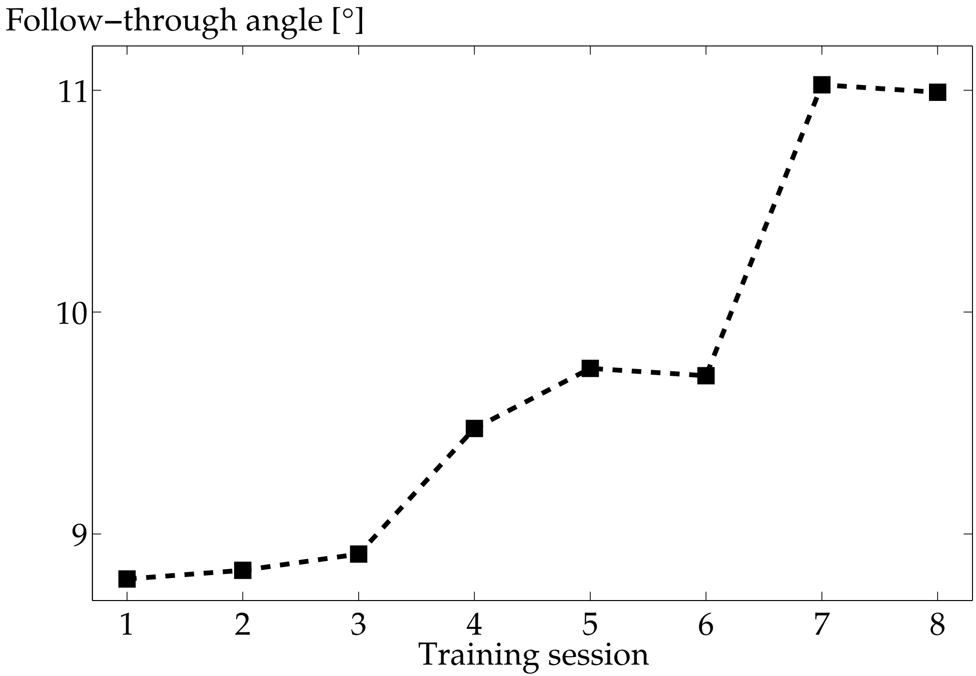

Figure 7.4: Median values of follow-through angle (P22, y-axis) for each training session (1 to 8, x-axis).

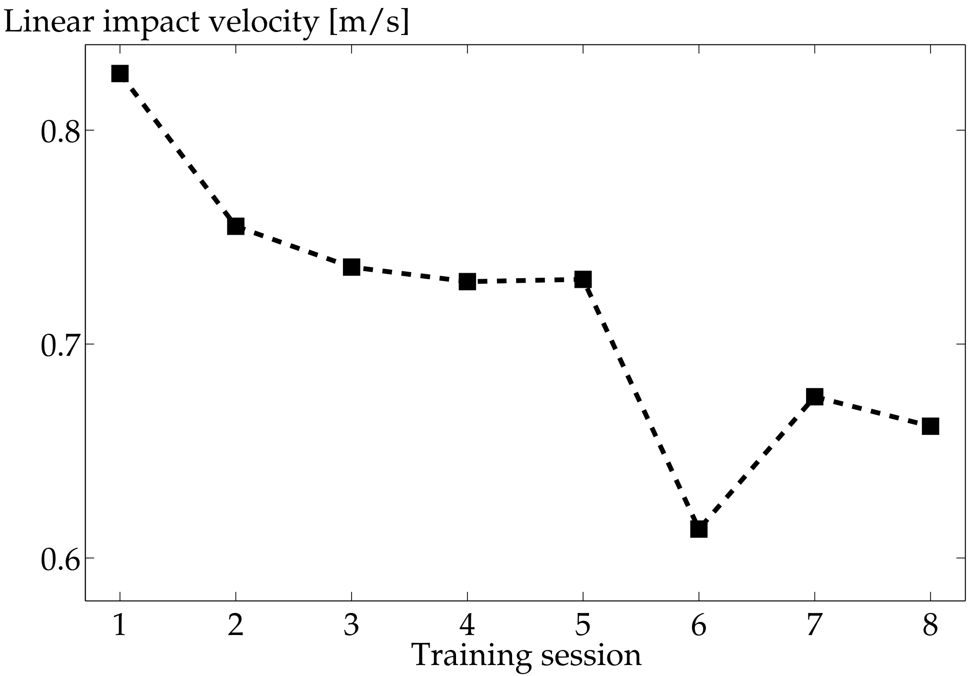

Figure 7.5: Median values of linear impact velocity (P24, y-axis) for each training session (1 to 8, x-axis).

Furthermore, the movement amplitude is lower on shorter putt distances and thus more similar to training swings. In essence, we speculate that the modeling power of the combination of HMM and AB is sufficient to distinguish putts from longer distances and training swings but not sufficient for putts from short distances.

There are several opportunities to improve the putt detection performance. First, the system can be restricted to long distance putts performed from more than 3 m. With this restriction, actual putts and training swings might be distinguished more easily. Second, the HMM can be enhanced as proposed in chapter 5.5 to omit the erroneous state detections (see chapter 7.2.3). With this approach the modeling power of putt candidate detection might increase. Third, the AB classifier can be enhanced with more features to increase the discrimination capabilities of the putt candidate classification. Currently, only state lengths are considered. The generic features presented in this thesis (chapter 5.2.2) might be able to boost the classifier accuracy and thus reduce misdetections. Fourth, more data could increase the discriminatory power of both the putt candidate detection with HMM and the putt candidate classification with AB.

The amount of evaluation data in the MODEL study was limited and a more extensive evaluation was performed with the TRAINING study.

TRAINING Dataset

The overall result of the putt detection on the TRAINING dataset was 68.2%. However, a more detailed analysis revealed detection rates of 83.0% or higher for eight out of eleven participants. We analyzed the video recordings of the three participants where putt detection failed. Their putts had a limited backswing and overall small movement amplitude and could not be regarded as a controlled movement execution. Obviously, the technique variation that caused misdetections was not represented in the MODEL dataset and therefore not detected in the TRAINING study. One explanation might be that participants in the MODEL dataset received basic technique instruction of how to putt properly (chapter 7.1.2). In contrast, the participants in the TRAINING dataset received only a short task description to omit the influence on the motor learning process (chapter 7.1.3). The findings on the TRAINING dataset showed that our system successfully detected putts if the movement followed a pendulum technique that is generally accepted to be most effective in putting [Dela 97]. Despite the fact that it would be possible to improve the putt detection by incorporating the technique of minimum backswing into

HMM modeling, we did not investigate this as we targeted a system for training an effective putting execution.

The results showed that the presented system can only be recommended to athletes that are able to follow this technique. The instructions as given in the MODEL study might be sufficient to putt with this technique. However, the detection rate of 83% or higher for the subpopulation that already followed this technique implies that the putt detection needs further enhancement. The four proposals mentioned above, restriction to longer putts, more accurate HMM and AB modeling as well as a higher amount of training data are assumed to enhance the putt detection performance.

The low false positive rate of 2.4% showed that random movements and training swings were well distinguished from actual putts. The participants were free to perform training swings prior to the putt, which is a common procedure in putting. However, a more detailed analysis showed that training swings are not commonly performed by unexperienced golfers.

Comparison of Datasets

A comparison of the MODEL and the TRAINING datasets revealed a lower detection rate in the TRAINING dataset. A possible explanation for the lower performance in the TRAINING dataset might be a different data collection setup. The MODEL study comprised a controlled environment where subjects were aware of the fact that data are collected. In contrast, during free putting in the TRAINING study, subjects trained independently and data collection was conducted in a subliminal way. We speculate that the MODEL and TRAINING study are influenced by the Hawthorne effect [Roet 03] in a different way. The Hawthorne effect describes the phenomenon that subjects are biased by the fact that they are part of a scientific study. Subjects of the MODEL study might be aware of the observation and their movements change due to a higher concentration or pressure. However, as we were interested in the performance of the system during putt training, the performance of the TRAINING dataset is a more realistic estimation of the expected putt detection performance. It would also be interesting to evaluate the performance of the system on lower and higher distances than the 3 m that were used in the TRAINING study.

A further comparison of the MODEL and the TRAINING datasets revealed a lower false positive rate in the TRAINING dataset. The subjects were free to perform training swings or arbitrary movements in the TRAINING study, but our analysis suggests that beginners do not commonly perform training swings prior to a putt. This is especially the case for the given task of performing consecutive putts.

To conclude the putt detection discussion, it can be summarized that the TRAINING dataset provided a high amount of disjoint evaluation data for a meaningful evaluation of the true performance of the system. The results (Tab. 7.4) underline the applicability of the system for automatic putt detection in training practice. However, detection rates are not sufficient for consumer products and possible ways to improve the detection were proposed. The results of putt detection and false positive rate cannot be compared with literature. Detection algorithms were either not described [Marq 07], not existing due to manual labeling [Sim 10, King 08] or neither mentioned nor evaluated [Kooy 13].

7.5.2 Training Progress

General Aspects of Training Progress

The analysis of the training progress during repetitive training is one application example for the mobile golf putt analysis system. The results showed that putting performance increased between pre test (10.0%) and post test (39.1%). Thus, the repetitive training increased the subjects' potential to hole the ball and the subjects learned to play more successful. The subsequent analysis describes the progress of kinematic parameters during this learning progress. It reveals changes in putting kinematics during learning with repetitive training.

We analyzed the training progress by selecting the most relevant kinematic parameters with parameter ranking. The resulting parameters showed a high correlation (positive or negative) with the increasing training session number. These findings are an observation of the learning progress that happened according to the putting performance increase between pre and post test. However, our analysis cannot directly link these parameters changes to performance improvement or even skill learning. It rather reveals the most relevant parameter changes during training. The progression of these parameters is discussed in more detail in a separate section below.

The presented system can be used to investigate kinematic changes caused by different training paradigms. Differential learning, an example for the Dynamic Pattern Theory (chapter 4.2), proposes the implementation of multiple task variations during skill acquisition for superior motor learning performance [Scho 00]. To define an appropriate range of variation, most studies align on recommendations of Schöllhorn [Scho 00]. These recommendations are derived mostly from theoretical considerations as well as practical experience and an important question is how variations affect movement processing. See [Prea 13] for an overview of the movement variability and [Beck 10] for an example that analyzes the influence of movement variation on skill learning.

The presented golf putt analysis system can be used for two purposes: to measure task variation [Schm 14b] and to measure the effect of task variation on kinematic parameters [Schm 14a]. Researchers are e.g. interested in the kind of variation that is most beneficial for learning. Therefore, the variation itself and its influence on learning can be investigated by means of the presented system. The presented application can support the development of a more biomechanically founded framework for the optimal range of variation in motor learning. Furthermore, it can be used to measure the progression of kinematic parameters during training to gain a better understanding of the training progress.

Discussion of Parameters

In contrast to the group differences recorded at a single time instance (e.g. [Sim 10]), our analysis revealed trends during training intervention and can therefore not be compared directly. Training progress analysis described a learning path (progress of novices with training) in contrast to the outcome of a learning progress (novices vs. experts). Selected kinematic parameters are now discussed in more detail.

The list of relevant training progress parameters contained two duration parameters (P1, P5). These were the putt duration and the FT duration which were positively correlated with training progress. Thus, subjects tend to putt slower with repetitive training and we speculate that subjects performed a more controlled movement with progressing training. Our findings can be compared to the results for expert-novice-differences regarding timing [Sim 10]. The authors reported that experts spend considerably more time in FT. In agreement, our training progress analysis revealed an increasing absolute FT duration (P5) with training progress for the constant putting distance in the TRAINING study.

The increased duration of the FT was accompanied by an increase in the FT angle (P22) with training progress. Our findings can be compared to the results for expert-novice-differences that revealed a higher horizontal movement amplitude in summed foreswing and follow-through (downswing amplitude, [Dela 97]). The FT angle (P22), a measure related to the downswing amplitude, also increased with the training progress. Thus, our training progress analysis confirms the findings from [Dela 97].

Our training progress analysis revealed decreasing linear velocity on impact (P24) with training progressing. This finding is equal to the results in [Sim 10] where expert and novice differences were analyzed.

The trends that we found with the training progress analysis coincide with the findings for the comparison of experts and novices. Our analysis illustrates how novices improve and we observed the same aspects that differentiate experts from novice players. However, a progress instead of a result is analyzed. Not only can the progress analysis show the trends of kinematic changes, it can also monitor the training progress.

The additional value of the presented analysis is the progress information. For example, the increase in FT duration (P5, Fig 7.3) was discontinued in training session 7 and the decrease in linear impact velocity (P24, Fig 7.5) was discontinued in session 6. The acceleration and deceleration phases are also visible in the FT angle (P22, Fig: 7.4). We speculate that there was either an outlier in the data or a phase in training where learning changes. It can be speculated that the learning focus either shifted to other aspects or that the general training progress decelerated or even ceased. These speculations should be investigated from a dynamical systems perspective, as it indicates that adaptations follow different time scales and task outcomes are products of dynamical stability and instability of different parameters [Newe 01, Scho 09] (see also chapter 4.2.2).

7.5.3 General Aspects

The driving force for the presented golf putt analysis application is the unobtrusive, mobile and automatic character of the proposed system that offers advantages for athletes (feedback training) and researchers (motor learning). Athletes, not distracted by recording equipment or markers, can train in their usual training environment and results are available in real-time. Thus, the presented system covers the requirements for measuring and information systems to support sport performance (chapter 4.3). Additionally, researchers have the possibility to collect a higher amount of data. The presented system automatically logged kinematic parameters without requiring user interaction. Thus, the analysis workload is lower compared to traditional video analysis where videos have to be post processed to deduce kinematic parameters. This facilitates long-term analysis and results can be based on a higher amount of data compared to existing literature. In the study described in [Dela 97] 400 putts were collected and in the study described in [Sim 10] 300 putts were collected. The presented analysis of the TRAINING dataset is based on 3902 putts.

A drawback of the presented system is the missing parameter validation. However, the underlying segmentation model is an established approach to define putt phases and the extraction of many parameters is straightforward [Kooy 13]. Nevertheless, the precision of all kinematic parameters has to be validated in further research.

Depending on the results of the parameter validation, the precision of the system might need further enhancement. It has to be investigated how noise and sensor drift (chapter 3.1.3) influence the parameter precision. This is especially important if the parameter set is enhanced with displacement measures that require two integration steps (chapter 3.1.2). We speculate that precise displacement parameters require more sophisticated movement tracking techniques like the Kalman Filter [Burc 10]. From a data processing perspective, the Kalman Filter might be a valuable extension to enhance the number of kinematic parameters and their precision.

Chapter 8

Swimming Application

This chapter presents the third of the three BSN applications that are described in this thesis. A BSN application for the analysis of head kinematics during swimming that was implemented on a sensor node (chapter 3.3.3) is described. The development of the presented analysis used the methodology to tackle the accuracy-cost tradeoff, best practices to estimate classifier accuracy with feature selection and considerations regarding energy-efficient BSN signal classification introduced in chapters 5.2 to 5.4.

The literature review in chapter 1.2.6 revealed different aspects for athlete support in swimming. An augmented wearable feedback system for swimming would be beneficial for exercise tracking and technique coaching. A promising sensor location to minimize water resistance and maximize user comfort is the head but its potential needs to be further investigated. Besides tracking training statistics, coaching would benefit from assessing the fatigue state of an athlete.

The following section describes the development of a system to analyze head kinematics during swimming for the goal of exercise tracking and fatigue classification. The exercise tracking was implemented and evaluated on the sensor node and is an important building block towards a head worn augmented feedback system for swimming. The system was developed under consideration of the accuracy-cost tradeoff and used an implementation and test environment for the Shimmer sensor nodes that was introduced in [Kugl 13].

This chapter is structured as follows: first, data collection is described. Second, the data analysis pipeline is introduced and the most important aspects of the on-node implementation are summarized. Finally, results are presented and discussed. Parts of this section have previously been published in [Jens 13, Jens 16a].

8.1 Data Collection

8.1.1 Hardware

The Shimmer platform described in chapter 3.3.3 was used for data collection. Nodes were configured to log 3-D acceleration and 3-D angular velocity with a range of $\pm$ 1.5 g (accelerometer) and $\pm$ 500 °/s (gyroscope) and a sampling rate of 204.8 Hz. In general, the sensor nodes were capable of sensing with predefined default sampling rates like 10.24 Hz, 51.2 Hz, 102.4 Hz and 204.8 Hz.

8.1.2 Protocol

11 subjects (4 female, age: 16.1 $\pm$ 1.9 years, height: 178.9 $\pm$ 9.4 cm, weight: 66.3 $\pm$ 11.7 kg) were recruited for the study and signed an informed consent form. The subjects were well trained junior swimmers and competed in the German second league. We recorded 200 m medley sessions that consisted of 50 m of butterfly, backstroke, breaststroke and freestyle swimming respectively. Data were recorded in a 50 m indoor pool. Subjects were asked to maintain a session time of 80% of their maximum performance to be able to compare performance in recovered and fatigued state. We equipped each athlete with a single IMU placed on the back of the head and waterproofed it with a plastic bag.

Each session started inside the pool and on the poolside to avoid the influence of the start phase. Data comprised the start phase, four swimming styles, two turn styles and a resting phase before and after the session (REST). The session was performed in the normal order of butterfly (BUTT), backstroke (BACK), breaststroke (BREA) and freestyle (FREE). All subjects used the front crawl technique in the freestyle part. Subjects were free to perform their preferred turn style so that we recorded different turn styles as well as the start phase at the beginning (TURN). We recorded two sessions for each subject, the first one after warm-up (RECOVERED) and the second one after a 90 min training session (FATIGUED). The recorded sessions were labeled with the different swimming styles, turns and rests by using video recordings that were collected in parallel (Tab. 8.1). Turns, including preparation and wall push-off, were labeled with a fixed interval of four seconds as precise labeling with the recorded video was difficult. Resting phases at the beginning and the end were labeled with a fixed

Session	**Description**	**Label**	**Duration [s]**
Warm-up			
RECOVERED	Waiting at pool side	REST	8
	Start phase	TURN	4
	Butterfly	BUTT	individual
	Tumble or flip turn	TURN	4
	Backstroke	BACK	individual
	Tumble or flip turn	TURN	4
	Breaststroke	BREA	individual
	Tumble or flip turn	TURN	4
	Freestyle	FREE	individual
	Waiting at pool side	REST	8
Training session (90 min)			
FATIGUED	Waiting at pool side	REST	8
	Start phase	TURN	4
	Butterfly	BUTT	individual
	Tumble or flip turn	TURN	4
	Backstroke	BACK	individual
	Tumble or flip turn	TURN	4
	Breaststroke	BREA	individual
	Tumble or flip turn	TURN	4
	Freestyle	FREE	individual
	Waiting at pool side	REST	8
Cool-down			

Table 8.1: Data collection protocol of the swimming study.

interval of eight seconds so that the same amount of data was available for TURN and REST. The remaining phases had individual length according to the athlete's performance. One session (FATIGUED) was excluded for analysis due to sensor malfunction.

