

# Intelligent Sensing Techniques for Desk Workplace Ergonomics

Master's Thesis of

**Josh Romanowski**

At the Department of Informatics

Reviewer:	PD Dr. Victor Pankratius
Second reviewer:	Prof. Dr. Michael Beigl
Advisor:	PD Dr. Victor Pankratius

Duration: September 1, 2021 – February 28, 2022

Karlsruher Institut für Technologie  
Fakultät für Informatik  
Postfach 6980  
76128 Karlsruhe

---

I declare that I have developed and written the enclosed thesis completely by myself, and have not used sources or means without declaration in the text.

I followed the rules for securing a good scientific practice of the Karlsruhe Institute of Technology (Regeln zur Sicherung guter wissenschaftlicher Praxis im Karlsruher Institut für Technologie (KIT)).

**Karlsruhe, February 28, 2022**

Signed Romanowski

.....  
(Josh Romanowski)



# Abstract

Wearable devices with integrated inertial measurement units have enabled new sensing applications over extended periods. Work ergonomics is a particular domain where wearables and intelligent sensing can improve the health and well-being of many people. Persons in office environments are particularly prone to musculoskeletal disorders (MSDs) due to sustained bad posture. As a result, posture-related problems have become the third most frequently reported work health risk factor.

The goal of this work is to explore the application of Edge-AI technology for a continuous assessment of ergonomic risk factors in a desk workplace environment. The approach evaluates the ergonomic risk situation of a user based on their posture and the workplace constraints, and it infers suggestions for improvement. Relevant movements are captured continuously by a sensor network using accelerometers and gyroscopes mounted on a wearable glass frame as well as on an office chair armrest. This work proposes a novel domain-specific machine learning and state transition model for the long-term derivation of posture through the classification and aggregation of events that can occur in the sensor time series. Additionally, this work creates a manifold of movements that enables reasoning and recommendation capabilities for ergonomically relevant gestures and body movements. Finally, a software implementation successfully demonstrates the techniques in practice.



# Zusammenfassung

Wearables mit integrierten inertialen Messeinheiten ermöglichen neue Sensoranwendungen, die über längere Zeiträume hinweg messen. Die Arbeitsergonomie ist ein besonderer Bereich, in dem Wearables und intelligente Sensorik die Gesundheit und das Wohlbefinden vieler Menschen verbessern können. Personen in Büroumgebungen sind aufgrund anhaltender Fehlhaltungen besonders anfällig für Muskel-Skelett-Erkrankungen. Haltungsbedingte Probleme sind dadurch inzwischen der dritthäufigste gemeldete Risikofaktor für die Gesundheit am Arbeitsplatz.

Das Ziel dieser Arbeit ist es, die Anwendung von Edge-AI-Technologien für eine kontinuierliche Bewertung ergonomischer Risikofaktoren am Schreibtischarbeitsplatz zu untersuchen. Der erstellte Ansatz bewertet die ergonomische Risikosituation eines Nutzers auf Basis der Körperhaltung und der Arbeitsplatzbedingungen und leitet daraus Verbesserungsvorschläge ab. Relevante Bewegungen werden kontinuierlich von einem Sensornetzwerk mit Beschleunigungssensoren und Gyroskopen erfasst, die an einem tragbaren Brillenrahmen sowie an einer Armlehne eines Bürostuhls angebracht sind. Diese Arbeit schlägt ein neuartiges domänenspezifisches maschinelles Lernverfahren und Zustandsübergangsmodell für die langfristige Ableitung von Körperhaltungen durch die Klassifizierung und Aggregation von Ereignissen vor, die in den Sensorzeitreihen auftreten. Zusätzlich wird ein Manifold von Bewegungen erstellt, der Schlussfolgerungen und Empfehlungen für ergonomisch relevante Körperbewegungen ermöglicht. Schließlich demonstriert eine Software-Implementierung die Techniken erfolgreich in der Praxis.



# Contents

<b>Abstract</b>	<b>v</b>
<b>Zusammenfassung</b>	<b>vii</b>
<b>1 Introduction</b>	<b>1</b>
<b>2 Related Work</b>	<b>3</b>
2.1 Sensing for Desk Workplace Ergonomics . . . . .	3
2.1.1 Academic Context . . . . .	3
2.1.2 Patents . . . . .	5
2.2 Movement Manifolds for Ergonomics . . . . .	5
2.3 Summary . . . . .	6
<b>3 Desk Workplace Ergonomics</b>	<b>7</b>
3.1 Terminology and Scope . . . . .	7
3.2 Ergonomic Risk Factors of Desk Work . . . . .	8
3.3 Considered Postures . . . . .	9
3.4 Environmental Factors . . . . .	10
3.5 Summary . . . . .	10
<b>4 Sensing Posture with IMUs</b>	<b>11</b>
4.1 Considerations for the use of IMUs . . . . .	11
4.2 Sensor Placement . . . . .	12
4.3 Sensor Implementation . . . . .	13
4.4 Sensor Synchronization . . . . .	15
4.5 Feasibility . . . . .	16
4.6 Processing Pipeline Overview . . . . .	18
4.7 Summary . . . . .	19
<b>5 Data Preprocessing</b>	<b>21</b>
5.1 Denoising . . . . .	21
5.2 Normalization . . . . .	22
5.3 Data Augmentation . . . . .	22
5.4 Summary . . . . .	24
<b>6 State Transition Model</b>	<b>25</b>
6.1 Fundamental Idea . . . . .	25
6.2 States and Transitions . . . . .	26
6.3 Technical Background . . . . .	27
6.3.1 Convolutional Neural Networks . . . . .	27
6.3.2 Hidden Markov Models . . . . .	29
6.4 Transition Extraction . . . . .	29
6.4.1 Transition Detection . . . . .	30

6.4.2	Noise-Transition Classification . . . . .	31
6.4.3	Transition Separation . . . . .	32
6.5	Transition Classification . . . . .	34
6.6	Posture Derivation . . . . .	35
6.6.1	Posture Model . . . . .	35
6.6.2	Application Bias . . . . .	37
6.7	Summary . . . . .	37
<b>7</b>	<b>Ergonomic Application</b>	<b>39</b>
7.1	Posture Evaluation . . . . .	39
7.2	Environmental Evaluation . . . . .	42
7.3	User Application . . . . .	42
7.3.1	User Notification . . . . .	43
7.3.2	Data Presentation . . . . .	44
7.4	Summary . . . . .	47
<b>8</b>	<b>Ergonomic Movement Manifold</b>	<b>49</b>
8.1	Manifold Basics . . . . .	49
8.2	Learning the Movement Manifold . . . . .	50
8.3	Explainability of Movement Data using the Manifold . . . . .	50
8.4	Reasoning and Data Generation using the Manifold . . . . .	51
8.5	Summary . . . . .	52
<b>9</b>	<b>Evaluation</b>	<b>53</b>
9.1	Data Collection . . . . .	53
9.1.1	Desk Setup . . . . .	53
9.1.2	Data Collection Process . . . . .	54
9.1.3	Collected Data . . . . .	55
9.2	Transition Extraction . . . . .	56
9.2.1	False Extraction Rate . . . . .	56
9.2.2	Transition Extraction Performance . . . . .	56
9.3	Transition Classification . . . . .	60
9.3.1	Classification Metrics . . . . .	60
9.3.2	Feature Selection . . . . .	61
9.3.3	Data Augmentation & Manifold . . . . .	63
9.3.4	Transition Classifier Errors . . . . .	64
9.4	Posture Extraction . . . . .	68
9.4.1	BLEU-Score . . . . .	68
9.4.2	Posture Model Performance . . . . .	68
9.5	End-to-End Pipeline comparison . . . . .	70
9.5.1	Metric . . . . .	70
9.5.2	Comparison of Pipelines . . . . .	71
9.6	Ergonomic Movement Manifold . . . . .	73
9.7	Summary . . . . .	78
<b>10</b>	<b>Discussion</b>	<b>79</b>
10.1	Sensing Posture with IMUs . . . . .	79
10.2	State Transition Model . . . . .	80
10.3	Ergonomic Movement Manifold . . . . .	80
10.4	Ethical Considerations . . . . .	81
10.5	Personalization . . . . .	82
10.6	Summary . . . . .	82