Data were split in two disjoint sets for the exercise tracking part. We used a set of ten subjects (both sessions) for algorithm design and selection as well as model training. The remaining subject (one session) was used to test the on-node implementation.

8.2 Data Analysis

The description of the data analysis comprises five sections. First, problem and data are defined for the exercise tracking and fatigue detection part. Second, an overview of the data analysis system is given. Third, the feature extraction algorithms that were used for this application are described. Fourth, the on-node implementation of the exercise tracking part is introduced. Finally, an overview of the experimental setup is given.

8.2.1 Problem and Data Definition

The two data analysis problems of exercise tracking and fatigue detection were approached with pattern classification. The implementation followed the working phase of the classification pipeline (Fig. 2.1). All processing was done in a sliding window approach.

Exercise Tracking

The first aim of the data analysis was swimming exercise tracking implemented on a sensor node. The analysis detected different phases (REST, TURN, BUTT, ...) of a swimming exercise. This phase information can be used to identify swimming style and derive lane count (and therefore distance) as well as split times. A pattern recognition algorithm was developed to discriminate the phases of REST, TURN, BUTT, BACK, BREA and FREE. To analyze kinematic data during a swimming session, the exercise tracking methodology was divided into a two step approach with an algorithm design phase and an on-node implementation of the resulting model.

The goal of the exercise tracking implementation was to run on a sensor node to facilitate real-time processing on a wearable sensor. Therefore, two measures were of interest for the design of the swimming exercise tracking: the estimated accuracy of the classifier and the associated computational cost. The accuracy-cost tradeoff between the algorithms was approached with the method presented in chapter 5.2. The data analysis was performed as single classification problem to show the applicability of the above-mentioned design consideration. The accuracy-cost considerations included the classification system, the window size and the sampling rate. These factors influence accuracy and cost (chapters 5.2 and 5.4).

Fatigue Detection

The second purpose of our analysis was fatigue detection. Therefore, a pattern recognition system was developed to discriminate FATIGUED and RECOVERED sessions using data from the swimming intervals BUTT, BACK, BREA and FREE. The fatigue detection used the same pattern recognition methodology as the exercise tracking to discriminate both conditions. The computational cost and sampling rate were not investigated for on-node implementation, but we investigated different window sizes regarding their influence on classification accuracy.

8.2.2 Data Analysis System

The working phase (chapter 2) of the data analysis system for both problems comprised feature extraction, feature preprocessing and classification. The training phase (chapter 2) comprised feature selection (only exercise tracking) and model evaluation. We compared different algorithms for both data analysis questions (Tab. 8.2). Data was evaluated with leave-one-subject-out cross-validation (chapter 5.2.6).

The feature extraction algorithms as described in chapter 5.2 were computed in one computation step on basis of a complete data interval (or window). On a sensor node, data is sampled and simultaneously processed. Sampling has to be performed on defined time intervals to ensure equidistant samples and therefore a constant sampling rate. We used the update computations of the statistical measures to avoid non-equidistant sampling and therefore ensure the correct timing of computational blocks (sampling, feature extraction, ...).

Extensive computations that take longer than the time between sampling events disrupt equidistant sampling. The algorithms presented in the following section are based on [Knut 98, Peba 08] and are designed to balance the computational load.

Step	**Algorithm**	**Setting**
Feature Extraction	Signal energy	-
	Minimum	-
	Maximum	-
	Mean Value	Update version
	Variance	Update version
	Standard deviation	Update version
	Skewness	Update version
	Kurtosis	Update version
Feature Preprocessing	Normalization	-
Classification	AdaBoost (AB)	Iterations: 50
	Classification via log. regression (CLR)	Attribute selection: none
	Rule-based classification (PART)	-
	Support Vector Machine (SVM)	Kernel: linear, degree: 3, C: 0.5
Feature Selection	Best first, forward (BF-F)	Cross-validation folds: 5, leave-out-subjects: 2, look-up cache: 2, non-improving nodes (termination): 3
	Best first, backward (BF-B)	
Evaluation	Cross-validation	Leave-out-subjects: 1

Table 8.2: Overview of compared exercise tracking data analysis algorithms. Default settings were used if not stated differently.

Therefore, intermediate results are computed in each new sampling value to balance the computational load and avoid a single extensive computational step at the end of an interval (or window). The next sections describe the algorithms in more detail.

8.2.3 Feature Extraction

This section describes the update version of the mean value, variance, standard deviation, skewness and kurtosis feature as motivated in chapter 8.2.2. An update version for the features signal energy, minimum and maximum is straight forward and incorporated the same cost as described in chapter 5. The computational cost and memory demand of the update algorithms is summarized in Tab. 8.3.

Mean Value

The computation of the mean value (M_{update}) was implemented with Alg. 3 which was originally described in [Knut 98]. The computational cost were $2(T-1)$ additions and subtractions as well as $T-1$ divisions. The memory demand was one INT parameter for the interval size T.

Algorithm 3: Computes the mean value in update manner.

$\delta \leftarrow h_t^s$
$\delta_{\text{norm}} \leftarrow \delta$
$M_{\text{update}} \leftarrow \delta_{\text{norm}}$
for $t = 2$ **to** $t = T$ **do**
    $\delta \leftarrow h_t^s - M_{\text{update}}$
    $\delta_{\text{norm}} \leftarrow \delta / t$
    $M_{\text{update}} \leftarrow M_{\text{update}} + \delta_{\text{norm}}$
end

Variance

The computation of the variance (σ^2_{update}) was implemented according to Alg. 4 [Knut 98]. We assumed that $T-1$ was precomputed for the cost analysis. The computational cost was $4(T-1)$ additions and subtractions, $(T-1)+1$ divisions and $T-1$ multiplications. The memory demand was one INT parameter for the interval size T.

Algorithm 4: Computes the variance in update manner.

$\delta \leftarrow h_t^s$
$M_{\text{update}} \leftarrow \delta$
$\sigma^2_{\text{update}} \leftarrow 0$
for $t = 2$ **to** $t = T$ **do**
  $\delta \leftarrow h_t^s - M_{\text{update}}$
  $M_{\text{update}} \leftarrow M_{\text{update}} + \delta / t$
  $\sigma^2_{\text{update}} \leftarrow \sigma^2_{\text{update}} + \delta * (h_t^s - M_{\text{update}})$
end
$\sigma^2_{\text{update}} \leftarrow \sigma^2_{\text{update}} / (T-1)$

Standard Deviation

The standard deviation σ_{update} was computed from the variance σ^2_{update} (Alg. 4). The computational cost were $4(T-1)$ additions and subtractions, $(T-1)+1$ divisions, $T-1$ multiplications and a square root operation. The memory demand was one INT parameter for the interval size T.

Skewness

The skewness (V_{update}, Eq. 8.1) was calculated with

$$V_{\text{update}} = (\sqrt{T} * M3) / \sqrt{M2 * M2 * M2} \tag{8.1}$$

The intermediated results $M2, M3$ were determined with Alg. 5 [Peba 08]. We assumed that the computations can be simplified for $t = 2$ and $t = 3$ due to multiplications by zero and that $T > 2$. The computational cost was:

- Additions/subtractions: $4(T-1) + 9(T-2) + 11(T-3) = 24T - 55$
- Divisions: $(T-1) + (T-2) + (T-3) + 1 = 3T - 5$
- Multiplications: $1 + 7(T-1) + 13(T-2) + 15(T-3) + 3 = 35T - 74$
- Square root operations: 2

The memory demand was one INT parameter for the interval size T.

Kurtosis

The kurtosis (W_{update}, Eq. 8.2) was calculated with

$$W_{\text{update}} = (T * M4)/(M2 * M2) - 3 \tag{8.2}$$

The intermediate results $M2, M4$ were determined with Alg. 5 [Peba 08]. We assumed that the computations can be simplified for $t = 2$ and $t = 3$ due to multiplications with zero and that $T > 2$. The computational cost were:

- Additions/subtractions: $4(T-1) + 9(T-2) + 11(T-3) + 1 = 24T - 54$
- Divisions: $(T-1) + (T-2) + (T-3) + 1 = 3T - 5$
- Multiplications: $1 + 7(T-1) + 13(T-2) + 15(T-3) + 2 = 35T - 75$

The memory demand was one INT parameter for the interval size T.

Algorithm 5: Computes the intermediate results for skewness and kurtosis in update manner.

$n \leftarrow 0$
$\delta \leftarrow h_t^s$
$\delta_{\text{n1}} \leftarrow \delta$
$\delta_{\text{n2}} \leftarrow \delta_{\text{n1}} * \delta_{\text{n1}}$
$\omega \leftarrow 0$
$M1 \leftarrow \delta_{\text{n1}}$
$M2 \leftarrow 0$
$M3 \leftarrow 0$
$M4 \leftarrow 0$
for $t = 2$ **to** $t = T$ **do**
    $n \leftarrow t - 1$
    $\delta \leftarrow h_t^s - M1$
    $\delta_{\text{n1}} \leftarrow \delta / t$
    $\delta_{\text{n2}} \leftarrow \delta_{\text{n1}} * \delta_{\text{n1}}$
    $\omega \leftarrow \delta * \delta_{\text{n1}} * n$
    $M1 \leftarrow M1 + \delta_{\text{n1}}$
    $M4 \leftarrow M4 + \omega * \delta_{\text{n2}} * (t * t - 3 * t + 3) + 6 * \delta_{\text{n2}} * M2 - 4 * \delta_{\text{n1}} * M3$
    $M3 \leftarrow M3 + \omega * \delta_{\text{n1}} * (t - 2) - 3 * \delta_{\text{n1}} * M2$
    $M2 \leftarrow M2 + \omega$
end

Algorithm	Computational Cost						Memory Demand	
	$+,-$	$\times$	$\div$	$<$	$\sqrt{x}$	e^x	INT	FLT
Mean	$2T-2$	–	$T-1$	–	–	–	1	–
Variance	$4T-4$	$T-1$	T	–	–	–	1	–
Standard deviation	$4T-4$	$T-1$	T	–	1	–	1	–
Skewness	$24T-55$	$35T-74$	$3T-5$	–	2	–	1	–
Kurtosis	$24T-54$	$35T-75$	$3T-5$	–	–	–	1	–

Table 8.3: Cost overview for the update version of feature extraction algorithms analog to Tab. 5.1. The variable T denotes the length of the segmented input signal.

8.2.4 On-node Implementation

Implementation and testing on embedded hardware like a BSN sensor node is challenging due to limited simulation, test and debugging capabilities. Recently, a toolchain that supports rapid prototyping and testing of on-node processing algorithms was proposed for the Shimmer platform [Kugl 13].

The most important technical improvements of this toolchain are:

- The porting of the Contiki operating system (Contiki OS, [Dunk 04]) to the Shimmer platform

- The enhancement of the simulation tools Cooja (network level simulation, [Oste 06]) and MSPSim (instruction-set level simulation, [Erik 09]) to support bluetooth radio

- The possibility of simulating prerecorded data

From an application perspective, these enhancements have several benefits. First, they enable the implementation of C-based programs that are capable of running on a Shimmer sensor node and are compatible with the communication protocols of the tools provided by the manufacturer (see chapter 3.3.3). Second, bluetooth connectivity can be simulated and tested (noise, device distance). Third, prerecorded data can be used to develop, debug and test the embedded implementation. Fourth, the energy consumption of an embedded implementation can be estimated. Finally, all these steps can be performed on simulated hardware without the presence of the actual sensor nodes. In consequence, this allows fast development, debugging, testing and simulation of signal processing algorithms for Shimmer sensor hardware.

The limited resources of the sensor node hardware required specific techniques for implementing an on-node classification system. The first aspect, fixed-point arithmetic was introduced in chapter 5.4.3. In the presented on-node implementation, all floating-point numbers were multiplied with 1024, which corresponds to a precision of around three decimal places. As 1024 is a power of two, multiplications and division could be performed as shifts that are known to be faster than multiplications.

The second aspect was offline preprocessing. The classification constants were multiplied offline so that their fixed-point version was already available as constant for the on-node implementation. This aspect was introduced in chapter 5.4.4.

The nature of the selected classifier, SVM, allowed another offline preprocessing step. The third aspect, avoidance of computations, was also introduced in chapter 5.4.4.

8.2.5 Experimental Setup

Exercise Tracking

The experimental setup for the swimming exercise tracking measured every classification pipeline combination varying in classifier and feature selection strategy (Tab. 8.2) as well as the following sampling rates and windows sizes. These experiments were conducted to explore the complete space of possible settings in the accuracy-cost comparison. The influence of classifier (Tab. 5.1), feature selection strategy (chapters 5.2.6 and 5.3), sampling rate (chapter 5.4) and window size (Tab. 5.1) were simultaneously considered.

Data were resampled to lower sampling rates to test the classification accuracy. We used default sampling rates of the Shimmer platform that were 10.24 Hz, 51.2 Hz, 102.4 Hz and 204.8 Hz.

Different window sizes were tried to evaluate classification accuracy and cost. The number of patterns for each class and subjects was balanced so that no class or subjects had considerably more patterns. This balancing was performed to avoid specific classes being preferred by the classifier and due to the fact that different amount of data was recorded for each subject and class (Tab. 8.1). The windowing was a two step process. First, the longest swimming style interval (BUTT, BACK, BREA, FREE) of all subjects was selected. The target pattern number was selected as the number of non-overlapping windows that fit in this selected interval. Second, overlapping windows for all other intervals were chosen so that the target pattern number was achieved for each class. We investigated window sizes between 3.5 s and 1.0 s with increments of 0.5 s. Longer windows were not used to allow the data balancing for the TURN class as described above.

Overall, 288 experiments were conducted (Tab. 8.2). They varied in the classification models (4 variations), sampling rate (4 variations), windows size (6 variations) and feature selection strategy (3 variations). All experiments were conducted with the ECTS that was introduced in chapter 5.2.6.

The target of the analysis was a granularity of 0.5 s. That means that a classification decision was available every 0.5 s in every window size setting. To achieve this granularity, window sizes of above 0.5 s were overlapping so that a classification decision was available every 0.5 s. Due to overlapping, this required performing multiple feature extractions in parallel. We accounted for this computational effort with normalizing the computational cost to one second to make the different window sizes of the experiments comparable.

Fatigue Detection

The experimental setup for the fatigue detection comprised the four classifiers AB, CLR, PART and SVM as described in Tab. 8.2. Furthermore, the window sizes of 2.5 s, 5 s, 10 s and data of the complete lane were investigated. The recorded data was not downsampled and processed as recorded with a sampling rate of 204.8 Hz. Two kinds of experiments were conducted. First, data from all swimming styles were grouped and compared. This experiment targeted the general detection of fatigue not related to the swimming style. Second, each swimming style was investigated individually. This experiment targeted the detection of fatigue during specific swimming styles. In both experiments, two classes were distinguished (FATIGUED, RECOVERED). The complete dataset of 11 subjects was evaluated and all experiments were conducted with the ECTS (chapter 5.2.6).

On-node Implementation

We used the presented toolchain (chapter 8.2.4) to implement, test and validate the on-node processing algorithm for exercise tracking. First, we were interested in the validation of the data analysis model that was trained in the algorithm design phase. Therefore, we used the prerecorded disjoint testing set to assess the classification accuracy of the on-node implementation. This experiment was performed with the simulation toolchain and results were logged. For evaluation, the result of each analysis interval was compared to the majority class of the reference labeling.

Second, we investigated the power consumption to compare on-node processing (ON-NODE) and data streaming (STREAM). We therefore measured the power consumption for both conditions on real sensor hardware with a digital multimeter capable of measuring with 1 Hz. The testing dataset was used during experiments. Measurements were performed for 100 s and the power consumption value of each second was used to calculate the mean power consumption and its standard deviation.

Third, we estimated the power consumption with the simulation toolchain to compare the simulation and the measurement result. As the simulation toolchain did not simulate power consumption directly, we measured the idle and active times of the microcontroller as well as the bluetooth chip. This measurement was performed by instrumenting the code and counting the cycles where the corresponding module was active. Using the power consumption information in the data sheets (Tab. 8.4), we were able to estimate the power consumption of the microcontroller and bluetooth chip for the conditions ON-NODE and STREAM. Further, we measured the BASELINE consumption of real sensor hardware without computing, sampling or radio activity. We assumed that this measurement reflected the power consumption of the peripherals. The BASELINE measurement cannot be estimated with simulation and, therefore, had to be recorded on real sensor hardware. The power consumption for the ON-NODE condition was estimated as

$$I_{\mathrm{O}} = t_{\mathrm{CPU}} \times I_{\mathrm{CPU}} + t_{\mathrm{LPM}} \times I_{\mathrm{LPM}} + I_{\mathrm{BASE}} \tag{8.3}$$

Thereby, t_{CPU} and I_{CPU} denote the time and power consumption in CPU active mode, t_{LPM} and I_{LPM} denote the time and power consumption in CPU low power mode and I_{BASE} denotes the sensor hardware BASELINE consumption. We neglected the effort for transferring results from the on-node condition (e.g. via wireless radio, feedback display, ...).

The power consumption for the STREAM condition was estimated as

$$I_{\mathrm{S}} = t_{\mathrm{CPU}} \times I_{\mathrm{CPU}} + t_{\mathrm{LPM}} \times I_{\mathrm{LPM}} + t_{\mathrm{BT}} \times I_{\mathrm{BT}} + t_{\mathrm{IDLE}} \times I_{\mathrm{IDLE}} + I_{\mathrm{BASE}} \tag{8.4}$$

Thereby, t_{BT} and I_{BT} denote the time and power consumption of the bluetooth chip in active mode, t_{IDLE} and I_{IDLE} denote the time and power consumption of the bluetooth chip in idle mode. Time measurements were performed for 100 s and used to compute the mean power consumption.