---

<b>11 Conclusion and Outlook</b>	<b>83</b>
<b>Bibliography</b>	<b>85</b>
<b>Appendix</b>	<b>91</b>
A Expert Interview . . . . .	91
B Example Raw Data . . . . .	93
C Dataset Information . . . . .	94



# List of Figures

3.1	Areas of ergonomics . . . . .	8
3.2	Four degrees of freedom for posture . . . . .	9
4.1	Sketch of the sensor placement . . . . .	12
4.2	Image of an Arduino Nicla Sense ME board . . . . .	13
4.3	Images of the used sensor boards . . . . .	14
4.4	Example for raw motion data during unsupervised desk work . . . . .	16
4.5	Example for raw motion data of the glasses sensor during posture changes .	17
4.6	Example for raw motion data of the glasses sensor between posture changes	17
4.7	Example for sensor drift in the raw orientation data of the glasses sensor . .	18
4.8	Overview of the processing pipeline . . . . .	18
5.1	Plots of different time series data augmentation techniques . . . . .	23
5.2	Example time warping time distortion curve . . . . .	24
5.3	Example magnitude warping scaling curve . . . . .	24
6.1	Fundamental idea of the state transition model . . . . .	26
6.2	Visualization of the state transition understanding of posture . . . . .	27
6.3	Visualization of 2D-convolution . . . . .	28
6.4	Example for transition extraction . . . . .	30
6.5	Visualization of the transition extraction using gyroscope extreme values . .	30
6.6	Visualization of topological prominence . . . . .	31
6.7	Visualization of the filtering of noise-transitions . . . . .	32
6.8	Architecture of the noise-transition classifier . . . . .	32
6.9	Example for transitions where the transition windows overlap . . . . .	33
6.10	Example for the masking of close transitions . . . . .	33
6.11	Example transition classification for a sequence of posture changes . . . . .	34
6.12	Architecture of the transition classifier . . . . .	35
6.13	Example posture derivation for a sequence of posture changes . . . . .	36
7.1	Plots of the scaling curves for the posture and activity scores . . . . .	41
7.2	Example ergonomic scoring . . . . .	41
7.3	Examples for desktop notifications. . . . .	43
7.4	Examples for smartphone notifications. . . . .	44
7.5	Mockups of the data presentation application . . . . .	46
8.1	Illustration of the principal component analysis . . . . .	50
8.2	Visualization of vector combination on the manifold . . . . .	51
9.1	Sketch of the desk setup for data collection . . . . .	54
9.2	Overview of the data collection process . . . . .	55
9.3	Noise-transition removal for example unsupervised desk work . . . . .	57
9.4	Transition extraction examples . . . . .	59

9.5	Comparison of true and derived transitions . . . . .	64
9.6	Confusion matrices of the transition classifier . . . . .	66
9.7	Confusion matrices of the transition classifier after data augmentation . . . . .	67
9.8	Comparison of true and derived postures . . . . .	70
9.9	Cumulative explained variance of the first principal components . . . . .	73
9.10	Manifold structure for inverse transitions and transitions with the same direction . . . . .	75
9.11	Derivation of two example transitions on the manifold . . . . .	76
9.12	Comparison of derived transition windows using the manifold method and actual recordings . . . . .	77
B.1	Example raw sensor data . . . . .	93

# List of Tables

4.1	Specifications of the used sensors on the Arduino Nicla Sense ME board . . .	14
7.1	Guide values for desk environmental and desk setup ergonomic factors . . .	42
9.1	Transition extraction performance . . . . .	58
9.2	Confusion matrix for a binary classification task . . . . .	60
9.3	Transition classifier performance per selected features . . . . .	62
9.4	Transition classifier performance for different data . . . . .	63
9.5	Posture model performance for different datasets . . . . .	69
9.6	End to end model performance comparison . . . . .	73
C.1	Total recorded transitions per transition type and data set . . . . .	94



# Chapter 1

## Introduction

Inertial Measurement Units (IMUs) have had significant improvements in size, cost, and energy efficiency that facilitate their use in everyday wearable devices [1]. In combination with Edge-AI and on-device sensor data processing, new technical opportunities are now emerging in a wide spectrum of applications [2]. Especially for the application area of fitness but also increasingly in the health domain many intelligent sensing solutions emerged in the last years [3].

Work ergonomics is a domain where wearable devices and intelligent sensing can improve the health of a large number of people. One particularly relevant area of application is the desk workplace. For example, according to the Sixth European Working Conditions Survey by Eurofound in 2015, 57% of workers in the European Union including the United Kingdom (EU28), work with computers, smartphones, and laptops at least a quarter of the time [4]. Similarly, a Eurostat study has shown that the use of computers and the internet in enterprises has steadily increased from 44% in 2012 up to 54% in 2019 [5].

Together with a steady increase in desk work, the amount of time spent sitting is also increasing. 58% of people in the EU28 spend at least a quarter of their work time sitting, with 28% spending even all or most of their time sitting [4]. This makes workers at a desk workplace particularly prone to musculoskeletal disorders (MSDs) if they maintain a poor sitting posture. Poor posture and low sitting activity have been confirmed by many studies to correlate with MSDs and other health risks [6, 7, 8]. Despite the recognition of prolonged sitting as the third most frequently reported work risk factor in the EU28 [9], there is still no established solution for desk work ergonomics.

The goal of this work is to explore intelligent sensing techniques for the automatic assessment of desk workplace ergonomic risk factors under considerations of low-power devices and long-term sensing. This work presents a sensor setup using accelerometers and gyroscopes on the frame of wearable glasses and on the desk chair to measure relevant movements. A novel domain-specific AI approach, the state transition model, extracts the user's posture, leveraging the long-term static nature of desk work. The approach derives the posture by extracting and accumulating interesting events from the data which optimizes both power usage and model accuracy. The extracted posture and workplace

conditions are used to determine workplace ergonomic risk factors such as sustained bad posture. Based on the assessment, the system gives immediate suggestions for posture improvement. The application of manifolds to the ergonomic movement data provides further understanding of the data. Overall, this work provides valuable insights into typical ergonomic risk factors of desk work. It also provides non-invasive supporting technology with local privacy so that the approaches can be applied to real desk workplaces.

This work is structured as follows: Chapter 2 gives an overview of related work in the area of intelligent sensings for desk work ergonomics. Chapter 3 introduces domain knowledge about desk work ergonomics relevant for this work. Chapter 4 presents the wearable IMU-based sensor setup for sensing the user's posture during desk work. Chapter 5 explains data preprocessing techniques that prepare the collected sensor data for further processing. Chapter 6 presents the state transition model that derives the posture from the preprocessed sensor data. Chapter 7 describes the ergonomic assessment of the derived posture and applications for presenting the assessment to the user. Chapter 8 presents explaining and reasoning about ergonomic movement data through the application of manifolds to the data. Chapter 9 evaluates the techniques using a practical implementation with recorded data. Chapter 10 discusses the findings of this work before Chapter 11 concludes by summarizing the results and providing an outlook on future work.

# Chapter 2

## Related Work

This work investigates different intelligent sensing techniques for desk workplace ergonomics. In particular, it looks at the sensor setup, the processing of the sensor data to extract the user's posture from the sensor data, the ergonomic assessment of posture, and the application of manifolds to the ergonomic movement for explainability and reasoning about the data. There is already work in research and patents on extracting and ergonomically evaluating posture. However, previous approaches come with various shortcomings and limitations. The first section of this chapter (Section 2.1) discusses related research on sensing and data processing techniques. Concerning the application of manifolds to ergonomic movements, there is limited previous research. Section 2.2 describes how related work applies manifolds to ergonomic movements.

### 2.1 Sensing for Desk Workplace Ergonomics

Section 2.1.1 looks specifically at academic works on sensing for desk workplace ergonomics and Section 2.1.2 takes a closer look at patents in this area.

#### 2.1.1 Academic Context

Paliyawan et al. [10] use a Kinect camera to capture video footage of a desk worker. Their work uses the Kinect software to extract the orientation of 32 body joints which they use to describe the body posture. Paliyawan et al. apply different machine learning methods to classify whether the user is sitting still or moving. They also use a threshold-based approach on the Kinect body joint data to determine when the user stands up. Paliyawan et al. assign a health risk level, depending on how long the user sits still and doesn't stand up. If the user's risk level is high, the user is then alarmed and given a report. According to Paliyawan et al.'s work, the motivation for this approach is to encourage the user to take small breaks, which prevents ergonomic injuries. Their work concludes that the system is useful to users without interrupting the user's work. It was also found that the visual variety of the tracked objects negatively affects system performance.

Wu et al. [11] also use the Kinect camera and body joint extraction to determine the user's posture. Their work however explores how the extracted posture can be used to actively

actuate furniture to match ergonomic guidelines. In particular, Wu et al. ask the user to perform two predefined postures to extract postural features of the user. Based on these postural features they calculate an ergonomically correct configuration of the desk chair and table. The calculated configuration is either automatically adjusted or feedback is given to the user. After an initial setup, the desk setup is not corrected anymore. Wu et al. conclude that their sensing and actuation approach significantly the user's postures and helped users conform to ergonomic guidelines. Similar to Paliyawan, Wu et al. noted that the accuracy of the model is impacted by the variety of clothes worn by the users.

Both Paliyawan et al. and Wu et al. use a video-based approach for posture extraction to perform an ergonomic assessment of the posture. Collecting and processing video data requires significantly more processing capabilities than inertial measurement-based approaches. Video data is also considerably more impeding on the user's privacy. Paliyawan et al. mention the user's privacy in their work, but state that the original video data is deleted after processing, which in their opinion avoids the problem. Wu et al. only initially configure the desk setup and after that doesn't give feedback anymore.

Prueksanusak et al. [12] use an array of pressure sensors embedded in the seating platform and the backrest of a user's office chair to measure posture. Using different machine learning algorithms, they classify the collected time series of the sensors into five sitting postures. For real-time processing, Prueksanusak et al. send the user's data to a cloud server. The distribution of postures over time is presented to the user using graph visualizations. With this, Prueksanusak et al. aim to promote the user's understanding of their sitting posture. Prueksanusak et al. conclude that the approach is capable to differentiate the five sitting postures.

Jeong et al. [13] use a combination of pressure sensors embedded in the seating platform and distance sensors in the backrest of a user's office chair to measure posture. With this data, they aim to detect different sitting postures. Jeong et al. use a k-Nearest-Neighbor classifier to distinguish between eleven postures in the sensor time series. They conclude that the mixture of pressure and distance sensors can accurately measure posture. Although Jeong et al. do not propose any specific application of the proposed posture extraction they mention the possibility to integrate the data with a real-time feedback system.

The use of pressure and distance sensors in the user's chair requires fewer processing capabilities and power compared to a video-based system. The mentioned works on chair-embedded sensing approaches only use the chair to sense posture. Therefore, postures of the head can't be detected. Similar to the works with cameras, the mentioned works on pressure and distance-based sensing do not look at the mostly static nature of desk work to design their processing.

Jun et al. [14] use three body-worn inertial measurement units (IMUs) to measure posture. The sensors are placed on the user's forehead using a headband, the arm using an armband, and taped on the thorax. Jun et al. use the IMUs as inclinometers determining the thorax angle, the head angle the arm angle, and the neck angle relative to a base posture. Comparing the determined angles to the optimal ergonomic posture, Jun et al. determine different measures to assess posture. One example of such a measure is the percentage of time spent in predefined non-neutral angular ranges. They conclude that their IMU-based approach can capture the user's posture despite noting that there is inter-person variability in the results.

Severin et al. [15] also use an IMU-based approach to measure the head posture. In contrast to Jun et al., they create a neck wearable sensor array consisting of three IMUs. Severin et al. propose a threshold-based system that discriminates between three posture risk

levels depending on the angles of the sensors. The work also proposes real-time feedback using text and audio. They conclude that the wearable can detect forward head posture and that their system is portable and easy to use.

Sinha et al. [16] attach a smartphone to the rear upper trunk of the user. They collect data from the inbuilt accelerometer, magnetometer, and gyroscope of the smartphone. Moving windows of the sensor time series are then used to calculate different morphological features such as mean absolute value, standard deviation, etc. Sinha et al. then use different machine learning algorithms to classify between five sitting postures based on these features. They conclude that it is possible to distinguish between the considered five sitting postures using only the smartphone IMU data.

Jun et al., Severin et al., and Sinha et al. use a placement of IMUs that is not natural to a user working at a desk. Jun et al. require the user to tape sensors to their body. Severin et al. use a neck wearable which is not commonly used. And Sinha et al. attach a smartphone to the user's upper trunk. Therefore, these approaches offer significant problems for practical usability due to the impediment of the user.

Jun et al. and Severin et al. directly use the orientations as determined by IMU sensor fusion to infer information of the posture. This approach is prone to the accumulation of sensor drift over time in the sensor orientation fusion due to the errors of the sensor signals. Both works also don't specifically regard the effect of sensor drift in their analysis. This is particularly problematic when implementing an ergonomic assessment solution that captures posture over an extended period of time, i.e., over several hours.

### 2.1.2 Patents

The Inwerk GmbH [17] patented a system for interactive office ergonomics that uses an IMU which is placed on the office chair of a user to detect their movement. The patent uses the IMU data to infer "moves" as a measure of the user's activity similar to a pedometer. The patent does mention the intent to collect a movement profile but does not explain an algorithm to extract the features from the IMU data. The patent includes user notification with either vibration, sound, or by actuating LEDs. These notifications are also released when environmental factors cross guide values.

In their patent, the Wilkhahn Wilkening + Hahne GmbH & Co. KG [18] present a system that combines a seat and a portable device including at least one movement sensor for the characterization of a person's movement. The patent uses the measurements to create a movement profile. The movement profile contains movements along the body axis and scattering of movements around the neutral positions of the seat. The patent presents the idea of assessment of the movement profile within a time period. This assessment is then envisioned to notify the user about the characteristics of their movement.

Both mentioned patents use IMUs to measure the user's movement during office work. They use the collected movement data to assess a movement profile with the goal to release notifications when guide values are exceeded. However, similar to research on sensing for desk work ergonomics the patents directly estimate movement characteristics from the IMU data without addressing the long-term static nature of desk work.

## 2.2 Movement Manifolds for Ergonomics

As there is very limited work on the specific application of manifolds to desk work ergonomic movements, this section considers related work that uses manifolds for ergonomics in general.

Lin et al. [19] use movement manifolds to recognize patient handling activities for the ergonomics of nursing workers. They use a wearable insole to sense and characterize information about patient handling activities. The work uses manifold learning to project the high-dimensional raw pressure data onto a low-dimensional manifold space. Similarity comparison of the trajectories of the movements on the manifold with learned manifolds is then used to determine the sensed activity. The work concludes that the manifold can detect the intrinsic structure of the pressure data and helps achieve good activity recognition accuracy.