Hardware	Mode	Variable	Value [mA]
CPU	Active	I_{CPU}	4.0
	Low power	I_{LPM}	0.075
Bluetooth chip	Active	I_{BT}	40.0
	Idle	I_{IDLE}	25.0

Table 8.4: Power consumption of CPU and bluetooth chip in different modes. Values were the mean of the ranges given in the data sheet reference [Texa 15, Rovi 15].

8.3 Results

The results for the exercise tracking data analysis experiments are presented first. Second, the results of the fatigue detection are shown. Third, we present the results of the on-node implementation evaluation.

8.3.1 Exercise Tracking

This section is divided into the evaluation of the different classifiers, the window size, the sampling rate and overall accuracy-cost tradeoff of all experiments.

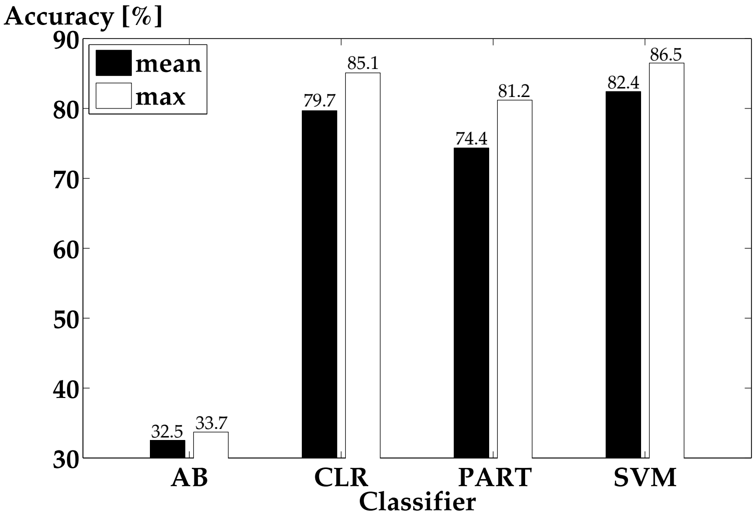

Figure 8.1: Classification accuracy of the swimming phase detection grouped by classifier. The plot shows the mean accuracy of correctly classified analysis windows over all experiments in black and the maximum accuracy in white. Classifiers are abbreviated with adaBoost (AB), classification via logistic regression (CLR) and support vector machine (SVM).

Classifier

We computed the mean and the maximum accuracy for each classifier using all experiments of the specific classifier (Fig 8.1). AB showed the worst performance (mean: 32.5%, maximum 33.7%) and SVM the best performance (mean: 82.4%, maximum 86.5%).

Window Size

We selected the best performing classifier from the classifier evaluation described above, the SVM classifier, to assess the influence of the windows size (Fig. 8.2). For investigating the window size, only experiments with complete feature set were considered. The results for different sampling rates were grouped by window size and averaged. The mean accuracy of the experiments grouped by window size decreased with the window size and achieved performance ranging from 85.3% (3.5 s) to 79.6% (1.0 s) (Fig. 8.2).

Sampling Rate

We selected the best performing classifier from the classifier evaluation described above, the SVM classifier, to assess the influence of the sampling rate (Fig. 8.3). For investigating the sampling rate, only experiments with complete feature set were considered. The results for different window sizes were grouped by sampling rate and averaged. The mean accuracy of the experiments grouped by sampling rate was in the range from 82.4% to 83.0% (Fig. 8.3).

Accuracy-cost Tradeoff

We used the three best performing classifiers (CLR, PART, SVM, see Fig. 8.1) to assess the accuracy-cost tradeoff of each classification experiment (Fig. 8.4). The cost measure of mathematical operations was considered and all operations were summed to the overall number of operations and normalized to one second. The three best results were annotated with cost and accuracy:

- **SVM A**: 10.24 Hz, 3.5 s, BF-B, 85.4% accuracy, 19578 operations/s
- **SVM B**: 51.2 Hz, 3.5 s, BF-F, 86.0% accuracy, 915512 operations/s
- **SVM C**: 204.8 Hz, 3.5 s, BF-F, 86.5% accuracy, 128640 operations/s

The best results from the two remaining classifiers were also annotated with cost and accuracy:

- **CLR**: 204.8 Hz, 3.5 s, BF-F, 85.1% accuracy, 231021 operations/s
- **PART**: 204.8 Hz, 3.5 s, –, 81.2% accuracy, 309507 operations/s

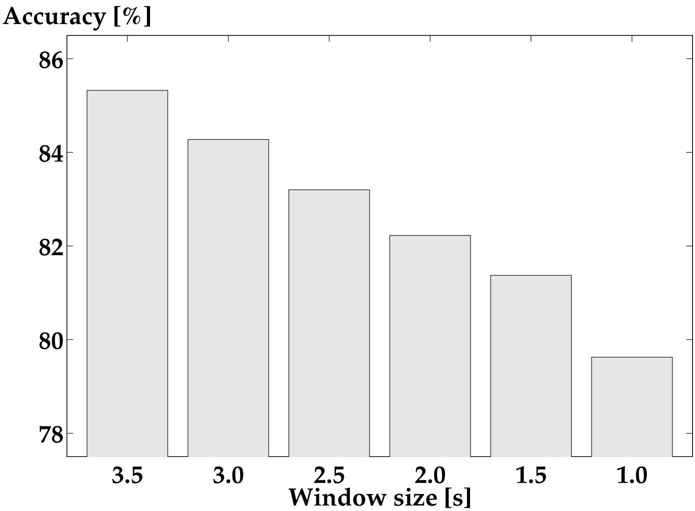

Figure 8.2: Classification accuracy of the support vector machine classifier grouped by window size. Only experiments with complete feature set were considered.

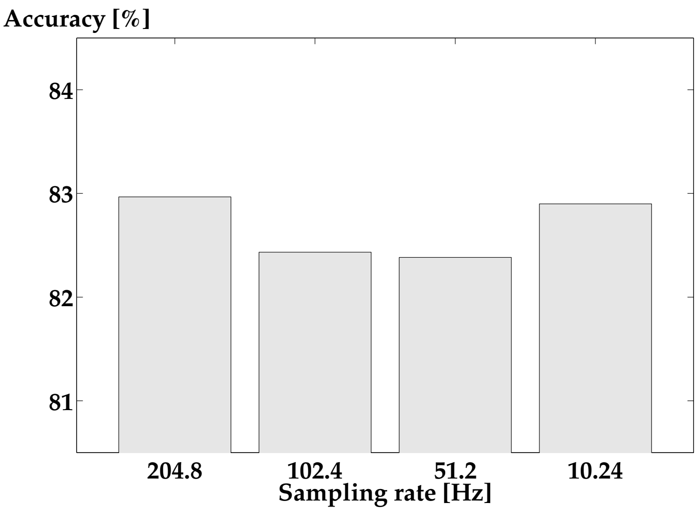

Figure 8.3: Classification accuracy of the support vector machine classifier grouped by sampling rate. Only experiments with complete feature set were considered.

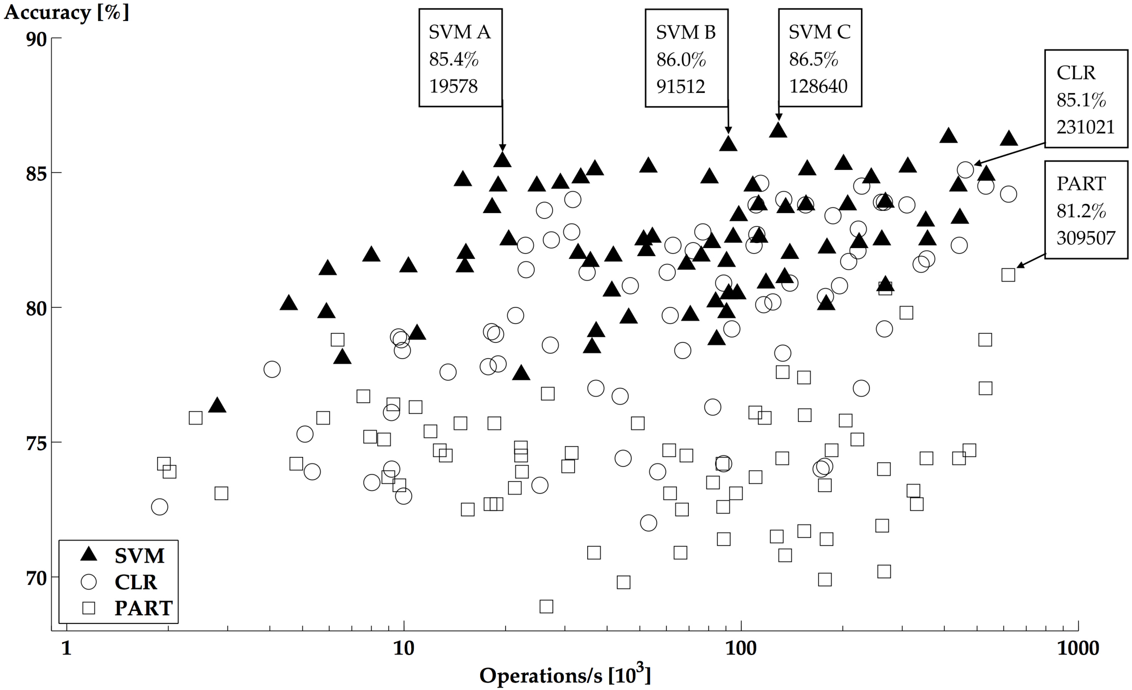

Figure 8.4: Scatter plot of the accuracy and cost of each experiment of the CLR, PART and SVM classifier. The cost axis is shown in logarithmical scale. Selected results are further specified with accuracy and number of operations. Classifiers are abbreviated with adaBoost (AB), classification via logistic regression (CLR) and support vector machine (SVM).

Classifier	Feature Selection	Accuracy [%]	Computational Cost						Memory	
			$+,-$	$\times$	$\div$	$<$	$\sqrt{x}$	e^x	INT	FLT
SVM (C)	BF-F	86.5	36811	17433	7902	2168	6	0	16	289
SVM (B)	BF-F	86.0	20734	20322	3245	1444	11	0	27	467
SVM (A)	BF-B	85.4	4502	4306	706	265	10	0	27	467
CLR	BF-F	85.1	104037	103985	15809	7177	13	0	32	264
PART	–	81.2	141810	137442	21552	8685	18	0	259	191

Table 8.5: Detailed cost analysis of selected exercise tracking experiments. The most promising results of the scatter plot analysis (Fig. 8.4) and the best results of the CLR and the PART classifier are shown. Results are sorted by accuracy. Classifiers are abbreviated with adaBoost (AB), classification via logistic regression (CLR) and support vector machine (SVM).

We compiled detailed results of the best outcomes (SVM A, SVM B, SVM C) and the best-performing results of the CLR and the PART classier (Tab. 8.5). SVM C showed the lowest cost in all operations except for the square root operation. CLR and PART showed the highest cost in all operations. The memory demand was in the same order for all experiments, while PART required the highest amount of integer constants and SVM required the highest number of float constants.

We selected model SVM A for on-node implementation as it used the lowest number of operations and lowest sampling rate.

8.3.2 Fatigue Detection

We compared the proposed classifiers and compiled the highest accuracy throughout experiments for each interval size (Tab. 8.6). The best result for each experiment is shown in bold font and varied between the tested intervals. Classification of an individual swimming style (59.6% - 75.7%) outperformed classification of the complete data set (58.7%).

Experiment	Interval Size			
	2.5 s	5.0 s	10.0 s	Lane
All Styles	58.6	57.7	**58.7**	52.4
BUTT	57.0	54.3	**59.6**	57.1
BACK	**75.7**	70.6	70.8	66.7
BREA	60.5	57.2	**67.1**	52.4
FREE	57.5	58.7	60.0	**71.4**

Table 8.6: Results of the fatigue detection experiments for different interval sizes. The accuracy [%] of the best-performing classifier is shown.

8.3.3 On-node Implementation

The accuracy of disjoint testing set classification was 76.0%.

The measured power consumptions for the different experiments showed that considerably less power is consumed in ON-NODE condition (Tab. 8.7). ON-NODE (13.0 mA) was by far less demanding than STREAM (32.7 mA) and saved more than 60% of power. Subtracting the BASELINE demand, the proportionate saving increased to more than 90%. The values for hardware measurement and simulation estimation are comparable.

Condition	Measurement	Power Consumption [mA]
BASELINE	Hardware	11.8 ± 0.01
ON-NODE	Hardware	13.0 ± 0.55
	Simulation	14.7
STREAM	Hardware	32.7 ± 3.0
	Simulation	37.5

Table 8.7: Results for the power consumption experiments. Hardware measurements [1 Hz] are shown in mean ± standard deviation and simulation results as mean.

8.4 Discussion

8.4.1 Exercise Tracking

The results showed that a reliable classification of swimming styles, turns and breaks using kinematic data collected at the back of the head is possible. However, the presented results investigated data intervals and specific metrics of interest like lane count, split times and swim style detection for a complete lane were not deduced. The accuracy of these metrics has to be investigated in further research. A possible improvement would be the incorporation of context like the sequence of swimming–turn–swimming and the fact that often the same swimming style is used for the complete lane. Furthermore, to enhance the system performance, we propose to use a hierarchical analysis approach with three steps.

1. Discriminate REST, TURN and the combination of all swimming styles

2. Discriminate BACK, FREE and the combination of BREA and BUTT

3. Discriminate BREA and BUTT

This approach would address the need of event detection to deduce lane count and lap count in step 1. Specific algorithms for this event detection (e.g. thresholding) might be more suitable than windowed classification. The fact that BREA and BUTT are confused most often is addressed in steps 2 and 3 and could be tackled with specific feature sets and window sizes when analyzed in a separate classification step. Additionally, the proposed new approach would simplify the current six class problem to easier classification problems with a lower amount of classes.

Results cannot be directly compared with literature as the windowed classification of inertial swimming data was so far only described for the wrist and the back position (Siirtola et al., [Siir 11]). The authors reported an accuracy of 88.9% (wrist) and 95.4% (back) when classifying BACK, BREA, FREE and TURN data with 10 Hz. The classifier comparison for the head kinematics analyzed in this thesis revealed that SVM and CLR are the most promising algorithms for the classification (Fig. 8.1). We believe that the AB classifier did not generalize well and hence achieved the reported low accuracy. Additional AB experiments with less and more iterations (10 to 200 iterations, not shown here) confirmed its overall low performance. Due to the superior performance of the SVM regarding mean and maximum classification rate, this classifier was used to investigate the influence of sampling rate and window size in more detail.

A reduction of the sampling rate had only little influence on the classification accuracy when averaging all experiments of the SVM classifier for one sampling rate (Fig. 8.3). Similar results in the range of below 1% were achieved when comparing the sampling rates of 10.24 Hz, 50.2 Hz and 102.4 Hz. These results show the same trend as the findings for sensor positions on the back and the wrist where similar sampling rates were achieved for sampling with 5 Hz, 10 Hz and 25 Hz [Siir 11].

The results showed better performance for increasing window sizes using the SVM classifier (Fig. 8.2). Comparing the results with recordings on wrist and back, the optimal window size in our experiments (3.5 s, Fig. 8.2) differed from the 2 s value used in [Siir 11]. We speculate that 2 s are a good interval size if swimming strokes are clearly defined (e.g. wrist position). However, the data characteristic is different when collecting head kinematics during swimming. The head kinematics are a combination of swimming stroke movement and breathing stroke movements. Thus, data does not solely reflect the swimming stroke itself, which explains the need of a different data window size. We speculate that longer data windows are needed to increase the amount of information for classification.

We investigated the misclassifications in more detail. Our results revealed that BREA and BUTT were confused most often. These results cannot be compared with prior art as butterfly swimming was not considered in related investigations [Siir 11, Pans 10]. These results can be explained with highly similar head movements due to the nature of these swimming styles. In butterfly and breaststroke swimming, the arms are used simultaneously. Further, swimming and breathing strokes mostly coincide. Despite their similarities, the classifier was able to differentiate these styles with moderate accuracy (BUTT: 75.0%, BREA: 77.7%).

The selection of the most suitable classification model for on-node implementation was an accuracy-cost tradeoff. The optimal region was characterized by high accuracy and a low number of operations per second (Fig. 8.4). We selected three promising candidates from the extensive comparison. They all used the SVM classifier as expected from the mean and maximum classifier comparison. The sampling rate was taken in consideration for the final implementation decision as sampling puts considerable effort on the computational resources. The selected classification model was a SVM classifier with reduced feature set, 10.24 Hz sampling rate and 3.5 s interval size as expected from the previous results.

A drawback of the accuracy-cost comparison was that the number of summed operations was used as cost measure although operations require a different number of machine cycles and therefore different computational runtime. One possibility to overcome this issue would be to transform the mathematical operations to machine cycles of a specific hardware platform. This limits the comparison power to a specific platform but gives a single measure for the computational cost of the algorithm. For other classification problems, this mapping was performed for different microcontrollers that are commonly used in BSN sensor nodes [Jens 16b]. The presented results can be used to compute the overall number of machine cycles needed for classification in further research. This result can be used for classifier selection instead of the summed number of operation.

We compared the cost of selected experiments in more detail using the measures described in chapter 5.2.1. The most promising results of the accuracy-cost comparison and the best-performing pipeline settings for CLR and PART were selected (Fig. 8.4, Tab. 8.5). The detailed analysis confirmed the selection based on the accuracy-cost comparison as the classifier that required the least operations was selected. The SVM (A) experiment had the lowest numbers except for the square root operation. Due to the considerable difference to the other algorithms, we can assume that the actual computational cost (e.g. in machine cycles) is the lowest for the SVM (A) experiment. Although being a demanding operation, the increased number of square root operations (compared to SVM (C)) cannot compensate for the lower amount of the other mathematical operations. The memory demand to save the classification model was in the same order, but highest for SVM (A) and SVM (B). When assuming a demand of 2 Bytes for an integer and 4 Bytes for a float, the model of SVM (A) would require 1.9 KB of memory and the model with the lowest demand would require 1.1 KB. Thus, the difference is very low and will most probably not affect the selection decision. Overall, 10 KB RAM were available on the microcontroller so that the memory demand of the classification system would use a considerable amount of the available RAM.

These results underline the need for a simultaneous and detailed analysis of the computational cost of algorithms in the design phase. The most accurate classifier proved to be a non-optimal choice for on-node implementation as considerably higher costs were estimated to achieve slightly better accuracy. The presented analysis enables a direct comparison to assess this tradeoff. The detailed overview of computational cost revealed the difference in specific operations that can be used to compare algorithms in more detail. This becomes more interesting when comparing specific hardware platforms that feature computing capabilities such as a hardware divider or floating point-unit. However, the presented analysis needs to be enhanced to machine cycles for better comparability with a single measure. It can be concluded that the provided accuracy-cost comparison is a useful tool to support the implementation decision on embedded hardware.

8.4.2 Fatigue Detection

In the context of fatigue detection, we were able to discriminate swimming intervals recorded at the beginning and at the end of a 90 min swimming exercise. In general, the classification results between 58.7% and 75.7% revealed the limited power to distinguish the FATIGUED and RECOVERED condition. The study setup investigated long-term fatigue during a two hour training session. During such a session, the athlete's physiological conditions change. These changes appear in the metabolism as metabolic products (e.g. lactate) accumulate and the energy storage is emptied (chapter 4.1). Further changes appear in chemical muscle cell processes and neuromuscular functioning due to body water and therefore ion loss (chapter 4.1). Additionally, the cognitive load during exercise changes the psychological state of an athlete. Our results imply that these processes influence the head kinematics during swimming. However, the change is only detectable to a limited degree.

The results suggest that fatigue can be detected with higher precision when investigating a swimming style individually. We conclude that the change in head kinematics due to fatigue depends on the swimming style and that it is harder to detect fatigue when combining data from all swimming styles. The different results for the four swimming styles further imply that fatigue influences head kinematics more characteristically in backstroke (BACK) and freestyle (FREE). These findings can be explained with the characteristics of backstroke swimming. The head is not moved for breathing and generally shows less movements than in the other swimming styles. We speculate that fatigue effects body tension and leads to lateral body and head movements as well as an altered floating position. The generic features used for analysis do not directly address these expected fatigue effects. We assume that features that reflect these effects can achieve higher accuracy. Further, we expect an altered stroke characteristic (shorter, less powerful, smaller amplitude) in fatigued state. As there is no overlaying head movement for breathing, these kinematic changes seem to be detectable especially well in backstroke swimming. The high accuracy of the fatigue classification for freestyle might be explained with an interaction of short-term and long-term fatigue. Freestyle is the last swimming style in a medley race. Thus, athletes are fatigued in both scales, long-term (training session) and short-term (medley race). The number of breathing and swimming strokes might be increased due to metabolic changes and the accumulation of lactate. This might lead to a higher amount and maybe also to more uncontrolled head movement than in the RECOVERED condition. It might also be possible that the short-term fatigue effects are more pronounced due to long-term fatigue and make it therefore easier to distinguish the FATIGUED and RECOVERED condition.

A more detailed investigation of the best-performing classifiers revealed that the SVM classifier most often achieved the highest accuracy. Thus, the same classification paradigm is suitable for exercise tracking and fatigue detection. The results imply that, in general, longer intervals are more suitable for the classification of fatigue. This might be due to a higher number of breathing and swimming strokes in one interval but needs to be investigated in more detail. As outlook, we propose to adapt the feature set to the specific kinematic effects of fatigue. Nevertheless, the results are encouraging and it would be interesting to investigate the changes in head kinematics with short-term fatigue. Another interesting aspect would be to investigate physiological and psychological effects separately.

We assume that the exercise tracking and the fatigue detection analysis would benefit from a larger dataset. More data can reveal generic kinematic differences in a more robust way. For fatigue detection, well-trained junior athletes were assessed. It would also be interesting to analyze data from recreational swimmers. We expect that fatigue can be detected with higher accuracy as the underlying affects are more clearly visible for untrained athletes.

8.4.3 On-node Implementation

The accuracy of the disjoint testing set classification showed the correctness of the on-node implementation. It further underlines the applicability of the presented toolchain consisting of ECST for algorithm design and selection as well as Cooja, Contiki and MSPSim for implementation, debugging and test. These tools proved their value for the development of a sophisticated data analysis algorithm for on-node processing in swimming exercise tracking.

The resulting accuracy, however, is below the estimated classification rate during algorithm selection. This might be due to the fact that most of the subjects in the training set were males while the test subject was female. One reason for the lower accuracy might be the fact that females show different head movement characteristics than males. Another reason might be that the implemented classification model does not generalize well. This can be due to a low variation in the training set or due to overtraining on the prevalent movement characteristics. As the number of subjects in the training set is relatively low, additional data might improve the classification accuracy for females. Another reason for the lower accuracy might be the loss of precision due to fixed-point instead of floating-point operations. The ECST internally uses floating-point operations and fixed-point arithmetic would require adapting the algorithms implemented in the ECST. Techniques to ensure that the working phase results of the ECST and the target hardware implementation match need to be investigated in future research. The fixed-point version used a precision of around three decimal places. The precision was not three decimal places as a multiplier of 1024 (instead of 1000) was used to be able to map the multiplication to a computationally cheap shift operation. The described precision caused rounding errors and might therefore be responsible for a reduction of the classification accuracy. For the SVM classifier, specific versions for embedded classification were developed and might be useful for more accurate results, see e.g. [Boni 07].

A general disadvantage of the presented analysis is the high number of classes. Solving a classification problem with six classes is quite complex as the discriminatory power might be too low to separate all classes well. In the case of a SVM, another aspect has to be considered. The one-on-one evaluation with majority vote in the multi-class case can lead to situations where two or more classes share the same amount of votes, which results in an unambiguous class decision. A solution to the presented six class problem could be a hierarchical classifier that separates similar classes from each other in a first step and refines the decision in a second step. This approach was proposed in chapter 8.4.1.

The power consumption experiments on real hardware showed that on-node processing consumed considerably less energy than streaming the data via bluetooth radio. Thus, on-node processing can enable longer node runtimes. However, one important aspect has to be considered. The classification decision might be useful on the sensor node when the decisions are solely logged on the node or if the node has some kind of feedback capability (e.g. LED display, vibration, headphone). If this is not the case, additional effort for transferring the on-node processing result is needed. Depending on the application and real-time requirements, this can be real-time result streaming or infrequent communication. The effort for these kinds of communication schemes has to be estimated for the specific application case.

The simulation experiments showed that it is possible to accurately estimate the power consumption for the target hardware. The estimation via measuring the mode timing for CPU and bluetooth chip proved to be accurate enough to reflect the true power consumption situation. We expect the main source of error in the fact that a range of power consumption values is given in the manufacturers' data sheets. We used the mean value of the given range for our analysis as the true consumption of a specific hardware component could not be measured. We expect that the true power consumption was lower than the value we used for analysis. Thus, the simulation measurement overestimated the power consumption compared to the hardware measurement.

Another possibility to compare algorithms is to compare the time in each of the considered power modes of the CPU and bluetooth chip. As these values are directly measurable, they accurately reflect differences and make algorithms comparable. The main drawback of the simulation results is the fact that the BASELINE power consumption can only be measured. It would be useful to further investigate the main power consumers that are responsible for this value.

Chapter 9

Summary, Discussion and Conclusion

This chapter reviews the results in the context of the complete thesis and contains five parts. First, results from the design considerations (chapters 5) and application examples (chapter 6 to 8) are summarized. Second, the results are subsequently discussed in the context of the complete thesis. Third, the contributions of this thesis (chapter 1.3) are discussed. Fourth, an outlook for future research is given. Fifth, conclusions from the results, discussion and outlook are drawn.

9.1 Summary of Results

This sections summarizes the results of the chapters that contain the design consideration and application examples.

- **Chapter 5** described four considerations for the design of embedded classification systems. The most important aspect was a generic methodology to support the solution of the accuracy-cost tradeoff. This methodology comprised mid-level cost metrics, a cost analysis for popular pattern recognition algorithms and a software package that implements the presented methodology. This software package, the ECST, was used in chapter 8 to design an embedded classification system.

  The chapter further contained a design consideration to accurately estimate the accuracy of classification systems that include a feature selection step. This design consideration was included in the ECST and used throughout the experiments in chapter 8.

In addition, design considerations for energy-efficient embedded classification were presented in chapter 5. The aspects of power saving through hardware low-power mode, sample rate reduction, fixed-point arithmetic as well as processing step avoidance and offline processing were presented. These design considerations were used for the development of the swimming application presented in chapter 8.

The fourth aspect of this chapter was the accurate modeling of HMMs for sequential signal analysis with BSNs. This aspect was used for the plyometric training application presented in chapter 6.

In summary, the results of chapter 5 were design considerations for embedded classification system development.

- **Chapter 6** presented an application to determine the ground contact time in the drop jump exercise of plyometric training. The analysis was based on kinematic data collected on the foot and data were analyzed using the design consideration of accurate HMMs for sequential data analysis. Different populations were analyzed and the results showed a jump detection rate of 92.6% and above as well as an absolute error of 10.8 ms and lower.

- **Chapter 7** described an application for kinematic golf putt analysis. This application shows the opportunities of wearable athlete support systems as it facilitates training in the field, augmented feedback, analysis of a high number of trials and long-term data analysis.

  From a technical perspective, the system provided automated putt detection and real-time parameter extraction with a BSN. As application example, data from a motor learning study (11 subjects) comprising long-term data and a high number of trials were analyzed for training progress.

  The presented system achieved a putt detection rate of over 83% on a disjoint population when subjects followed a basic putt technique. The training progress analysis revealed relevant kinematic parameters and their progression during training intervention.

- **Chapter 8** presented an application for swimming exercise tracking that was implemented as embedded classification system on a BSN sensor node. The aspects of accuracy-cost tradeoff solution, accuracy estimation with feature selection and energy-efficiency were used as design considerations. Chapter 8 also presents an analysis for fatigue detection in swimming which underlines the opportunity of long-term data analysis.

Data from 10 subjects and two conditions (RECOVERED, FATIGUED) were used to train and compare different systems for swimming phase classification. The accuracy-cost reports of the ECST and the energy-efficiency considerations were used to identify the most suitable classification system for implementation. The fatigue detection part investigated different classifiers for their accuracy in discriminating the RECOVERED and FATIGUED condition.

The on-node implementation of the exercise tracking system achieved comparable accuracy as estimated from the experiments and the results were achieved on data of a disjoint subject. The energy consumption of this on-node implementation saved more than 60% of power compared to data streaming. Fatigue could be classified with up to 75.7% accuracy.

9.2 Discussion of Results

The presented BSN applications underline the application opportunities of wearable athlete support systems. Data can be collected in the field where athletes actually perform and where activity is less biased than in a lab environment. We speculate that wearable systems also reduce the influence of the Hawthorne effect [Roet 03] that occurs in scientific data collection studies and states that subjects are biased by the fact that they take part in such a study. Subjects perform in the common athletic environment and without obtrusive equipment. These two aspects might also reduce the Hawthorne effect. The wearable nature of BSN applications was of high importance in all presented applications.

The wearable nature opens new possibilities regarding the amount of data that can be collected. Researchers, athletes and coaches have access to numerous consecutive trials and continuous long-term data for analysis. Processes in contrast to conditions can be analyzed and provide a new and possibly different view on aspects like training control and individualization, fatigue and performance progression. The possibility to collect a high amount of data was of high importance in the golf putt and swimming applications.

The wearable nature also opens new possibilities for augmented feedback applications. If data analysis is performed in real-time and on the wearable system, results are instantly available and can be used as augmented feedback. This thesis demonstrated two possibilities to generate augmented feedback: data analysis on the sensor node and on the processing node. On the one hand, energy-efficient on-node processing can increase the BSN runtime and is a major step to minimize components and therefore enhance the system integration. On the other hand, processing nodes have high computational capabilities for demanding data analysis tasks and often integrate user interaction. The selection of on-node processing or analysis on a processing node is highly application-dependent. The possibility for augmented feedback was of high importance in the golf putt and swimming applications.

Real-time processing on the wearable system and long runtimes of the technical components are challenging demands for BSN applications. System architecture and algorithms need to be carefully designed with special emphasis on energy-efficiency. The presented applications were designed on basis of several design considerations to address these needs. The methodology to tackle the accuracy-cost tradeoff showed its applicability in the swimming application and for other problems of biosignal classification [Jens 16b]. The energy-efficiency considerations and the proposed accuracy estimation in experiments with feature selection were also successfully used to develop the swimming application. The accurate modeling of sequential data classification with HMMs showed its applicability in the plyometric training application. The application examples underline the importance of these design considerations.

This thesis presented BSN applications in the field of sports. However, similar systems are required in related domains like fitness, health and medicine. The applicability of the presented design considerations in medicine was shown with a mobile activity tracker for mental health [Jens 15a]. This application example shows that the requirements for BSN applications in medicine and sports are highly related and that the presented methodology can be transferred to other fields.

The presented design considerations targeted an embedded implementation of a classification system. The ability to run with high accuracy on resource-constrained hardware was the main motivation for solving the accuracy-cost tradeoff. The same methodology with its underlying cost measures can be used for classification tasks where processing speed or real-time processing is the crucial aspect. Using classical pattern recognition problems, processing speed might be crucial for the automatic analysis of handwritten characters as more text can be analyzed in shorter time. Real-time processing might be a crucial requirement for a face detection application in a digital camera to optimize user interaction. In summary, the presented methodology to support the solution of the accuracy-cost tradeoff can in principle be used for all tasks of cost-aware classification system design.

9.3 Discussion of Contributions

The design of athlete support systems involves several technical challenges and their use facilitates several application opportunities. This chapter discusses the contribution of this thesis regarding both aspects. First, the technical challenges in terms of design considerations are reviewed. Second, the application opportunities in terms of application examples are assessed.

9.3.1 Design Considerations

Contribution 1

Wearable athlete support systems demand complex signal processing to present meaningful results to the user. This thesis investigated the field of pattern classification that provides such algorithms and specifically considered their growing application for embedded processing. A considerable challenge in the design of classification systems for embedded implementation is the accuracy-cost tradeoff. The literature review in chapter 1.2.1 revealed that algorithms (e.g. classifiers) are often selected in an early design phase. Thereby, the space of possible algorithms is not completely explored as required according to the No Free Lunch theorem and specific classifiers are preselected without taking into account their classification accuracy and computational cost. Thus, optimal solutions are either excluded from further consideration or identified by implementing all possible solutions. The methodology presented in this thesis (chapter 5.2) provided the first step towards a different approach as all possible solutions for a problem at hand can be quantitatively compared regarding accuracy and cost before implementing them. The presented software package that implemented the methodology (ECST, chapter 5.2.6) showed its applicability in selecting the most suitable classification system for a athlete support system (chapter 8). We believe that the presented approach supports the fast selection of accurate and cost-effective classification systems as all analyzed algorithms can be compared without actually implementing them. This further facilitates the rapid development of wearable athlete support systems on resource-constraint hardware.

Contribution 2

In the design of embedded classification systems, an **accurate estimation of classification accuracy** is needed. Comparing different algorithms is only valuable if the measures reveal the expectable accuracy and true cost. This thesis contributes best practices in accuracy estimation for experiments with cross-validation and feature selection (chapter 5.3). As feature selection can influence the computational cost and estimated classification accuracy, this design consideration is an important aspect in the context of the accuracy-cost tradeoff. Although being important in the context of the thesis, the accurate estimation of classification accuracy in experiments with feature selection is desired in all classification tasks - for instance to quantify the capabilities for group separation or to identify the most relevant features for group separation. A comparison of the correct and incorrect embedding of feature selection in classification with cross-validation revealed considerable differences [Jens 12a]. We believe that this design consideration should be applied in all classification experiments that use cross-validation and feature selection as it facilitates more accurate classification accuracy estimates.

HMMs are a popular methodology to classify sequential data using context information. However, recent inertial data analysis applications like [Dada 13a, Jens 14b] did not exploit the full potential of the HMM framework as observable temporal phases coincided with model states. This thesis contributes a design consideration (chapter 5.5) that described the **accurate modeling of HMMs**. The crucial aspect was the data driven modeling of model states and the fact that observable temporal phases consisted of multiple model states. The results with accurate HMMs presented in this thesis outperformed a similar analysis presented in [Jens 14b]. We believe that the presented accurate modeling of HMMs can increase the analysis performance of inertial data analysis tasks. This improvement further enhances the analysis capabilities of athlete support systems that require sequential data analysis tasks.

Contribution 3

Energy-efficiency is a crucial demand for wearable athlete support systems and thus for their data analysis algorithms. Less energy consumption enables longer runtimes or smaller form factors which are both desirable properties for wearable systems. This thesis contributes design considerations for energy-efficient classification algorithms for BSN sensor nodes (chapter 5.4) to reduce or even omit wireless radio communication. The on-node implementation requires a careful consideration of algorithms but also additional considerations for the specific embedded implementation. The comparison of the on-node data analysis and the traditional data streaming revealed considerably lower energy consumption on the sensor node when data were analyzed locally (chapter 8). We believe that the presented design considerations support the implementation of sophisticated classification algorithms on BSN sensor node hardware. Using on-node processing, smaller or longer running wearable athlete support systems can be developed.

9.3.2 Application Examples

Wearable athlete support systems facilitate new views on athletic performance as additional information for coaches and athletes is instantly accessible while performing in the field. This information can be used for several purposes like increasing training efficiency, injury prevention and understanding crucial aspects of a sport. This thesis contributes three application examples that underline different opportunities of wearable athlete support systems. We believe that the presented applications are an inspirational source for similar systems as the underlying design considerations, analysis algorithms and evaluation methods can be transferred or even reused in another context. The specific applications and their contributions are presented in more details in the next paragraphs.