Chen et al. [20] created a system for posture estimation of construction workers using a single IMU. Their goal is to enable easy measurement of the construction worker's posture for safety monitoring. They use a neural network-based approach to infer the surface level, the walking activity, and the gait phase from IMU data. Chen et al. then use manifold learning with a Gaussian process dynamic model to translate from the extracted low-dimensional movement characteristics back to a high-dimensional joint angle model of the full human posture. They conclude that this approach can derive posture within small deviations.

As the works of Lin et al. and Chen et al. demonstrate, there is already a body of work on the use of manifold learning with the application to ergonomics. However, the mentioned works do not specifically consider movements at the desk workplace but for other applications. Also, the mentioned works use manifolds for activity recognition and posture reconstruction. There is a particularly large gap in the use of manifolds for explainability and reasoning about ergonomically relevant movements.

## 2.3 Summary

Previous work on sensing for desk ergonomics has focused primarily on different sensing approaches for the user's movements and approaches to feature extraction and ergonomic evaluation directly from sensor data. None of the mentioned works utilizes the long-term mostly static nature of desk work to develop specialized data processing techniques. The high computing power and battery requirements of camera-based approaches make processing hard to implement directly on-device. Pressure-based sensing approaches focus on the desk chair and disregard the head posture. Previous work on IMU-based sensing directly uses the orientation or position information of the sensor fusion to extract posture which makes the approaches susceptible to sensor drift. Also, previous sensor placements in IMU-based approaches impede the user which hinders practical usability. The use of movement manifolds for explaining and reasoning about ergonomically relevant movements are barely explored. The work explores intelligent sensing techniques to address the major shortcomings of previous work on desk work ergonomics. The following chapter presents the domain knowledge on desk workplace ergonomics that is relevant for this work.

# Chapter 3

## Desk Workplace Ergonomics

Domain knowledge about desk workplace ergonomics plays an important role throughout this work. The following chapter introduces the desk work ergonomics to provide an overview of the relevant domain knowledge for this work. Section 3.1 introduces the terminology of desk work ergonomics and the reason why this particular application scope is relevant. Section 3.2 presents relevant desk workplace ergonomic risk factors overall and the specific importance of sitting posture. Section 3.3 introduces the four postures regarded in this work. Section 3.4 discusses relevant environmental factors.

### 3.1 Terminology and Scope

The International Ergonomics Association defines "ergonomics" (from Greek *ergon* (work) and *nomos* (laws)) as follows [21]:

"Ergonomics [...] is the scientific discipline concerned with the understanding of interactions among humans and other elements of a system, and the profession that applies theory, principles, data, and methods to design in order to optimize human well-being and overall system performance."

Ergonomics is an interdisciplinary science comprising subjects such as anatomy, physiology, psychology, but also engineering, human-computer interaction, and many more. Figure 3.1 gives an overview of the different areas of ergonomics. The goal of ergonomics is to apply all these disciplines to design products, workplaces, or systems that suit the people who use them.

In general, the application of ergonomics to workplace design is of interest because people spend a significant part of their lives at work. Therefore, ergonomics has the potential to severely impact the lives of many people. In recognition of the importance of ergonomics, the EU has issued a series of directives requiring its member states to take measures to encourage the ergonomic design of workplaces [22].

In the context of digitalization, a significant and constantly growing proportion of working time takes place at a desk. For example, according to the Sixth European Working Conditions Survey by Eurofound in 2015, 57% of workers in the EU28 work with computers,

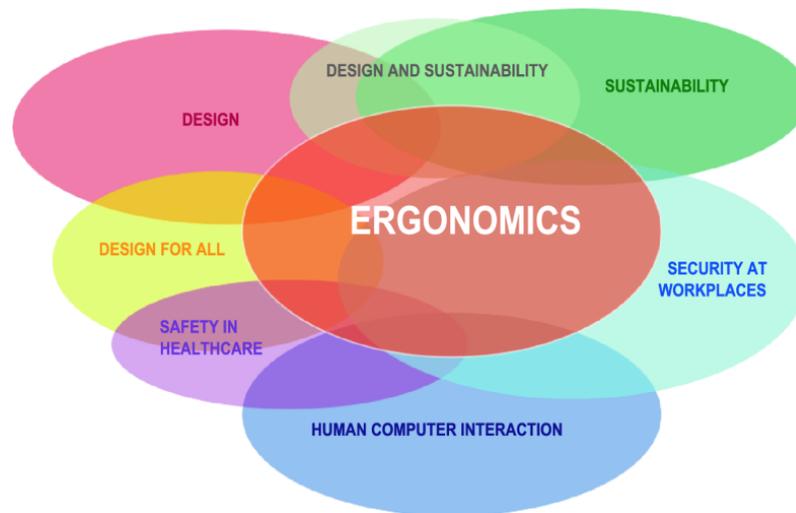


Figure 3.1: Areas of ergonomics (image taken from [23]). The bubbles represent different disciplinary areas that are incorporated in ergonomics. Ergonomics is a multi-disciplinary science aiming to optimize human well-being.

smartphones, and laptops, etc. at least a quarter of their working hours [4]. Similarly, an Eurostat study has shown that the use of computers and the internet in enterprises has steadily increased since 2012 up to 54% in 2019 [5]. A continuation of the trend towards more computer work is expected. Therefore, the desk workplace is a particularly important application area of ergonomics. Desk workplace ergonomics is simply the application of ergonomics to the desk workplace.

### 3.2 Ergonomic Risk Factors of Desk Work

People at the desk workplace are particularly exposed to ergonomic risk factors which can be defined as "actions or conditions that increase the likelihood of injury to the musculoskeletal system" [24]. The scope of risk factors in the desk workplace is multifaceted and includes individual physiology, work environment, technology, work organization, and psychosocial factors, among others [25].

Ergonomic risk factors related to sitting posture at the desk are especially relevant. In particular, it has been shown that prolonged periods of sitting and poor sitting posture are associated with a wide variety of physical conditions such as back pain, an increased risk for musculoskeletal disorders (MSDs), health risks such as coronary heart disease, diabetes, overweight, certain types of cancer, and even psychological disorders such as depression and anxiety [7, 26, 27]. Since it is predicted that the proportion of desk work and the associated duration of sitting will increase, the health risks of incorrect sitting posture are attracting more and more attention. The risk of poor posture and low activity at work is also recognized by government bodies [8] and in public media [28]. A physiotherapist interviewed as part of this work indicated that about 20-40% of the work-related issues they treated were related to sitting.

The assessment of ergonomic risk factors is complicated by the fact that there is a multitude of individual factors that make each person's ergonomic risk situation different. Such factors include gender, physical exercise, mental stress, and more [29]. For example, a person that does sports outside of the work at the desk can compensate for lower sitting activity while a person who does not exercise suffers from a greater risk of postural problems. Also, different people have different musculoskeletal dispositions which is why the same posture may have different implications for them.

Despite the awareness of the severity of the related issues for desk workers and the extensive research on the ergonomics of desk work, there is a lack of practical smart sensor solutions for the automatic assessment of desk workplace ergonomic risk factors as shown in Chapter 2. Such sensing solutions could enable a personalized objective assessment of the user’s desk workplace ergonomic risk situation. This is particularly striking as intelligent sensor solutions especially in the application area of fitness [30, 31, 32] but also increasingly for health are already widely used [33].

Another relevant factor is that the long-term nature of desk ergonomics also makes comprehensive medical studies impractical. A sufficiently long-term controlled study requires a considerable amount of effort and is difficult to implement in the normal working environment of individuals.