Contribution 4

Athlete support systems are often used as additional source of information. Athletic parameters might not be directly observable, athletes might not be able to assess all crucial parameters at once or a coach might not always be present for assessment. In these cases, an automatic extraction of parameters using a technical system is beneficial. This thesis contributes an application to assess the crucial parameter of ground contact time (GCT) in the drop jump exercise for plyometric training (chapter 6). The application showed how sequential data can be analyzed to detect jump events and subsequently measure the GCT. The latter can be used to create the correct training setup (e.g. drop height) and maintain the desired training intensity (e.g. shortest possible stretch-shortening cycle). This application might be inspirational for sequential data analysis tasks that involve event detection and temporal phase duration determination.

Contribution 5

Another advantage of automatic athletic parameter extraction is the possibility of long-term progress analysis. Parameters can be tracked over longer periods of time and results can be based on a high number of trials. These opportunities facilitate a different view on athletic performance as progress instead of specific states can be evaluated. This thesis contributes a system for long-term golf putt technique analysis (chapter 7). The system detected golf putts in real-time and automatically extracted kinematic putt parameters. This functionality can be used to create augmented feedback applications. We collected data from a training intervention study with the presented system. These data were used to identify crucial parameters and their progress was analyzed in more detail. These insights can be used to individualize training or test the effectiveness of different training interventions. Thus, this application can be an inspiration for the development of augmented feedback and training progress analysis applications.

Contribution 6

The development of innovative wearable athlete support systems is challenging and gets even more complex if the environment puts an additional demand on the system. The third application example, a swimming exercise tracker, showed how an embedded classification system was realized on unobtrusive off-the-shelf BSN sensor node hardware. This thesis contributes a practical application of the above-mentioned design considerations, a toolchain overview to create such a system and a head-worn swimming exercise and fatigue tracking application (chapter 8). The analysis showed that the presented methodology to tackle the accuracy-cost tradeoff supported the selection of the most suitable classification system. Furthermore, it was shown how swimming exercises can be tracked with software running on a single head-worn BSN sensor node. These results are the first step towards an athlete support system for swimming. More importantly, the design methodology and toolchain can be reused for other systems that analyze kinematic data. The presented application might be an inspiration of how embedded classification can be employed for wearable athlete support systems.

9.4 Outlook

The outlook of this thesis is structured in three parts. First, enhancements of the methodology to tackle the accuracy-cost tradeoff and the ECST are summarized. Second, possible applicational aspects of future BSN applications are described. Third, the value of long-term data for fatigue research is presented.

9.4.1 Accuracy-cost Tradeoff and ECST

The embedded classification design considerations and the ECST can be enhanced in various aspects. More algorithms like additional classifiers and features, but also additional processing steps like signal preprocessing have to be added in further development to make the tool a complete collection of state-of-the-art algorithms. Additionally, the most cost-effective implementations of the classifiers should be considered in the tool as this is the algorithm version that a user is interested in.

The presented methodology provides a mid-level analysis between high-level algorithm complexity in Landau-notation and low-level code instrumentation. It allows a comparison that is independent of specific hardware. However, the cost metrics can be used to deduce the specific number of machine cycles for one specific CPU as shown in [Jens 16b]. This additional step has the advantage that classification systems can be compared with one measure: the number of machine cycles. However, the exact number of cycles highly depends on the actual implementation and compiler. Thus, the number of machine cycles is difficult to predict and cannot yet be used to predict the actual runtime of the algorithm. We therefore propose to enhance the current analysis with extended target-dependent hardware profiles. Using an analysis of the generated assembler instructions, the exact number of cycles could be computed and the cost measures can be further specified to the number of cycles and subsequently to runtime. This extension would allow a precise estimation of machine cycles and runtime for given hardware.

Currently, the presented analysis did not differentiate between integer and floating point operations. This would be a reasonable extension as most microcontrollers do not integrate a floating point processing unit (FPU) and some algorithms could be significantly sped up using only integer operations. Additionally, the range of variables has to be considered and further specified. A variable of low range can be represented with 8 bit, while others need a range of 16 bit. For floating point variables, 32 bit or 64 bit representations can be chosen. In a target-dependent analysis, different datatypes and their corresponding mathematical operations have to be considered.

Another practical add-on is the export of the learned model in pseudocode or C code. This would further speed up the rapid prototyping as the working phase implementation can be directly integrated in the software project. Code blocks for feature extraction and feature normalization are also possible. These enhancements would make the ECST a design tool with integrated code generator.

9.4.2 Integration, Sensors and Human Computer Interface

The presented applications all rely on the mobile nature of BSNs. However, a further miniaturization of components would be beneficial. This can either be achieved with smaller standard components or smart integration for specific applications. Thereby, smart integration often demands a higher level of specialization to dimension hardware for a specific application. The ECST provides usable information to support the design step of such a specific application.

It is also assumed that the proper integration of feedback displays (e.g. screens, speakers) is an important aspect. Thus, enhanced integration of BSN components is strongly related to giving relevant and usable augmented feedback. The results can be used to develop augmented feedback applications, but more research is needed to find innovative solutions. Accordingly, the final feedback systems have to be validated for the value they provide.

One drawback of this thesis is that the value of the feedback for the athlete was not assessed. This has to be addressed in further research. It was e.g. not evaluated whether the display of putt parameters is valuable or how this feedback should be presented. Different aspects like training plan adherence or learning efficiency through augmented feedback are of interest. As described in chapter 4.2.4, several aspects like study design, feedback presentation and feedback frequency need to be considered.

Other types than inertial sensors have been used in BSNs (chapter 3). Different physiological and kinematic signals can be collected. Most importantly, IMUs can be enhanced with a magnetometer to form inertio-magnetic measurement units. Besides, pressure sensing with insoles gained popularity for motion analysis. The presented analysis methodology might be usable for other signal types. However, different features and preprocessing steps are needed as signals contain higher frequencies (e.g. EMG), artifacts (e.g. EEG), outliers (e.g. GPS) or noise (e.g. magnetic field sensing). Additional signals might provide useful information to enhance the presented applications. As an example, the swimming fatigue detection might benefit from an additional ECG monitoring.

Beside the technical miniaturization and integration of other sensor types, innovative concepts for Human Computer Interfaces (HCI) are needed. This is true for common BSN use case, but even more for augmented feedback applications. As summarized in chapter 4.2.4, the timing, content and modality of feedback are very important for optimal results. On the one hand, mobile BSN nodes with feedback capabilities (e.g. Google Glass, Smart Watches) exist and have to be investigated regarding their value for such applications. On the other hand, currently uncommon (e.g. tactile) or previously unused (e.g. gustatory, olfactory) forms of sensing and feedback might be useful for specific applications. These visionary ideas have to be investigated in future research.

9.4.3 Fatigue Research

The long-term nature of the BSN data analysis might be beneficial for the research on different kinds of progresses. Examples are performance progress in motor learning research as well as the disease progress in medical research. To give a specific example in sports, opportunities regarding athlete fatigue are summarized. The presented systems are capable of collecting progress data that might reflect the continuous advancement of fatigue. The results might be beneficial for coaches and athletes in these specific scenarios.

- **Substitution:** Fatigue detection applications might support coaches in deciding for player substitution in team sports as athletes might not able to judge their physiological state in game situations.

- **Training control:** Some training programs demand intensive movements and optimal adaptation is only guaranteed if the athlete is not fatigued. One example is plyometric training but this is also the case for sprinting movements and highly cognitive tasks. Fatigue detection can indicate when such forms of training are beneficial.

- **Injury prevention:** The risk of injury increases in high-intensive training due to advanced loads on muscles and joints and the like. Fatigue detection can be beneficial to avoid dangerous situations in high-intensity training. The benefits in training control and injury detection are strongly coupled.

- **Athlete safety:** Fatigue detection is of great interest in dangerous environments as cold and heat might speed up the evolvement of fatigue (e.g. water loss, cooling) [Busk 96]. Water sports are especially dangerous as athletes might drown when they get tired or unconscious. Fatigue detection might contribute to individual athlete monitors in extreme environments.

9.5 Conclusion

This thesis contributes to the field of wearable athlete support systems. Specifically, the data analysis part of such systems and the use of pattern recognition methods for this purpose were covered. The main aspects were considerations for the design of embedded classification systems and application examples for training in the field. The presented methods, ideas and approaches to design and implement wearable athlete support systems can support researchers and engineers to realize such systems not only in the field of sports but also in similar domains like health, fitness and medicine.

A methodology to support the solution of the accuracy-cost tradeoff in embedded classification system design was presented. The new approach simultaneously compared the classification accuracy and the computational cost, the number and type of mathematical operations and parameters, to support classification system selection. The methodology was implemented in a software package and successfully used to select the most suitable classification system for embedded implementation of a wearable athlete support system. The presented methodology and software are valuable tools for engineers that are confronted with designing embedded classification system in cost-sensitive environments (e.g. wearable, real-time).

Additional design considerations to accurately design and accurately estimate classification systems were presented. These considerations are best practices for increased analysis fidelity and were shown to be beneficial in specific applications. Furthermore, energy-efficiency considerations for the design and implementation of athlete support systems were given. These are general proposals to support the energy-efficient implementation of embedded classification systems. These aspects are worth considering when designing and implementing classification systems on resource-constraint hardware.

The practical value of the technical design considerations manifests in the presented inertial data analysis applications. The plyometric training application and the swimming exercise tracker were specifically based on these considerations. Besides, the application examples showed the different opportunities that wearable athlete support systems provide. These are augmented feedback for athletes and coaches, data analysis in the field and long-term data analysis. This additional value can e.g. be used for enhanced training control, a better understanding of performance through large-scale and long-term data and augmented feedback applications. The presented applications and their application opportunities should be an incitation for other innovative wearable athlete support systems.

The presented methods and applications can easily be transferred to similar domains like health and medicine as inertial sensing is a popular modality for applications like activity recognition [Leut 13] and the rating of gait disorders [Kluc 13]. The presented methods were successfully used for the development of a mobile activity tracker for mental health [Jens 15a]. This application shows that the development of wearable systems for other domains can benefit from the content of this thesis.

Chapter A

Embedded Classification Software Toolbox (ECST) User Interface

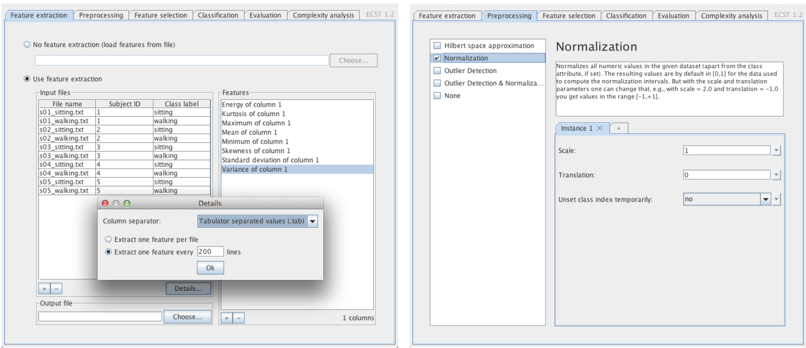

Figure A.1: Screenshots of the feature extraction (left) and feature preprocessing (right) screens of the ECST. On the **feature extraction screen**, data were partitioned (subject, class, interval) and features were specified for extraction. On the **feature preprocessing screen**, one or more algorithms were selected for subsequent execution (left section). In addition, one or more instances of the same algorithm were selected and properties were set (bottom right section). An algorithm description was given on the top right. The screens of the ECST were organized in tabs, see top of each screenshot.

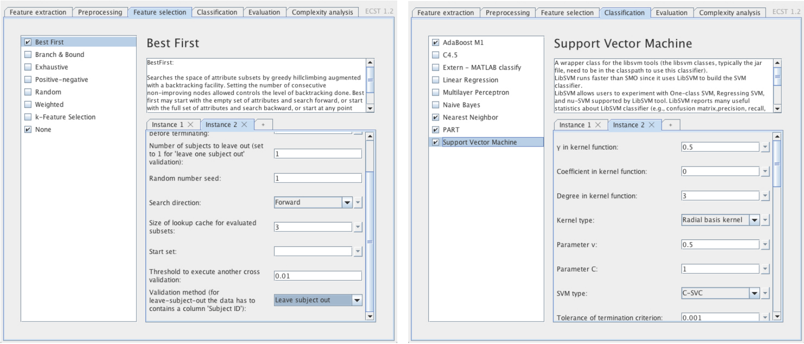

Figure A.2: Screenshots of the feature selection (left) and classification (right) algorithm selection screens of the ECST. On both screens, one or more algorithms were selected for subsequent execution (left section). In addition, one or more instances of the same algorithm were selected and properties were set (bottom right section). An algorithm description was given on the top right.

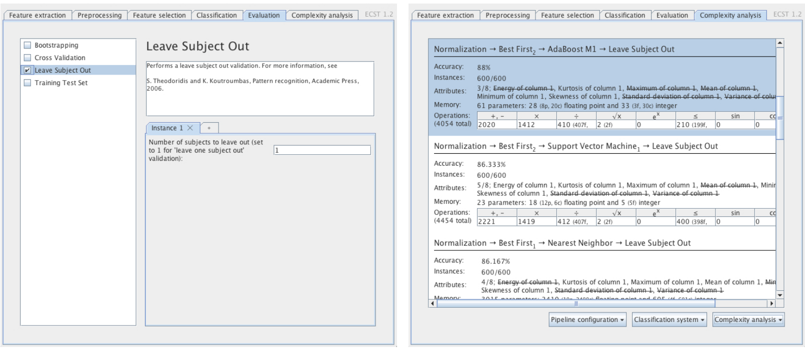

Figure A.3: Screenshots of the evaluation algorithm selection (left) and accuracy-cost report (right) screens of the ECST. On the **evaluation screen**, one or more algorithms were selected for subsequent execution (left section). In addition, one or more instances of the same algorithm were selected and properties were set (bottom right section). An algorithm description was given on the top right. On the **accuracy-cost report screen**, accuracy and cost results for each pipeline configuration were displayed and ordered by accuracy. The number and type of required operations and parameters, the number of patterns (here: instances) and the selected features (here: attributes) were shown. The required operations and parameters (cost measures) were further specified for their occurrence in the feature extraction (f), preprocessing (p) and classification (c) step.

List of Figures

List of Tables

Bibliography

[Aha 91] D. W. Aha, D. Kibler, and M. K. Albert. “Instance-Based Learning Algorithms”. *Machine Learning*, Vol. 6, No. 1, pp. 37–66, 1991.

[Ahma 12] A. Ahmadi, O. Dehzangi, and R. Jafari. “Brain-Computer Interface Signal Processing Algorithms: A Computational Cost vs. Accuracy Analysis for Wearable Computers”. In: *Proc. of the International Conference on Body Sensor Networks (BSN 2012)*, pp. 40–45, London, UK, 2012.

[Akai 73] H. Akaike. “Information Theory and an Extension of the Maximum Likelihood Principle”. In: *Second International Symposium on Information Theory*, pp. 267–281, Budapest, Hungary, 1973.

[Aksa 83] G. Aksamit and W. Husak. “Feedback Influences on the Skill of Putting”. *Perceptual and Motor Skills*, Vol. 56, No. 1, pp. 19–22, 1983.

[Alex 05] D. L. Alexander and W. Kern. “Drive for Show and Putt for Dough?: An Analysis of the Earnings of PGA Tour Golfers”. *Journal of Sports Economics* , Vol. 6, No. 1, pp. 46–60, 2005.

[Angu 11] D. Anguita, L. Carlino, A. Ghio, and S. Ridella. “A FPGA Core Generator for Embedded Classification Systems”. *Journal of Circuits, Systems, and Computers*, Vol. 20, No. 2, pp. 263–282, 2011.

[Annu 12] J. Annuzzi, L. Darcey, and S. Conder. *Introduction to Android™ Application Development: Android Essentials, Fourth Edition.* Addison-Wesley Professional, 2012.

[Baca 09] A. Baca, P. Dabnichki, M. Heller, and P. Kornfeind. “Ubiquitous Computing in Sports: A Review and Analysis”. *Journal of Sports Sciences*, Vol. 27, No. 12, pp. 1335–1346, 2009.

[Bach 12] M. Bächlin and G. Tröster. “Swimming Performance and Technique Evaluation with Wearable Acceleration Sensors”. *Pervasive and Mobile Computing*, Vol. 8, No. 1, pp. 68–81, 2012.

[Bach 94] P. Bachmann. *Die analytische Zahlentheorie. Zahlentheorie: Versuch einer Gesammtdarstellung dieser Wissenschaft in ihren Haupttheilen*, B. G. Teubner, 1894.

[Ball 10] N. B. Ball, C. G. Stock, and J. C. Scurr. “Bilateral Contact Ground Reaction Forces and Contact Times During Plyometric Drop Jumping”. *The Journal of Strength & Conditioning Research*, Vol. 24, No. 10, pp. 2762–2769, 2010.

[Barn 87] V. Barnett and T. Lewis. *Outliers in Statistical Data, 2nd Edition.* John Wiley & Sons, 1987.

[Basm 81] J. Basmajian. “Biofeedback in Rehabilitation: A Review of Principles and Practices”. *Archives of Physical Medicine and Rehabilitation*, Vol. 62, No. 10, pp. 469–475, 1981.

[Bean 14] E. Beanland, L. C. Main, B. Aisbett, P. Gastin, and K. Netto. “Validation of GPS and Accelerometer Technology in Swimming”. *Journal of Science and Medicine in Sport*, Vol. 17, No. 2, pp. 234–238, 2014.

[Beck 10] H. Beckmann, C. Winkel, and W. I. Schöllhorn. “Optimal Range of Variation in Hockey Technique Training”. *International Journal of Sports Psychology*, Vol. 41, No. Special Issue, pp. 5–10, 2010.

[Benb 02] A. Y. Benbasat and J. A. Paradiso. “An Inertial Measurement Framework for Gesture Recognition and Applications”. In: *Gesture and Sign Language in Human-Computer Interaction*, pp. 9–20, Springer, 2002.

[Bern 67] N. Bernstein. *The Coordination and Regulation of Movements.* Pergamon Press, 1967.

[Bero 03] V. Beroulle, Y. Bertrand, L. Latorre, and P. Nouet. “Monolithic Piezoresistive CMOS Magnetic Field Sensors”. *Sensors and Actuators A: Physical*, Vol. 103, No. 1, pp. 23–32, 2003.

[Blum 02] B. Blumenstein, M. Bar-Eli, and G. Tenenbaum, Eds. *Brain and Body in Sport and Exercise.* Wiley, 2002.