This work conducted an interview with a physiotherapist who has over 20 years of experience in treating conditions associated with desk work. The interview consisted of a number of open questions to get an impression of relevant factors of desk ergonomics from the medical point of view. A summary of the interview can be found in the appendix (Section A). In the interview, the expert said that the relationship between sitting posture and MSDs is practically strongly established. However, they added that due to the long-term nature of ergonomics, the actual study situation is difficult. In the interview, it was emphasized that each patient’s risk situation was highly variable. People between 40 and 50 years of age and people that don’t do sports are most affected. However, there are also younger patients.

### 3.3 Considered Postures

Sustained poor posture is a particularly important desk workplace ergonomic risk factor. Research showed that office workers with chronic lower back pain sit in a more asymmetrical way than healthy workers [34]. This work focuses on four degrees of freedom (DoFs) of the head and trunk because research found these factors to be particularly relevant to the physical risk factors [35, 36]. For each of the four degrees of freedom, this work regards whether the posture is neutral (center) or biased towards either side. It thereby discretizes the postures. Figure 3.2 illustrates the four degrees of freedom.:

- **Head angle** denotes the vertical tilt of the head. The head angle may be either *center*, *up*, or *down*. A non-neutral head angle may for example occur when the workstation is not correctly set up so the user has to look down to see their monitor.

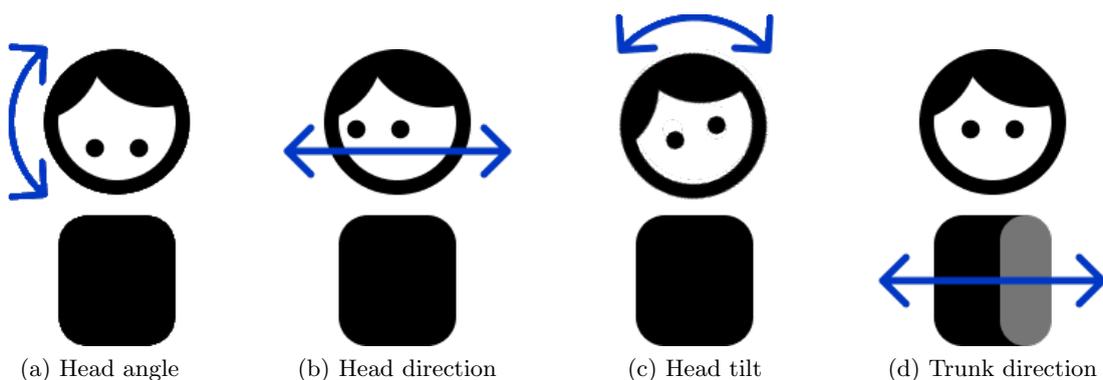


Figure 3.2: Four degrees of freedom (DoFs) for posture. For each of the DoFs, this work regards whether the posture is neutral (center) or biased to either side.

- **Head direction** identifies the torsion of the head around the body axis. The direction may be *center*, *left*, or *right*. A bias in the head direction is very prominent for dual-monitor setups where one screen is used more likely than the other. This is often the case when the monitors are used for different purposes.
- **Head tilt** is the horizontal tilt of the head. The head may be either *center*, *lefttilt* or *righttilt*. Head tilt is often a postural habit of users or may occur due to fatigue.
- **Trunk direction** denotes the torsion of the trunk around the body axis. The three possible configurations are *center*, *chairleft*, or *chairright*. Like head direction, a bias in trunk direction may be caused by predominantly using one screen a dual-monitor setup. This particular posture is also often a result of postural habit.

This work defines the overall posture of a person as the combination of the postures of the four degrees of freedom. For example, a person can look up and to the right at the same time, while their head tilt and trunk direction are neutral. Thus, the overall posture is described by specifying the posture of each of the four degrees of freedom.

### 3.4 Environmental Factors

Apart from the user's posture, the office environment and the desk setup of the user can impact their wellbeing at the desk workplace as well. Environmental factors that are relevant for desk workplace ergonomics include temperature, relative humidity, lighting, and indoor air quality (IAQ). In addition to these environmental factors, the workplace setup of the user has an impact on the ergonomic risk. The user's workplace setup includes how they set up their screen(s), desk, and chair. Relevant factors include but are not limited to the head-to-display angle, head-to-screen distance, chair seat height, chair seat depth, and workstation height [37]. There exist checklists and guidelines as well as specialized ergonomic furniture that aim to encourage good workplace setup and environmental conditions [38].

Although the user's sitting posture is the focus of this work, it also touches on how the assessment of the work environment can be included. Through this, this work aims to provide a perspective on a joint sensor-based assessment of postural and environmental ergonomic risk factors.

### 3.5 Summary

This chapter introduced desk workplace ergonomics as the application domain for this work. It showed that sustained poor posture and lack of postural activity are two particularly important desk workplace ergonomic risk factors. This chapter introduced four particular postures for which poor posture and low postural activity are investigated in this work. The following chapter introduces the sensing solution that measures the user's posture.

# Chapter 4

## Sensing Posture with IMUs

The automatic evaluation of postures requires a sensing setup that captures the user's body posture. The following chapter describes this work's sensing approach that uses inertial measurement units (IMUs) for measuring the user's posture. Section 4.1 discusses the considerations involved in using IMUs to measure posture. Section 4.2 presents concrete sensor placement using a sensor on a wearable glasses frame as well as on the armrest of the desk chair. Section 4.3 shows the practical sensor implementation. Section 4.4 addresses sensor synchronization between multiple sensors. Section 4.5 demonstrates typical characteristics of the recorded data in a feasibility study. Finally, Section 4.6 gives an overview of the processing pipeline to derive posture from the sensor data.

### 4.1 Considerations for the use of IMUs

This work uses inertial measurement units (IMUs) as the primary means to measure movement data for posture extraction. An IMU is a measurement device that combines an accelerometer, a gyroscope, and sometimes a magnetometer to measure movement [39]. Apart from using the raw sensor data of accelerometers and gyroscopes, IMUs combine sensor information of the sensors to derive device orientation or position. This is called sensor fusion [40].

This work uses IMUs instead of other sensing approaches, in particular cameras. One reason for this decision is that human motion capturing using IMUs creates less data compared to video-based approaches. The lower amount of data requires much lower computing capabilities and also less processing power. These advantages are particularly relevant to be able to implement the processing directly on mobile devices where compute power limitations and battery runtime play an important role.

A second important factor is that the collection of IMU data is much less invasive to the user's privacy than permanently recording video of the user [41]. Cameras not only capture the user's posture but everything in the frame. This includes possibly identifying or sensitive information about the user or bystanders. Therefore, the use of IMUs presents a purpose-bound and privacy-friendly sensor solution.

One significant disadvantage of using IMUs for position or orientation information using sensor fusion is the accumulation of sensor drift. As the measurements of gyroscopes and accelerometers are inherently noisy and biased, the integration of acceleration and angular velocity accumulates errors over time. Therefore, sensing with IMUs over longer periods of time requires a way of handling the accumulation of sensor drift. Different approaches combat this issue in practice, using more complex models like Kalman filters or by fusing the IMU data with other sensors such as for example GPS [40]. Another possible approach is to limit the time spans considered so that sensor drift is limited in the time frame under consideration.

In addition to the IMU data for extraction of a user’s posture this work utilizes environmental sensors. These sensors measure for example light, noise, humidity, and air pollution that are relevant factors for desk workplace ergonomics.

## 4.2 Sensor Placement

This work utilizes two sensor boards for sensing the user’s posture. One board is mounted on the side of the temple of a wearable glasses frame. The other is on the armrest of the office chair. See Figure 4.1 for a sketch of the sensor placement. This joint approach to measuring motion allows deriving more complex postures than known approaches that measure at only one location, i.e., only at the chair, or with a head wearable.