[Boni 07] A. Boni, F. Pianegiani, and D. Petri. "Low-Power and Low-Cost Implementation of SVMs for Smart Sensors". *IEEE Transactions on Instrumentation and Measurement*, Vol. 56, No. 1, pp. 39–44, 2007.

[Bosc 06] B. Boscha. "Golf Putter for, System and Method of Training a Golf Player". Feb. 9 2006. US Patent App. 11/236,260.

[Botr 88] N. M. Botros. "A Microprocessor-based Pattern Recognition Algorithm for In-vivo Tissue Differentiation". *Journal of Clinical Engineering*, Vol. 13, No. 2, pp. 115–120, 1988.

[Buci 06] C. Buciluǎ, R. Caruana, and A. Niculescu-Mizil. "Model Compression". In: *Proc. of the ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD 2006)*, pp. 535–541, Philadelphia, USA, 2006.

[Burc 10] R. Burchfield and S. Venkatesan. "A Framework for Golf Training Using Low-Cost Inertial Sensors". In: *Proc. of the International Conference on Body Sensor Networks (BSN 2010)*, pp. 267–272, Singapore, 2010.

[Burn 10] A. Burns, B. R. Green, M. J. McGrath, T. J. O'Shea, B. Kuris, S. M. Ayer, F. Stroiescu, and V. Cionca. "Shimmer™ - A Wireless Sensor Platform for Noninvasive Biomedical Research". *IEEE Sensors Journal*, Vol. 10, No. 9, pp. 1527–34, 2010.

[Busk 96] E. R. Buskirk and S. M. Puhl. *Body Fluid Balance: Exercise and Sport.* CRC Press, 1996.

[Call 09] A. Callaway, J. E. Cobb, and I. Jones. "A Comparison of Video and Accelerometer Based Approaches Applied to Performance Monitoring in Swimming". *International Journal of Sports Science and Coaching*, Vol. 4, No. 1, pp. 139–153, 2009.

[Cebe 10] M. Cebe and C. Gunduz-Demir. "Qualitative test-cost sensitive classification". *Pattern Recognition Letters*, Vol. 31, No. 13, pp. 2043–2051, 2010.

[Chak 13] N. Chakravorti, T. Le Sage, S. E. Slawson, P. P. Conway, and A. A. West. "Design and Implementation of an Integrated

Performance Monitoring Tool for Swimming to Extract Stroke Information at Real Time". *IEEE Transactions on Human-Machine Systems*, Vol. 43, No. 2, pp. 199–213, 2013.

[Cham 15] R. Chambers, T. J. Gabbett, M. H. Cole, and A. Beard. "The Use of Wearable Microsensors to Quantify Sport-Specific Movements". *Sports Medicine*, online preprint, 2015.

[Chan 07] R. Chan. "Swimming Lap Counter". Dec. 20 2007. US Patent App. 11/455,762.

[Chan 11] C. Chang and C. Lin. "LIBSVM: A Library for Support Vector Machines". *ACM Transactions on Intelligent Systems and Technology*, Vol. 2, No. 3, pp. 1–27, 2011.

[Chan 12] M. Chan, D. Estève, J. Fourniols, C. Escriba, and E. Campo. "Smart Wearable Systems: Current Status and Future Challenges". *Artificial Intelligence in Medicine*, Vol. 56, No. 3, pp. 137–156, 2012.

[Chau 95] L. Chau and W. Siu. "Direct Formulation for the Realization of Discrete Cosine Transform Using Recursive Structure". *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, Vol. 42, No. 1, pp. 50–52, 1995.

[Chen 03] X. Chen. "An Improved Branch and Bound Algorithm for Feature Selection". *Pattern Recognition Letters*, Vol. 24, No. 12, pp. 1925–1933, 2003.

[Cho 09] W. J. Cho. "Golf Putter Having a Function of Providing Putting Information". May 26 2009. US Patent 7,537,526.

[Chon 03] C. Chong and S. P. Kumar. "Sensor Networks: Evolution, Opportunities, and Challenges". *Proceedings of the IEEE*, Vol. 91, No. 8, pp. 1247–56, 2003.

[Chri 13] P. Christ, G. Sievers, J. Einhaus, T. Jungeblut, M. Porrmann, and U. Ruckert. "Pareto-optimal Signal Processing on Low-power Microprocessors". In: *Proc. of IEEE Sensors Conference*, pp. 1–4, Baltimore, USA, 2013.

[Ciac 93] E. J. Ciaccio, S. M. Dunn, and M. Akay. "Biosignal Pattern Recognition and Interpretation Systems, Part 2 of 4: Methods for Feature Extraction and Selection". *IEEE Engineering in*

Medicine and Biology Magazine, Vol. 12, No. 4, pp. 106–113, 1993.

[Coel 00] Y. Coello, D. Delay, V. Nougier, and J. P. Orliaguet. "Temporal Control of Impact Movement: The Time from Departure Control Hypothesis in Golf Putting". *International Journal of Sport Psychology*, Vol. 31, No. 1, pp. 24–46, 2000.

[Comm 13] Commons Math Developers. "Apache Commons Math, Version 3.3.0". `http://commons.apache.org`, 2013. [Online; accessed 03-July-2013].

[Corm 01] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. MIT press, 2001.

[Crai 00] C. M. Craig, D. Delay, M. A. Grealy, and D. N. Lee. "Guiding the Swing in Golf Putting". *Nature*, Vol. 405, No. 6784, pp. 295–296, 2000.

[Dada 12] F. Dadashi, F. Crettenand, G. P. Millet, and K. Aminian. "Front-Crawl Instantaneous Velocity Estimation Using a Wearable Inertial Measurement Unit". *Sensors*, Vol. 12, No. 10, pp. 12927–12939, 2012.

[Dada 13a] F. Dadashi, A. Arami, F. Crettenand, G. P. Millet, J. Komar, L. Seifert, and K. Aminian. "A Hidden Markov Model of the Breaststroke Swimming Temporal Phases Using Wearable Inertial Measurement Units". In: *Proc. of the International Conference on Body Sensor Networks (BSN 2013)*, p. no pagination, Cambridge, USA, 2013.

[Dada 13b] F. Dadashi, F. Crettenand, G. P. Millet, L. Seifert, J. Komar, and K. Aminian. "Automatic Front-crawl Temporal Phase Detection Using Adaptive Filtering of Inertial Signals". *Journal of Sports Sciences*, Vol. 31, No. 11, pp. 1251–1260, 2013.

[Dada 14] F. Dadashi, G. P. Millet, and K. Aminian. "Estimation of Front-Crawl Energy Expenditure Using Wearable Inertial Measurement Units". *IEEE Sensors Journal*, Vol. 14, No. 4, pp. 1020–1027, 2014.

[Dave 08] N. Davey, M. Anderson, and D. A. James. "Validation Trial of an Accelerometer-based Sensor Platform for Swimming". *Sports Technology*, Vol. 1, No. 4-5, pp. 202–207, 2008.

[Dela 97] D. Delay, V. Nougier, J.-P. Orliaguet, and Y. Coello. “Movement Control in Golf Putting”. *Human Movement Science*, Vol. 16, No. 5, pp. 597–619, 1997.

[Diet 02] T. G. Dietterich. “Machine Learning for Sequential Data: A Review”. In: *Structural, Syntactic, and Statistical Pattern Recognition*, pp. 15–30, Springer-Verlag, 2002.

[Doga 12] M. Dogan-Aslan, G. F. Nakipoglu-Yüzer, A. Dogan, I. Karabay, and N. Özgirgin. “The Effect of Electromyographic Biofeedback Treatment in Improving Upper Extremity Functioning of Patients with Hemiplegic Stroke”. *Journal of Stroke and Cerebrovascular Diseases*, Vol. 21, No. 3, pp. 187–192, 2012.

[Down 09] D. J. Downey and K. M. Massoth. “Swim Watch”. Dec. 3 2009. US Patent App. 12/129,399.

[Duda 01] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification, Second Edition*. Wiley, 2001.

[Dunk 04] A. Dunkels, B. Gronvall, and T. Voigt. “Contiki - a Lightweight and Flexible Operating System for Tiny Networked Sensors”. In: *Proc. of the IEEE International Conference on Local Computer Networks (LNC 2004)*, pp. 455–462, Tampa, USA, 2004.

[Engl 03] K. Englehart and B. Hudgins. “A Robust, Real-time Control Scheme for Multifunction Myoelectric Control”. *IEEE Transactions on Biomedical Engineering*, Vol. 50, No. 7, pp. 848–854, 2003.

[Erik 09] J. Eriksson, F. Österlind, N. Finne, N. Tsiftes, A. Dunkels, T. Voigt, R. Sauter, and P. J. Marrón. “COOJA/MSPSim: Interoperability Testing for Wireless Sensor Networks”. In: *Proc. of the International Conference on Simulation Tools and Techniques (Simutools 2009)*, pp. 27:1–27:7, Rome, Italy, 2009.

[Esko 09] B. M. Eskofier, M. Oleson, C. DiBenedetto, and J. Hornegger. “Embedded Surface Classification in Digital Sports”. *Pattern Recognition Letters*, Vol. 30, No. 16, pp. 1448–1456, 2009.

[Eyes 06] R. Eyestone, N. Hood, A. Gabbi, J. Farrington, E. Cassady, B. Maloney, R. Deragon, J. Lupher, and J. Satterwhite. “Method and System for Golf Swing Analysis and Training for Putters”. July 27 2006. US Patent App. 11/166,902.

[Ferr 95] F. Ferraris, U. Grimaldi, and M. Parvis. "Procedure for Effortless In-Field Calibration of Three-Axis Rate Gyros and Accelerometers". *Sensors and Materials*, Vol. 7, No. 5, pp. 311–330, 1995.

[Fick 11] T. Fickenscher, J. Holborn, and D. V. Thiel. "Underwater Wireless Optical Communication for Swimmer Feedback Using IrDA Transceiver". *Electronics Letters*, Vol. 47, No. 24, pp. 1335–1336, 2011.

[Fisc 81] M. A. Fischler and R. C. Bolles. "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography". *Communications of the ACM*, Vol. 24, No. 6, pp. 381–395, 1981.

[Fitt 67] P. M. Fitts and M. I. Posner, Eds. *Human Performance*. Brooks/Cole, 1967.

[Fitz 06] K. Fitzpatrick and R. Anderson. "Validation of Accelerometers and Gyroscopes to Provide Real-time Kinematic Data for Golf Analysis". In: E. F. Moritz and S. Haake, Eds., *The Engineering of Sport 6*, pp. 619–630, 2006.

[Flan 08] E. P. Flanagan and T. M. Comyns. "The Use of Contact Time and the Reactive Strength Index to Optimize Fast Stretch-shortening Cycle Training". *Strength and Conditioning Journal*, Vol. 30, No. 5, pp. 32–38, 2008.

[Flec 04] S. J. Fleck and W. J. Kraemer. *Designing Resistance Training Programs*. Human Kinetics, 2004.

[Frad 04] J. Fraden. *Handbook of Modern Sensors - Physics, Designs, and Applications*. Springer, 2004.

[Fran 98] E. Frank and I. Witten. "Generating Accurate Rule Sets Without Global Optimization". In: *Proc. of the Fifteenth International Conference on Machine Learning*, pp. 144–151, Madison, USA, 1998.

[Free 15a] Freescale Semiconductor Inc. "FXAS21002C 3-Axis Digital Angular Rate Gyroscope, Datasheet". `http://www.freescale.com/`, 2015. [Online; accessed 01-March-2015].

[Free 15b] Freescale Semiconductor Inc. "FXLS8471Q 3-Axis Linear Accelerometer, Datasheet". http://www.freescale.com/, 2015. [Online; accessed 01-March-2015].

[Free 15c] Freescale Semiconductor Inc. "FXOS8700CQ 6-Axis Sensor with Integrated Linear Accelerometer and Magnetometer, Datasheet". http://www.freescale.com/, 2015. [Online; accessed 01-March-2015].

[Free 15d] Freescale Semiconductor Inc. "MMA7361L, ±1.5g, ±6g Three Axis Low-g Micromachined Accelerometer, Datasheet". http://www.freescale.com/, 2015. [Online; accessed 01-March-2015].

[Freu 97] Y. Freund and R. E. Schapire. "Decision-theoretic Generalization of On-line Learning and an Application to Boosting". *Journal of Computer and System Sciences*, Vol. 55, No. 1, pp. 119–139, 1997.

[Fult 09a] S. K. Fulton, D. B. Pyne, and B. Burkett. "Quantifying Freestyle Kick-count and Kick-rate Patterns in Paralympic Swimming". *Journal of Sports Sciences*, Vol. 27, No. 13, pp. 1455–1461, 2009.

[Fult 09b] S. K. Fulton, D. B. Pyne, and B. Burkett. "Validity and Reliability of Kick Count and Rate in Freestyle Using Inertial Sensor Technology". *Journal of Sports Sciences*, Vol. 27, No. 10, pp. 1051–1058, 2009.

[Fult 11] S. K. Fulton, D. B. Pyne, and B. Burkett. "Optimizing Kick Rate and Amplitude for Paralympic Swimmers via Net Force Measures". *Journal of Sports Sciences*, Vol. 29, No. 4, pp. 381–387, 2011.

[Gabu 85] B. Gabutti. "An Algorithm for Computing Generalized Euler's Transformations of Series". *Computing*, Vol. 34, No. 2, pp. 107–116, 1985.

[Gans 08] J. G. Ganssle. *Embedded Hardware*. Elsevier, 2008.

[Garc 11] E. Garcia-Breijo, J. Atkinson, L. Gil-Sanchez, R. Masot, J. Ibanez, J. Garrigues, M. Glanc, N. Laguarda-Miro, and C. Olguin. "A Comparison Study of Pattern Recognition Algorithms Implemented on a Microcontroller for Use in an Electronic

Tongue for Monitoring Drinking Waters". *Sensors and Actuators: A Physical*, Vol. 172, No. 2, pp. 570–582, 2011.

[Gent 03] S. Gentil, J. Petersen, P. Roncalez, J. Schlekewey, and M. Wood. "System for Monitoring Repetitive Movement". July 24 2003. US Patent App. 10/056,304.

[Gent 72] A. M. Gentile. "A Working Model of Skill Acquisition with Application to Teaching". *Quest*, Vol. 17, No. 1, pp. 3–23, 1972.

[Gord 98] R. Gordon. "A Calculated Look at Fixed-point Arithmetic". *Embedded Systems Programming*, Vol. 11, No. 4, pp. 72–79, 1998.

[Guyo 03] I. Guyon and A. Elisseeff. "An Introduction to Variable and Feature Selection". *The Journal of Machine Learning Research*, Vol. 3, No. no number, pp. 1157–1182, 2003.

[Gwyn 93] R. G. Gwyn and C. E. Patch. "Comparing Two Putting Styles for Putting Accuracy". *Perceptual and Motor Skills*, Vol. 76, No. 2, pp. 387–390, 1993.

[Hack 06] C. Hacker, A. Batliner, and E. Nöth. "Are You Looking at Me, Are You Talking with Me: Multimodal Classification of the Focus of Attention". In: *Text, Speech and Dialogue*, pp. 581–588, Springer, 2006.

[Hall 09] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. Witten. "The WEKA Data Mining Software: An Update". *SIGKDD Explorations*, Vol. 11, No. 1, pp. 10–18, 2009.

[Hans 09] M. A. Hanson, H. C. Powell, A. T. Barth, and K. Riggenberg. "Body Area Sensor Networks: Challenges and Opportunities". *Computer*, Vol. 42, No. 1, pp. 58–65, 2009.

[Hans 12] M. Hanson, H. C. Powell, A. T. Barth, and J. Lach. "Application-Focused Energy-Fidelity Scalability for Wireless Motion-Based Health Assessment". *ACM Transactions on Embedded Computing Systems*, Vol. 11, No. S2, Article 50, p. no pagination, 2012.

[Hast 08] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning, 2nd Edition.* Springer-Verlag, 2008.

[Hay 06] D. Hay, A. G. de Souza, and S. Fukashiro. "Human Bilateral Deficit During a Dynamic Multi-joint Leg Press Movement". *Human Movement Science*, Vol. 25, No. 2, pp. 181–191, 2006.

[Hell 14] E. Hellman. *Android™ Programming: Pushing the Limits*. Wiley, 2014.

[Hwan 06] S. Hwang and W. Kim. "Fast and Efficient Method for Computing ART". *IEEE Transactions on Image Processing*, Vol. 15, No. 1, pp. 112–117, 2006.

[Ikka 13] A. Ikka and A. Mori. "Image-capturing Apparatus for Putting Practice and Training Putter Having Image-capturing Apparatus". Dec. 17 2013. US Patent 8,608,595.

[Inst 14] Instabeat. "Instabeat Waterproof Heart Rate Monitor". `http://www.instabeat.me`, 2014. [Online; accessed 26-April-2014].

[Inve 15a] InvenSense Inc. "Integrated Dual-Axis Gyro IDG-500, Datasheet". `http://www.invensense.com/`, 2015. [Online; accessed 01-March-2015].

[Inve 15b] InvenSense Inc. "ISZ-500 Single-Axis Z-Gyro Product Specification, Datasheet". `http://www.invensense.com/`, 2015. [Online; accessed 01-March-2015].

[Inve 15c] InvenSense Inc. "ITG-3701 Product Specification Revision 1.0, Datasheet". `http://www.invensense.com/`, 2015. [Online; accessed 01-March-2015].

[Inve 15d] InvenSense Inc. "MPU-6555 Product Specification Revision 1.0, Datasheet". `http://www.invensense.com/`, 2015. [Online; accessed 01-March-2015].

[Inve 15e] InvenSense Inc. "MPU-9255 Product Specification, Datasheet". `http://www.invensense.com/`, 2015. [Online; accessed 01-March-2015].

[Irla 10] J. C. Irlam and L. J. Irlam. "Motion Analysis Device for Sports". Aug. 12 2010. US Patent App. 12/642,830.

[Ishi 08] T. Ishikura. "Reduced Relative Frequency of Knowledge of Results Without Visual Feedback in Learning a Golf-putting

Task”. *Perceptual and Motor Skills*, Vol. 106, No. 1, pp. 225–233, 2008.