Figure 4.1: Sketch of the sensor placement (image taken from [42], modified). The red squares denote the position of the two sensors on the frame of wearable glasses and the armrest.

Previous work established the use of head-placed IMUs for detecting postures that are relevant for ergonomics [15]. The sensor placement on the glasses allows to detect even small movements and changes in orientation of the user’s head as the movement of the head translates into a movement of the glasses. Therefore, this sensor is particularly important for the detection of head posture. The placement on the side of the wearable glasses frame’s temple ensures that the sensor does not interfere with the user’s view or hinder head movement.

The second sensor on the desk chair’s armrest provides additional information on the movement and the orientation of the user’s spine. In particular, the combination of the sensor data of the head and the chair provides estimating information about the torsion of the spine. Additionally, the sensor on the armrest provides information about arm support on the chair and the inclination of the chair. An important consideration with the placement is that the board may not interfere with the use of the armrest.

This sensor placement provides a simple and unintrusive approach to measuring head and trunk postures. In particular, this setup is able to measure the posture for all of the

considered degrees of freedom (see Section 3.3). Additional sensors require more power and additional synchronization and computational effort. Therefore, this work uses only the two necessary sensors. Nevertheless, additional sensors on the backrest may provide more information when considering more and other postures. In particular, sensors on the backrest or under the seating platform can provide more data on the sitting posture. Note that the methods proposed in this work can generally be expanded to work with only one or more than two sensors.

### 4.3 Sensor Implementation

This work uses the Arduino Nicla Sense ME board as a sensor platform. See Figure 4.2 for an image of the board.

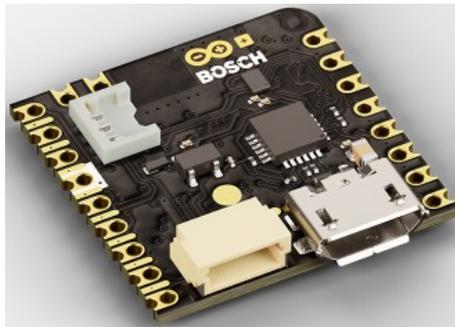


Figure 4.2: Image of an Arduino Nicla Sense ME board (image taken from [43], cropped).

The Arduino Nicla Sense ME is an Arduino-based development board. The board has a microelectromechanical systems (MEMS)-based IMU with an accelerometer and a gyroscope. Additionally, the board contains a magnetometer, a pressure sensor, and an environmental sensor that measures humidity, temperature, and volatile organic compounds (VOC). This work also utilizes the virtual orientation sensor provided by the BSX Lite Sensor Fusion library that is integrated into the IMU. This virtual sensor fuses the accelerometer, gyroscope, and magnetometer information to provide information about the orientation of the board. Additionally, the BSX Library contains calibration algorithms for the sensors [44]. The board also contains other physical and virtual sensors that this work doesn't use [43].

Table 4.1 contains a list of the used sensors, their axes, sensor ranges, resolutions, and the applied sampling rates. As the sensor specifications illustrate this particular setup can sense the user's movements very precisely. Also, the environmental sensors on the board enable additional evaluation of environmental ergonomic features.

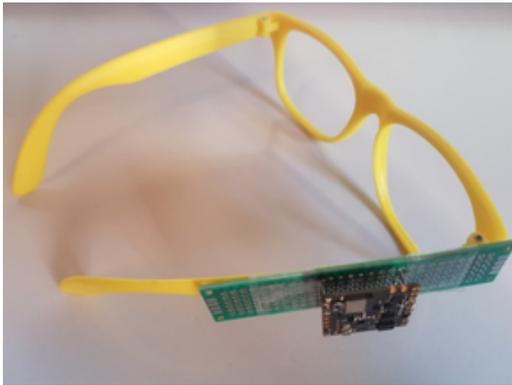
In the prototype setup, the sensor on the glasses frame is soldered to a breadboard as the form factor of the board prohibits the flat placement of the board on a surface. The breadboard is then taped to the side of the glasses (see Figure 4.3a). Weighing about 4 g, the weight of the sensor itself is not noticeable, barring the added weight of a battery. For a market-ready implementation of the approach, this work envisions integrating the sensor directly into smart glasses.

In this work's implementation, the board on the armrest is embedded into a 3D-printed frame that is taped to the armrest of the chair (See Figure 4.3b). This allows for placing the sensor such that it does not prevent the user from using their armrest. It also ensures the safety of the user and sensor in case of possible contact.

The Nicla board also contains a Bluetooth Low Energy (BLE) module that enables wireless communication between the sensors and the processing device.

Table 4.1: Specifications of the used sensors on the Arduino Nicla Sense ME board. If a field does not apply or no data was found, the field is crossed out.

Sensor	Sensor Type	Axes	Sensor Range	Sensor Resolution	Sampling Rate
BHI260AP	Accelerometer	x, y, z	$\pm 16 \text{ g}$ $\approx \pm 157.91 \text{ m/s}^2$	$\frac{1}{209} \text{ m/s}^2$	50 Hz
	Gyroscope	x, y, z	$\pm 2000^\circ/\text{s}$ $\approx \pm 34.91 \text{ rad/s}$	$\frac{1}{16}^\circ/\text{s}$	50 Hz
	Virtual Orientation Sensor	x, y, z, w	quaternion	-	50 Hz
BMM150	Magnetometer	x, y, z	$\pm 1300 \mu\text{T}$ (x,y) $\pm 2500 \mu\text{T}$ (z)	$0.3 \mu\text{T}$	50 Hz
BMP390	Pressure	-	300 hPa–1250 hPa	0.016 hPa	1 Hz
BME688	Temperature	-	$-40^\circ\text{C}$ – $85^\circ\text{C}$	$0.5^\circ\text{C}$	1 Hz
	Humidity	-	0%–100%	-	1 Hz
	VOC	b-VOC [45]	various scalings	-	1 Hz



(a) Image of the wearable glasses prototype with the attached sensor board.



(b) Image of the encased sensor board attached to the armrest of the chair

Figure 4.3: Images of the used sensor boards. The left image shows the wearable glasses prototype. The Nicla board is attached indirectly via a soldered breadboard to place the sensor firmly despite the uneven form factor of the Nicla Board. The right image shows the second Nicla board that is encased in a 3D-printed protection casing attached to the armrest of a desk chair.

## 4.4 Sensor Synchronization

To jointly use the information of the different sensors, the measurements have to be synchronized. However, sensor synchronization, especially with wireless transmission of the sensor data is non-trivial and subject to multiple synchronization issues.

1. The sensors have only local times and no understanding of a global time.
2. Each sensor's local clock may be subject to clock drift and offset.
3. The wireless transmission of the sensor data introduces different non-deterministic delays such as the time from accessing the network interface to sending, access time to the channel, transmission time at the physical layer, and receiving time spent in the arrival network interface.

In summary, a solution is necessary to tag the sensor data with an estimate of a timestamp of the time of measurement with respect to a global time. Note that "global" time here only means global for the system and not global with respect to e.g. Coordinated Universal Time. Time synchronization is a well-studied problem in the literature [46].

Since the methods described later in the processing pipeline of this work are capable of learning to adapt to small timing distortions, this work uses the following approach for sensor synchronization. The basic idea is to periodically send a timestamp every second from a central device to the sensors from which the sensors estimate the difference of their local time  $t_{local}$  to the time of the central device  $t_{central}$ .

$$\Delta t = t_{local} - t_{central} \quad (4.1)$$

To account for non-deterministic fluctuations in the time lag between the central device and sensor, the difference is estimated using the mean difference between received time and local time over  $k$  values. In practical experiments smoothing over  $k = 10$  timestamps results in a sufficiently stable estimation of the time lag.

$$\Delta t = \frac{1}{k} \sum_{i=1}^k t_{local,i} - t_{central,i} \quad (4.2)$$

Such fluctuations may occur due to particularities of the data transmission software stack or due to radio frequency noise during signal transmission.