[Iswa 06] K. Iswandy and A. Koenig. “Feature Selection with Acquisition Cost for Optimizing Sensor System Design”. *Advances in Radio Science*, Vol. 4, No. 7, pp. 135–141, 2006.

[Jain 00] A. K. Jain, R. P. Duin, and J. Mao. “Statistical Pattern Recognition: A Review”. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 22, No. 1, pp. 4–37, 2000.

[Jain 08] A. Jain and D. Nikovski. “Incremental Exemplar Learning Schemes for Classification on Embedded Devices”. *Machine Learning*, Vol. 72, No. 3, pp. 189–203, 2008.

[Jait 14] T. Jaitner, C. Rheinländer, M. Schmidt, S. Wille, and N. Wehn. “Eine flexible WSN-Plattform zur Leistungsdiagnostik und Trainingssteuerung im Leistung- und Hochleistungssport”. In: *Proc. of 16. Frühjahrsschule “Informations- und Kommunikationstechnologien in der angewandten Trainingswissenschaft” des Instituts für Angewandte Trainingswissenschaft*, pp. 9–20, Leipzig, Germany, 2014.

[Jall 12] P. Jallon. “System and Method for Observing the Swimming Activity of a Person”. March 22 2012. US Patent App. 13/260,510.

[Jame 04] D. A. James, N. Davey, and T. Rice. “An Accelerometer Based Sensor Platform for Insitu Elite Athlete Performance Analysis”. In: *Proc. of the IEEE Conference on Sensors 2004*, pp. 1373–1376, Vienna, Austria, 2004.

[Jame 11] D. A. James, R. I. Leadbetter, M. R. Neeli, B. J. Burkett, D. V. Thiel, and J. B. Lee. “An Integrated Swimming Monitoring System for the Biomechanical Analysis of Swimming Strokes”. *Sports Technology*, Vol. 4, No. 3-4, pp. 141–150, 2011.

[Jens 11] U. Jensen, P. Kugler, F. A. Dassler, and B. M. Eskofier. “Sensor-based Instant Golf Putt Feedback”. In: Y. Jiang and H. Zhang, Eds., *Proc. of the 8th International Symposium on Computer Science in Sport (IACSS 2011)*, pp. 49–53, Shanghai, China, 2011.

[Jens 12a] U. Jensen, F. A. Dassler, and B. M. Eskofier. "Classification of Kinematic Golf Putt Data with Emphasis on Feature Selection". In: *Proc. of the International Conference on Pattern Recognition (ICPR 2012)*, pp. 1735–1738, Tsukuba Science City, Japan, 2012.

[Jens 12b] U. Jensen, M. Ring, and B. M. Eskofier. "Generic Features for Biosignal Classification". In: R. Byshko, T. Dahmen, M. Gratkowski, M. Gruber, J. Quintana, D. Saupe, M. Vieten, and A. Woll, Eds., *Proc. of the Sportinformatik 2012*, pp. 162–168, Konstanz, Germany, 2012.

[Jens 13] U. Jensen, F. Prade, and B. M. Eskofier. "Classification of Kinematic Swimming Data with Emphasis on Resource Consumption". In: *Proc. of the International Conference on Body Sensor Networks (BSN 2013)*, p. no pagination, Cambridge, USA, 2013.

[Jens 14a] U. Jensen, F. A. Dassler, M. Schmidt, M. Hennig, T. Jaitner, and B. M. Eskofier. "A Mobile System to Investigate Putting Kinematics in Motor Learning". In: A. D. Haan, C. J. D. Ruiter, and E. Tsolakidis, Eds., *Book of Abstracts of the 19th Annual Congress of the European College of Sport Science*, pp. 207–208, Amsterdam, The Netherlands, 2014.

[Jens 14b] U. Jensen, A. Heinrich, and B. M. Eskofier. "A Signal-Shift Boosted Hidden Markov Model for Plyometric Training". In: International Society of Biomechanics, Ed., *Proc. of the 13th International Symposium on 3D Analysis of Human Movement (3D-AHM 2014)*, pp. 311–314, Lausanne, Switzerland, 2014.

[Jens 15a] U. Jensen, H. Leutheuser, S. Hofmann, B. Schuepferling, G. Suttner, K. Seiler, J. Kornhuber, and B. M. Eskofier. "A Wearable Real-time Activity Tracker". *Biomedical Engineering Letters*, Vol. 5, No. 2, pp. 147–157, 2015.

[Jens 15b] U. Jensen, M. Schmidt, M. Hennig, F. A. Dassler, T. Jaitner, and B. M. Eskofier. "An IMU-based Mobile System for Golf Putt Analysis". *Sports Engineering*, Vol. 18, No. 2, pp. 123–133, 2015.

[Jens 16a] U. Jensen, P. Blank, P. Kugler, and B. M. Eskofier. "Unobtrusive and Energy-efficient Swimming Exercise Tracking Using On-node Processing". *IEEE Sensors Journal*, Vol. 16, No. 10, 2016.

[Jens 16b] U. Jensen, P. Kugler, M. Ring, and B. M. Eskofier. "Approaching the accuracy-cost conflict in embedded classification system design". *Pattern Analysis and Applications*, Vol. 19, No. 3, pp. 839–855, 2016.

[Ji 07] S. Ji and L. Carin. "Cost-sensitive Feature Acquisition and Classification". *Pattern Recognition*, Vol. 40, No. 5, pp. 1474–1485, 2007.

[Jone 13] T. B. Jones and N. G. Nenadic. *Electromechanics and MEMS*. Cambridge University Press, 2013.

[Kapo 99] I. Kaporin. "A Practical Algorithm for Faster Matrix Multiplication". *Numerical Linear Algebra with Applications*, Vol. 6, No. 8, pp. 687–700, 1999.

[Karl 10] J. Karlsen. *Performance in Golf Putting*. PhD thesis, Norwegian School of Sport Sciences, 2010.

[Kerr 88] D. D. Kerrick and A. C. Bovik. "Microprocessor-based Recognition of Handprinted Characters From a Tablet Input". *Pattern Recognition*, Vol. 21, No. 5, pp. 525–537, 1988.

[King 08] K. King, S. W. Yoon, N. C. Perkins, and K. Najafi. "Wireless MEMS Inertial Sensor System for Golf Swing Dynamics". *Sensors and Actuators: A Physical*, Vol. 141, No. 2, pp. 619–630, 2008.

[Kluc 13] J. Klucken, J. Barth, P. Kugler, J. Schlachetzki, T. Henze, F. Marxreiter, Z. Kohl, R. Steidl, J. Hornegger, B. M. Eskofier, and J. Winkler. "Unbiased and Mobile Gait Analysis Detects Motor Impairment in Parkinson's Disease". *PLoS ONE*, Vol. 8, No. 2, 2013.

[Knut 98] D. E. Knuth. *The Art of Computer Programming, Volume 2: Seminumerical Algorithms, 3rd Edition*. Addison-Wesley, 1998.

[Kooy 13] D. J. Kooyman, D. A. James, and D. D. Rowlands. "A Feedback System for the Motor Learning of Skills in Golf". *Procedia Engineering*, Vol. 60, No. no number, pp. 226–231, 2013.

[Kraf 13] M. Kraft and N. M. White, Eds. *MEMS for Automotive and Aerospace Applications*. Woodhead Publishing, 2013.

[Kres 98] U. Kressel. "Pairwise Classification and Support Vector Machines". In: B. Schölkopf, C. Burges, and A. Smola, Eds., *Advances in Kernel Methods*, pp. 255–268, MIT Press, 1998.

[Kugl 11] P. Kugler, D. Schuldhaus, U. Jensen, and B. M. Eskofier. "Mobile Recording System for Sport Applications". In: Y. Jiang and H. Zhang, Eds., *Proc. of the 8th International Symposium on Computer Science in Sport (IACSS 2011)*, pp. 67–70, Shanghai, China, 2011.

[Kugl 13] P. Kugler, P. Nordhus, and B. M. Eskofier. "Shimmer, Cooja and Contiki: A New Toolset for the Simulation of On-node Signal Processing Algorithms". In: *Proc. of the International Conference on Body Sensor Networks (BSN 2013)*, p. no pagination, Cambridge, USA, 2013.

[Kuip 02] J. B. Kuipers. *Quaternions and Rotation Sequences.* Princeton University Press, 2002.

[Laur 00] J. Laurikkala, M. Juhola, and E. Kentala. "Informal Identification of Outliers in Medical Data". In: *Proc. of the 5th International Workshop on Intelligent Data Analysis in Medicine and Pharmacology*, pp. 20–24, Berlin, Germany, 2000.

[Le S 11] T. Le Sage, A. Bindel, P. Conway, L. Justham, S. Slawson, and A. West. "Embedded Programming and Real-time Signal Processing of Swimming Strokes". *Sports Engineering*, Vol. 14, No. 1, pp. 1–14, 2011.

[Le S 12] T. Le Sage, A. Bindel, P. P. Conway, L. M. Justham, S. E. Slawson, J. Webstera, and A. A. West. "A Multi-sensor System for Monitoring the Performance of Elite Swimmers". In: M. S. Obaidat, G. A. Tsihrintzis, and J. Filipe, Eds., *E-Business and Telecommunications*, pp. 350–362, Springer Berlin Heidelberg, 2012.

[Lee 08a] E. A. Lee. "Cyber Physical Systems: Design Challenges". In: *Proc. of the IEEE International Symposium on Object Oriented Real-Time Distributed Computing (ISORC 2008)*, pp. 363–369, Orlando, USA, 2008.

[Lee 08b] T. D. Lee, T. Ishikura, S. Kegel, D. Gonzalez, and S. Passmore. "Head-putter Coordination Patterns in Expert and Less Skilled

Golfers". *Journal of Motor Behavior*, Vol. 40, No. 4, pp. 267–272, 2008.

[Lee 11] J. B. Lee, B. J. Burkett, D. V. Thiel, and D. A. James. "Inertial Sensor, 3D and 2D Assessment of Stroke Phases in Freestyle Swimming". *Procedia Engineering*, Vol. 13, No. no number, pp. 148–153, 2011.

[Lee 12] J. B. Lee, Y. Ohgi, and D. A. James. "Sensor Fusion: Let's Enhance the Performance of Performance Enhancement". *Procedia Engineering*, Vol. 34, No. no number, pp. 795–800, 2012.

[Lenz 06] J. Lenz and A. S. Edelstein. "Magnetic Sensors and Their Applications". *IEEE Sensors Journal*, Vol. 6, No. 3, pp. 631–649, 2006.

[Leun 06] K. K. Leung and S. N. Chan. "Swim Stroke Monitor". Feb. 8 2006. EP Patent App. EP20,050,254,561.

[Leut 13] H. Leutheuser, D. Schuldhaus, and B. M. Eskofier. "Hierarchical, Multi-Sensor Based Classification of Daily Life Activities: Comparison with State-of-the-Art Algorithms Using a Benchmark Dataset". *PLos ONE*, Vol. 8, No. 10, 2013.

[Lieb 02] D. G. Liebermann, L. Katz, M. D. Hughes, R. M. Bartlett, J. McClements, and I. M. Franks. "Advances in the Application of Information Technology to Sport Performance". *Journal of Sports Sciences*, Vol. 20, No. 10, pp. 755–769, 2002.

[Ligh 80] L. H. Light, G. E. McLellan, and L. Klenerman. "Skeletal Transients on Heel Strike in Normal Walking with Different Footwear". *Journal of Biomechanics*, Vol. 13, No. 6, pp. 477 – 480, 1980.

[Lin 08] C. T. Lin, Y. C. Chen, T. Y. Huang, T. T. Chiu, L. W. Ko, S. F. Liang, H. Y. Hsieh, S. H. Hsu, and J. R. Duann. "Development of Wireless Brain Computer Interface With Embedded Multitask Scheduling and its Application on Real-Time Driver's Drowsiness Detection and Warning". *IEEE Transactions on Biomedical Engineering*, Vol. 55, No. 5, pp. 1582–1591, 2008.

[Luin 05] H. J. Luinge and P. H. Veltink. "Measuring Orientation of Human Body Segments Using Miniature Gyroscopes and Accelerometers". *Medical and Biological Engineering and Computing*, Vol. 43, No. 2, pp. 273–282, 2005.

[MacI 08] C. MacIntosh, T. Rice, S. Holthouse, and I. Van De Greindt. "Sports Sensor". Nov. 20 2008. US Patent App. 12/121,960.

[Mack 08] C. Mackintosh, D. James, N. Davey, R. Grenfell, and K. Zhang. "Monitoring Sports and Swimming". Jan. 24 2008. US Patent App. 11/843,204.

[MacK 10] S. J. MacKenzie and D. B. Evans. "Validity and Reliability of a New Method for Measuring Putting Stroke Kinematics Using the TOMI® System". *Journal of Sports Sciences*, Vol. 28, No. 8, pp. 891–899, 2010.

[Magi 11] R. A. Magill. *Motor Learning and Control: Concepts and Applications, Ninth Edition*. McGraw-Hill, 2011.

[Maka 10] H. Makaruk and T. Sacewicz. "Effects of Plyometric Training on Maximal Power Output and Jumping Ability". *Human Movement*, Vol. 11, No. 1, pp. 17–22, 2010.

[Marq 07] C. Marquardt. "The SAM PuttLab: Concept and PGA Tour Data". *International Journal of Sports Science and Coaching*, Vol. 2, Supplement 1, No. no number, pp. 101–120, 2007.

[Marw 03] P. Marwedel. *Embedded System Design*. Kluwer Academic Publisher, 2003.

[Marw 10] P. Marwedel. "Embedded and Cyber-Physical Systems in a Nutshell". In: *Proc. of the Design Automation Conference*, p. no pagination, Austin, USA, 2010.

[Mazi 14] S. Mazilu, U. Blanke, M. Hardegger, G. Tröster, E. Gazit, M. Dorfman, and J. M. Hausdorff. "GaitAssist: A Wearable Assistant for Gait Training and Rehabilitation in Parkinson's Disease". In: *IEEE International Conference on Pervasive Computing and Communications (PerCom 2014)*, pp. 135–137, Budapest, Hungary, 2014.

[McCl 03] D. McClymont. "Use of the Reactive Strength Index (RSI) as an Indicator of Plyometric Training Conditions". In: *Science and Football V: The Proceedings of the Fifth World Congress on Sports Science and Football*, pp. 408–416, Lisbon, Portugal, 2003.

[McLa 08] P. McLaughlin. *Kinematic and Centre of Pressure (COP) Parameters in Golf Putting*. PhD thesis, Victoria University, 2008.

[McLa 13] P. McLaughlin and R. Best. "Taxonomy of Golf Putting: Do Different Golf Putting Techniques Exist?". *Journal of Sports Sciences*, Vol. 31, No. 10, pp. 1038–1044, 2013.

[Mend 92] L. Mendoza. *Biomechanisch gestützte Techniksteuerung im Hochleistungssport: Exemplarische Darstellung und Analyse von Problemen, Methoden und Resultaten bei der Konstruktion und Anwendung eines biomechanischen Schnellinformationssystems*. PhD thesis, Goethe-Universität Frankfurt am Main, 1992.

[Muel 12] N. Mueller and K. Doerksen. "System and Method for Counting Swimming Laps". Sep. 27 2012. US Patent App. 12/837,920.

[Mynh 00] G. S. Mynhardt. "Apparatus for Determining When a Swimmer Turns". Nov. 16 2000. WO Patent App. PCT/IB2000/000,579.

[Mysl 88] J. Mysliwiec and T. J. Ellis. "Swimming Goggles Including a Timing Device". Oct. 11 1988. US Patent 4,776,045.

[Naka 10] M. Nakashima, Y. Ohgi, E. Akiyama, and N. Kazami. "Development of a Swimming Motion Display System for Athlete Swimmers' Training Using a Wristwatch-Style Acceleration and Gyroscopic Sensor Device". *Procedia Engineering*, Vol. 2, No. 2, pp. 3035–3040, 2010.

[Nest 08] Y. Nestoriuc, A. Martin, W. Rief, and F. Andrasik. "Biofeedback Treatment for Headache Disorders: A Comprehensive Efficacy Review". *Applied Psychophysiology and Biofeedback*, Vol. 33, No. 3, pp. 125–140, 2008.

[Newe 01] K. M. Newell, Y. T. Liu, and G. Mayer-Kress. "Time Scales in Motor Learning and Development". *Psychological Review*, Vol. 108, No. 2, pp. 57–82, 2001.

[Niem 90] H. Niemann. *Pattern Analysis and Understanding, Second Edition*. Springer-Verlag, 1990.

[Nigg 07] B. M. Nigg and W. Herzog. *Biomechanics of the Musculoskeletal System, Third Edition*. Wiley, 2007.

[Ohgi 02a] Y. Ohgi. "Microcomputer-based Acceleration Sensor Device for Sports Biomechanics". In: *Proc. of the IEEE Sensors Conference*, pp. 699–704, Orlando, USA, 2002.

[Ohgi 02b] Y. Ohgi, H. Ichikawa, and C. Miyaji. "Microcomputer-based Acceleration Sensor Device for Swimming Stroke Monitoring". *JSME International Journal Series C*, Vol. 45, No. 4, pp. 960–966, 2002.

[Ohgi 02c] Y. Ohgi, M. Yasumura, H. Ichikawa, and C. Miyaji. "Analysis of Stroke Technique Using Acceleration Sensor IC in Freestyle Swimming". *Engineering of Sport*, Vol. 7, No. no number, pp. 503–511, 2002.

[Ohgi 03] Y. Ohgi, H. Ichikawa, M. Homma, and C. Miyaji. "Stroke Phase Discrimination in Breaststroke Swimming Using a Tri-axial Acceleration Sensor Device". *Sports Engineering*, Vol. 6, No. 2, pp. 113–123, 2003.

[Oste 06] F. Osterlind, A. Dunkels, J. Eriksson, N. Finne, and T. Voigt. "Cross-Level Sensor Network Simulation with COOJA". In: *Proc. of the IEEE International Conference on Local Computer Networks (LNC 2006)*, pp. 641–648, Tampa, USA, 2006.

[Otto 06] C. Otto, A. Milenkovic, C. Sanders, and E. Jovanov. "System Architecture of a Wireless Body Area Sensor Network for Ubiquitous Health Monitoring". *Journal of Mobile Multimedia*, Vol. 1, No. 4, pp. 307–326, 2006.