When tagging data, each of the sensors adds the local delta to their local time to estimate the global time of measurement.

$$\hat{t}_{central} = t_{local} + \Delta t \quad (4.3)$$

Afterward, the tagged time series of the different sensors can be globally aligned. Without loss of generality, the calculation rule is explained for two time series. For more time series, the calculation rule can be performed pairwise until all time series are merged. For two sensor time series the alignment of their data frames can be performed as follows:

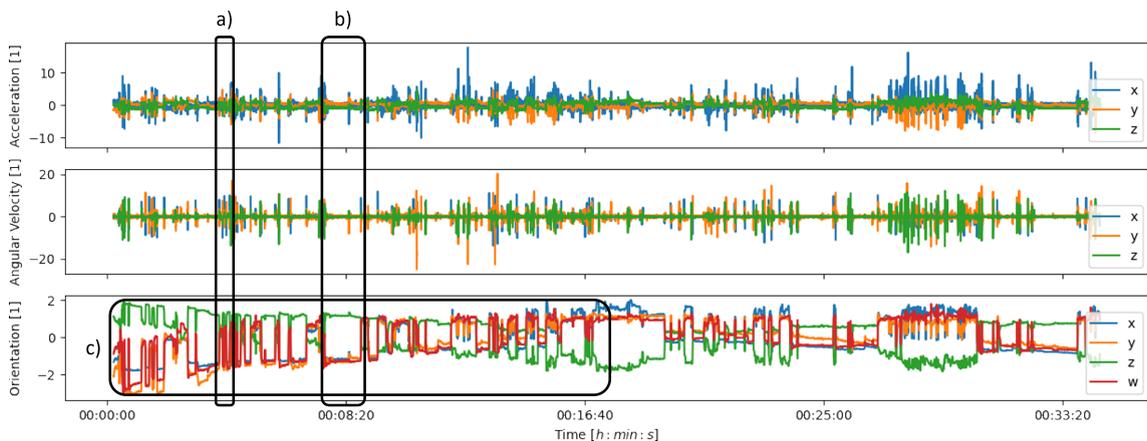
1. Trim both time series to the smallest common start and end timestamp. Different start and end timestamps are expected, as the moment of subscription to the sensor's notifications is expected to be subject lag.
2. All non-monotonic timestamps are removed. Non-monotonic timestamps can occur when the variations in the timelags between sensor and central device are too large for the averaging mechanism to handle. That is when the fluctuation is larger than the sampling rate of 50 Hz and thus the order of timestamps is no longer monotonic. However, when averaging over 10 timestamps such error could only be observed about every 2000 samples in practice. Therefore, the loss of data is accepted.

3. The time series are joined based on the timestamps of the sensor data. For each sample from the first sensor, the last sample from the second sensor whose timestamp is less than or equal to the timestamp is matched. (see also [47]). This work joins the data on the timestamps of the glasses sensor. This choice is purely by convenience.

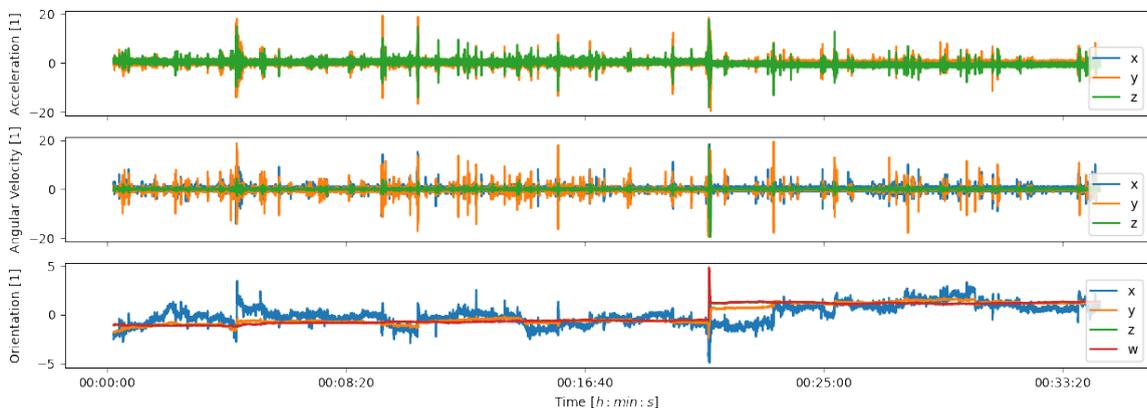
## 4.5 Feasibility

This section takes one concrete raw sensor time series example to demonstrate relevant characteristics of the raw data. Thereby, this section proves the overall feasibility of the sensing approach and shows important features of the data that influence the signal processing approach.

The concrete example is a roughly 30-minute long section of a 2-hour long recording. During the recording, the user performed unsupervised desk work. Figure 4.4 shows the raw motion sensor time series of the glasses and chair sensor. Note that the orientation data has four axes. This is because this work uses quaternions to represent the orientation. Quaternions provide a variety of advantages over other representations for rotations such as avoiding singularities and numerical advantages. The motion data is normalized for easier comparability of the data. The environmental data of the glasses sensor during this recording can be found in the appendix (Figure B.1c).



(a) Raw motion data of the glasses sensor



(b) Raw motion data of the chair sensor

Figure 4.4: Example for raw motion data during unsupervised desk work. The upper plot shows the motion data of the glasses and the lower plot of the chair sensor. The annotated shapes mark particular features of the data which are further explained in Figure 4.5, Figure 4.5 and Figure 4.7.

Looking at the raw data of the head-mounted sensors, three particularly notable characteristics of the dataset can be seen:

1. The acceleration and angular velocity are mostly static with distinct peaks whenever the user changes posture (Figure 4.5). The peaks are particularly clear to see looking at the angular velocity. Depending on the posture before and after a movement the height of the peaks and the affected axes during change of posture vary. The posture changes can also be seen in a change in the orientation of the sensor.

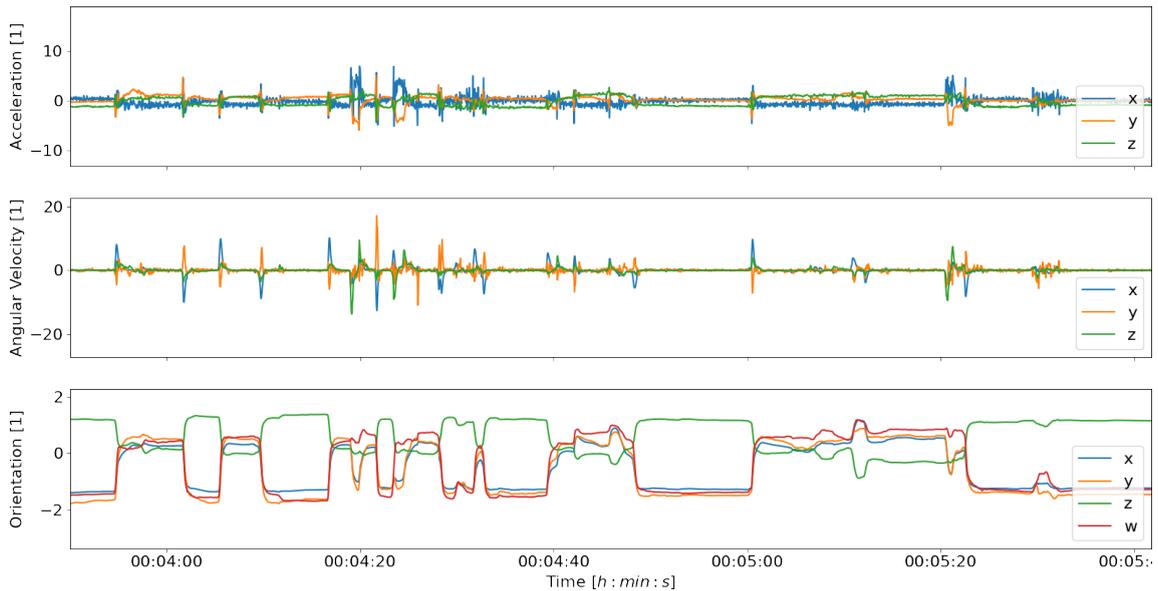


Figure 4.5: Example for motion data of the glasses sensor during posture changes. The data is mostly static with distinct peaks whenever the user changes posture

2. There are periods where the user does not change posture at all (Figure 4.6). These periods can be one or even up to multiple minutes long. In these periods the sensor data is mostly static.