[Pacl 02] P. Paclík, R. P. Duin, G. M. van Kempen, and R. Kohlus. "On Feature Selection with Measurement Cost and Grouped Features". In: *Structural, Syntactic, and Statistical Pattern Recognition*, pp. 461–469, Springer, 2002.

[Pans 10] J. Pansiot, B. Lo, and G.-Z. Yang. "Swimming Stroke Kinematic Analysis with BSN". In: *Proc. of the International Conference on Body Sensor Networks (BSN 2010)*, pp. 153–158, Singapore, 2010.

[Para 00] G. Paradisis and J. Rees. "Kinematic Analysis of Golf Putting for Expert and Novice Golfers". In: *Proc. of the International Symposium on Biomechanics in Sports (ISBS 2000)*, pp. 325–328, HongKong, 2000.

[Para 05] R. Paradiso, G. Loriga, and N. Taccini. "A Wearable Health Care System Based on Knitted Integrated Sensors". *IEEE Transactions on Information Technology in Biomedicine*, Vol. 9, No. 3, pp. 337–344, 2005.

[Patt 10] M. Patterson and B. Caulfield. "A Method for Monitoring Reactive Strength Index". *Procedia Engineering*, Vol. 2, No. 2, pp. 3115–3120, 2010.

[Peba 08] P. Pébay. "Formulas for Robust, One-Pass Parallel Computation of Covariances and Arbitrary-Order Statistical Moments". Tech. Rep. SAND2008-6212, Sandia National Laboratories, Livermore, USA, September 2008.

[Pede 06] R. Pedersen and M. Schoeberl. "An Embedded Support Vector Machine". In: *Proc. of the International Workshop on Intelligent Solutions in Embedded Systems*, pp. 1–11, Vienna, Austria, 2006.

[Perl 03] M. Perlmutter. "Methods and Systems for Analyzing the Motion of Sporting Equipment". Nov. 6 2003. US Patent App. 09/966,010.

[Poli 07] R. Polikar. "Bootstrap-inspired Techniques in Computational Intelligence: Ensemble of Classifiers, Incremental Learning, Data Fusion and Missing Features". *IEEE Signal Process Mag*, Vol. 24, No. 4, pp. 59–72, 2007.

[Prea 13] E. Preatoni, J. Hamill, A. J. Harrison, K. Hayes, R. E. Van Emmerik, C. Wilson, and R. Rodano. "Movement Variability and Skills Monitoring in Sports". *Sports Biomechanics*, Vol. 12, No. 2, pp. 69–92, 2013.

[Pusc 00] P. Puschner and A. Burns. "A Review of Worst-Case Execution-Time Analysis". *The International Journal of Time-Critical Computing Systems*, Vol. 18, No. 2, pp. 115–128, 2000.

[Quin 93] R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, 1993.

[Rabi 85] J. Rabinowitz. "Lap Counter". July 16 1985. US Patent 4,530,105.

[Rabi 89] L. R. Rabiner. "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition". *Proceedings of the IEEE*, Vol. 77, No. 2, pp. 257–286, 1989.

[Rahm 04] A. S. Rahman, M. M. Yap, A. Y. Shakaff, M. N. Ahmad, Z. Dahari, Z. Ismail, and M. S. Hitam. "A Microcontroller-based Taste Sensing System for the Verification of Eurycoma Longifolia". *Sensors and Actuators: B Chemical*, Vol. 101, No. 1-2, pp. 191–198, 2004.

[Roet 03] F. J. Roethlisberger and W. J. Dickson. *Management and the Worker*. Vol. 5, Psychology Press, 2003.

[Rovi 15] Roving Networks. "RN-42 data sheet". `https://www.sparkfun.com/datasheets/Wireless/Bluetooth/rn-42-ds.pdf`, 2015. [Online; accessed 02-January-2015].

[Scha 90] R. E. Schapire. "The Strength of Weak Learnability". *Machine Learning*, Vol. 5, No. 2, pp. 197–227, 1990.

[Schm 08] R. A. Schmidt and C. A. Wrisberg. *Motor Learning and Performance*. Human Kinetics, 2008.

[Schm 14a] M. Schmidt, M. Hennig, and T. Jaitner. "Different Levels of Task Variation in Learning a Golf Putting Skill". In: N. Balagué, C. Torrents, A. Vilanova, J. Cadefau, R. Tarragó, and E. Tsolakidis, Eds., *Book of Abstracts of the 18th Annual Congress of the European College of Sport Science*, p. 304, Barcelona, Spain, 2014.

[Schm 14b] M. Schmidt, M. Hennig, U. Jensen, B. M. Eskofier, and T. Jaitner. "Short Time Effects of Task Variation on Club Head Kinematics in Golf Putting". In: A. D. Haan, C. J. D. Ruiter, and E. Tsolakidis, Eds., *Book of Abstracts of the 19th Annual Congress of the European College of Sport Science*, Amsterdam, The Netherlands, 2014.

[Schm 92] D. Schmidtbleicher. *The Encyclopedia of Sports Medicine. Vol 3: Strength and Power in Sport*, Chap. Training for power events, pp. 169–179. Wiley-Blackwell, 1992.

[Scho 00] W. I. Schöllhorn. "Applications of Systems Dynamic Principles to Technique and Strength Training". *Acta Academiae Olympique Estonia*, Vol. 1, No. 8, pp. 67–85, 2000.

[Scho 01] B. Schölkopf and A. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, 2001.

[Scho 09] W. I. Schöllhorn, G. Mayer-Kress, K. M. Newell, and M. Michelbrink. "Time Scales of Adaptive Behavior and Motor Learning in the Presence of Stochastic Perturbations". *Human Movement Science*, Vol. 28, No. 3, pp. 319–333, 2009.

[Schu 95] E. G. Schukat-Talamazzini. *Automatische Spracherkennung*. Vieweg, 1995.

[Shan 49] C. E. Shannon. "Communication in the Presence of Noise". *Proceedings of the IRE*, Vol. 37, No. 1, pp. 10–21, 1949.

[Sigr 13] R. Sigrist, G. Rauter, R. Riener, and P. Wolf. "Augmented Visual, Auditory, Haptic, and Multimodal Feedback in Motor Learning: A Review". *Psychonomic Bulletin & Review*, Vol. 20, No. 1, pp. 21–53, 2013.

[Siir 11] P. Siirtola, P. Laurinen, J. Roning, and H. Kinnunen. "Efficient Accelerometer-based Swimming Exercise Tracking". In: *IEEE Symposium on Computational Intelligence and Data Mining (CIDM 2011)*, pp. 156–161, Paris, France, 2011.

[Sim 10] M. Sim and J.-U. Kim. "Differences Between Experts and Novices in Kinematics and Accuracy of Golf Putting". *Human Movement Science*, Vol. 29, No. 6, pp. 932–946, 2010.

[Slaw 08] S. E. Slawson, L. M. Justham, A. A. West, P. P. Conway, M. P. Caine, and R. Harrison. "Accelerometer Profile Recognition of Swimming Strokes". In: *The Engineering of Sport 7*, pp. 81–87, Springer, 2008.

[Slaw 12] S. E. Slawson, L. M. Justham, P. P. Conway, T. Le Sage, and A. A. West. "Characterizing the Swimming Tumble Turn Using Acceleration Data". *Journal of Sports Engineering and Technology*, Vol. 226, No. 1, pp. 3–15, 2012.

[Smit 11] S. W. Smith. *The Scientist and Engineer's Guide to Digital Signal Processing.* California Technical Publishing, 2011.

[Stam 13] A. Stamm, D. A. James, and D. V. Thiel. "Velocity Profiling using Inertial Sensors for Freestyle Swimming". *Sports Engineering*, Vol. 16, No. 1, pp. 1–11, 2013.

[Stit 10] J. T. Stites, G. Tavares, C. Madore, M. Kelly, R. J. Sander, J. A. Hadden, J. R. Held, G. L. Meanle, G. S. Kramer, and C. E. Bouton. "System for Determining Performance Characteristics of a Golf Swing". June 15 2010. US Patent 7,736,242.

[STMi 15a] STMicroelectronics N.V. "L3GD20H MEMS motion sensor: three-axis digital output gyroscope, Datasheet". `http://www.st.com/`, 2015. [Online; accessed 01-March-2015].

[STMi 15b] STMicroelectronics N.V. "LIS2DH MEMS digital output motion sensor: ultra low-power high performance 3-axis "femto" accelerometer, Datasheet". `http://www.st.com/`, 2015. [Online; accessed 01-March-2015].

[STMi 15c] STMicroelectronics N.V. "LSM6DS1 iNEMO inertial module: 3D accelerometer and 3D gyroscope, Datasheet". `http://www.st.com/`, 2015. [Online; accessed 01-March-2015].

[STMi 15d] STMicroelectronics N.V. "LSM9DS1 iNEMO inertial module: 3D accelerometer, 3D gyroscope, 3D magnetometer, Datasheet". `http://www.st.com/`, 2015. [Online; accessed 01-March-2015].

[Stor 04] D. Storek. "Sporting Equipment Provided with a Motion Detecting Arrangement". Dec. 23 2004. US Patent App. 10/605,414.

[Taba 97] S. Taba. "Electronic Timing Swimmer's Goggles". Nov. 11 1997. US Patent 5,685,722.

[Tan 93] M. Tan. "Cost-sensitive Learning of Classification Knowledge and its Applications in Robotics". *Machine Learning*, Vol. 13, No. 1, pp. 7–33, 1993.

[Teic 97] J. Teich. *Digitale Hardware/Software-Systeme.* Springer, 1997.

[Texa 15] Texas Instruments Inc. "MSP430F1611 data sheet". `http://www.ti.com/lit/ds/slas368g/slas368g.pdf`, 2015. [Online; accessed 02-January-2015].

[Theo 09] S. Theodoridis and K. Koutroumbas. *Pattern Recognition, Fourth Edition.* Elsevier, 2009.

[Tort 07] G. J. Tortora and B. H. Derrickson. *Principles of Anatomy and Physiology, 11th edition.* Wiley, 2007.

[Univ 14] University of Erlangen-Nuremberg, Speech Group. "JSTK - The Java Speech Toolkit". `https://code.google.com/p/jstk/`, 2014. [Online; accessed 25-May-2014].

[Vapn 95] V. Vapnik. *The Nature of Statistical Learning Theory.* Springer-Verlag, 1995.

[Vett 84] M. Vetterli and H. J. Nussbaumer. "Simple FFT and DCT Algorithms with Reduced Number of Operations". *Signal Processing*, Vol. 6, No. 4, pp. 267–278, 1984.

[Viit 88] J. T. Viitasalo. "Evaluation of Explosive Strength for Young and Adult Athletes". *Research Quarterly for Exercise and Sport*, Vol. 59, No. 1, pp. 9–13, 1988.

[Viol 01] P. Viola and M. Jones. "Rapid Object Detection Using a Boosted Cascade of Simple Features". In: *Proc. on the Conference on Computer Vision and Pattern Recognition (CVPR 2001), Vol. 1*, pp. 511–518, Kauai, USA, 2001.

[Wagn 06] K. Wagner. "Sport-Specific Measuring and Information Systems for Training Control". *International Journal of Computer Science in Sports*, Vol. 5, No. 2, pp. 90–93, 2006.

[Wals 04] M. Walsh, A. Arampatzis, F. Schade, and G.-P. Brüggemann. "The Effect of Drop Jump Starting Height and Contact Time on Power, Work Performed, and Moment of Force". *Journal of Strength and Conditioning Research*, Vol. 18, No. 3, pp. 561–566, 2004.

[Wein 10] J. Weineck. *Optimales Training.* Spitta, 2010.

[Weis 13] Y. Weiss, Y. Elovici, and L. L. Rokach. "The CASH Algorithm-cost-sensitive Attribute Selection Using Histograms". *Information Sciences*, Vol. 222, No. no number, pp. 247–268, 2013.

[Whea 10] A. L. Wheat and K. T. Larkin. "Biofeedback of Heart Rate Variability and Related Physiology: A Critical Review". *Applied Psychophysiology and Biofeedback*, Vol. 35, No. 3, pp. 229–242, 2010.

[Wilh 08] R. Wilhelm, J. Engblom, A. Ermedahl, N. Holsti, S. Thesing, D. Whalley, G. Bernat, C. Ferdinand, R. Heckmann, T. Mitra, F. Mueller, I. Puaut, P. Puschner, J. Staschulat, and P. Stenström. "The Worst-case Execution-time Problem - Overview of Methods and Survey of Tools". *ACM Transactions on Embedded Computing Systems*, Vol. 7, No. 3, Article 36, p. no pagination, 2008.

[Wils 05] J. S. Wilson. *Sensor Technology Handbook*. Elsevier, 2005.

[Witt 11] I. H. Witten, E. Frank, and M. A. Hall. *Data Mining, 3rd Edition*. Morgan Kaufman, 2011.

[Wolf 02] W. Wolf, B. Ozer, and T. Lv. "Smart Cameras as Embedded Systems". *Computer*, Vol. 35, No. 9, pp. 48–53, 2002.

[Wolp 97] D. H. Wolpert and W. G. Macready. "No Free Lunch Theorems for Optimization". *IEEE Transactions on Evolutionary Computation*, Vol. 1, No. 1, pp. 67–82, 1997.

[Worl 09] World Health Organization. *Global Health Risks: Mortality and Burden of Disease Attributable to Selected Major Risks*. World Health Organization, 2009.

[Wulf 07] G. Wulf and J. Su. "An External Focus of Attention Enhances Golf Shot Accuracy in Beginners and Experts". *Research Quarterly for Exercise and Sport*, Vol. 78, No. 4, pp. 384–389, 2007.

[Wulf 99] G. Wulf, B. Lauterbach, and T. Toole. "The Learning Advantages of an External Focus of Attention in Golf". *Research Quarterly for Exercise and Sport*, Vol. 70, No. 2, pp. 120–126, 1999.

[Yu 03] L. Yu and H. Liu. "Feature Selection for High-Dimensional Data: A Fast Correlation-Based Filter Solution". In: *Proc. of the International Conference International Conference on Machine Learning*, pp. 856–863, Washington, USA, 2003.

[Yuen 12] P. A. Yuen. "Swim Monitor". Jan. 26 2012. US Patent App. 13/262,073.

[Zhan 12] X. Zhang, Y. Liu, F. Zhang, J. Ren, Y. L. Sun, Q. Yang, and H. Huang. "On Design and Implementation of Neural-Machine Interface for Artificial Legs". *IEEE Transactions on Industrial Informatics*, Vol. 8, No. 2, pp. 418–429, 2012.

Glossary

accuracy-cost tradeoff describes the simultaneous consideration of classification accuracy and algorithm cost of embedded classification systems. In this thesis, algorithm cost is expressed in number and type of mathematical operations and memory demand.

augmented feedback is given by an external source like a trainer or a display. It provides information that the human is either not able to measure with his own senses or not able to process simultaneously [Sigr 13].

body sensor network (BSN) is a short-range wireless sensor network that consists of sensor nodes and processing nodes.

central processing unit (CPU) is the main processing unit of a computer or embedded system.

closed skill is a movement skill where environmental objects are stationary or at least predictable. Closed skills are also called self-paced and an example is golf.

code instrumentation is used to measure the runtime and memory consumption of computer programs. Therefore, the source code is enhanced with measurement logic.

concurrent feedback is feedback that is available during movement execution.

cyber-physical system (CPS) is a system integrating computational and physical processes. Physical processes are thereby controlled and monitored by embedded systems and networks. Computations and physical processes affect each other in feedback loops. Definition according to [Lee 08a].

data mining is the (semi-)automatic process of discovering meaningful patterns in data for the purpose of their interpretation.

embedded classification is the implementation of pattern recognition algorithms on embedded systems.

embedded system is an information processing system that is embedded in an enclosing product. Definition according to [Marw 03].

golgi tendon reflex is a muscle reflex where the agonist lengthens and the antagonist contracts on high muscle tension. The reflex is triggered by golgi receptors in the muscle.

graphical processing unit (GPU) is a dedicated graphic processor for computers or embedded system.

Hawthorne effect is a phenomenon that states that subjects of a data collection study are influenced by the fact that they observed [Roet 03]. The collected study data might be biased by this phenomenon.

hidden markov model is a technique to analyze sequential data for the purpose of pattern classification or pattern analysis.

inertial measurement unit (IMU) is a device to capture inertial (relative) movements. It consist of an accelerometer that senses linear acceleration and a gyroscope that senses angular velocity. Often, IMUs are implemented as microelectromechanical systems.

knowledge of performance (KP) is feedback that contains information about the quality and details of the movement that lead to the performance outcome.

knowledge of result (KR) is feedback that contains information about the outcome of movements or whether the goal was achieved or not.

Landau-notation is a measure that expresses the theoretical computational complexity of an algorithm in relation to the input data size.

microelectromechanical system is a electromechanical transducer that links electrical or magnetic forces and mechanical motion. Such a device can act as a kinematic sensor that transforms mechanical motion to electrical or magnetic force.

No Free Lunch theorem in the context of pattern recognition states that there is no single classifier that performs best in every pattern classification task. The original definition of this theorem can be found in [Wolp 97] and the definition in the context of pattern recognition in [Duda 01].

open skill is a movement skill where environmental objects are moving in an unpredictable fashion. Open skills are also called externally-paced and an example is basketball.

pattern is the instance of interest that a pattern recognition system processes. It e.g. can be a picture, a sound sample or a kinematic sensor recording.

pattern analysis is the in-depth analysis to derive the description of the pattern in an adequate level of abstraction.

pattern classification is the categorization of a pattern in an either predefined class (supervised classification) or undefined class (unsupervised classification).

pattern recognition is a methodology for data analysis which can be subdivided in the tasks of pattern analysis, pattern classification and data mining.

random access memory (RAM) is the main memory of a computer or embedded system.

stretch reflex is a muscle reflex where the muscle contracts when quickly stretched. The reflex is triggered by the spindle sensors in the muscle.

task-intrinsic feedback is created by the human sensory systems. The most important modalities are visual, auditory, tactile, kinesthetic and vastibular feedback.

terminal feedback is feedback that is available after task completion.

transducer is a device that transforms one type of energy into another [Wils 05].