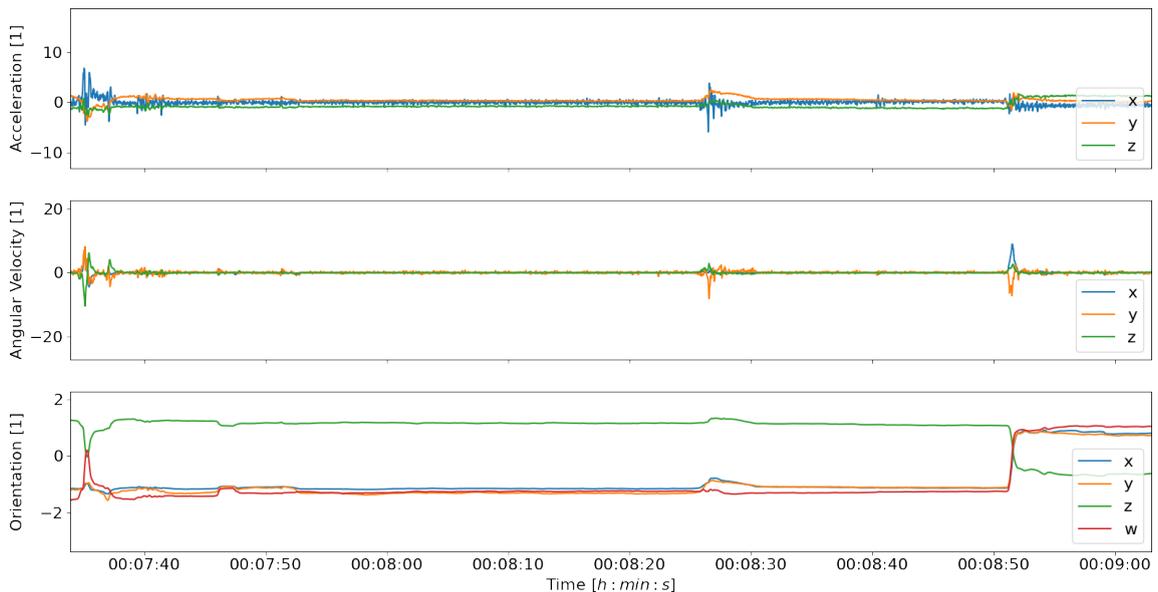


Figure 4.6: Example for motion data of the glasses sensor between posture changes. Between posture changes the data is mostly static.

- Looking at the orientation data over longer periods, it can be seen that the sensor orientation drifts over time (Figure 4.7). This indicates the accumulation of sensor errors over time in the orientation sensor fusion algorithm. Therefore, an approach for posture extraction over longer periods of time needs to account for the accumulation of sensor drift.

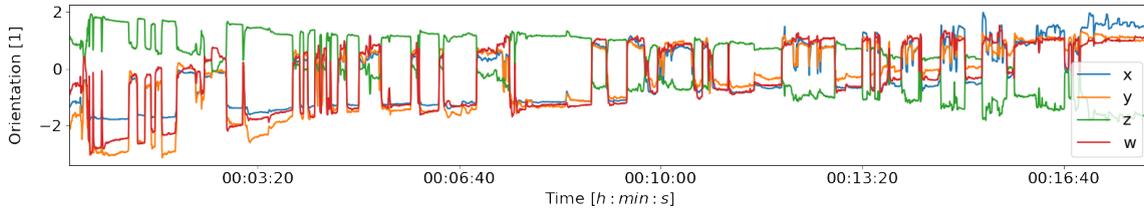


Figure 4.7: Example for sensor drift in the raw orientation data of the glasses sensor. Over sensor errors accumulate in the orientation data fusion.

The inspection of raw sensor data reveals the typical mostly static sensor time series with distinct peaks in acceleration and angular velocity, as well as the change in orientation whenever the user changes posture. This shows the general feasibility of the data to characterize the posture of the user. However, further processing is necessary to extract the actual posture from the raw sensor time series data.

## 4.6 Processing Pipeline Overview

Further processing is necessary to derive the posture from the motion data collected with the sensor setup. This section outlines the processing pipeline for deriving and assessing posture based on the sensor data. The following chapters then explain the different modules of the pipeline in-depth.

This work uses a modularized approach that divides the overall processing pipeline into seven separate modules. Figure 4.8 gives a graphical overview of the modularized pipeline.

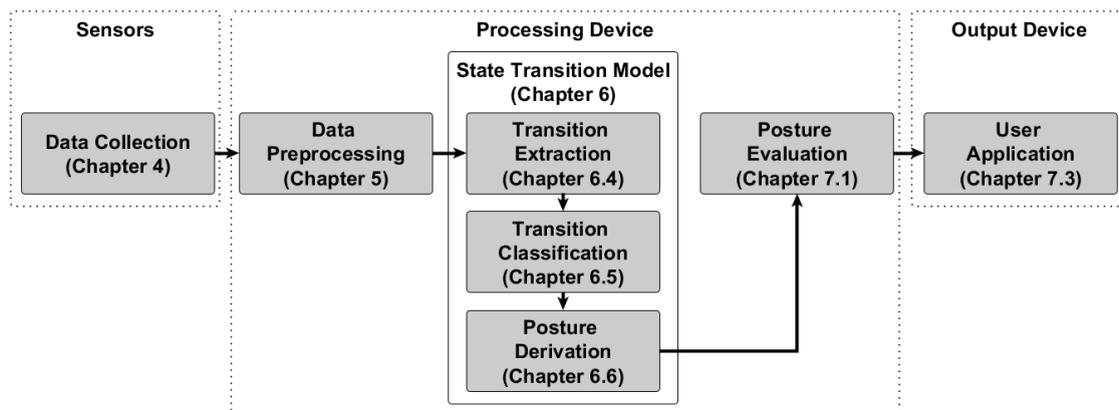


Figure 4.8: Overview of the processing pipeline. The gray blocks denote the modules of the pipeline. The arrows represent the data flow through the modules. Dotted borders mark the devices on which the modules are located. Note that one single physical device may take the role of multiple of the devices.

- The first module of the pipeline is the data collection module. This module measures relevant movements of the user and jointly collects the data on the processing device. Chapter 4 already outlined the sensor setup for data collection. Section 9.1 goes into detail about the practical data collection setup and process.

- The data preprocessing module prepares the collected raw data for further processing. Chapter 5 explains the applied signal processing methods and different approaches for the extension of the dataset.
- The state transition model (Chapter 6) performs the actual extraction of the posture from the sensor time series. This novel domain-specific machine learning approach builds on classifying and aggregating interesting events in the sensor time series when the user changes posture. The state transition model itself consists of three sub-modules:
  - The transition extraction module detects the moments in the time series where a change of posture occurs. Around each of these moments, a window of the sensor time series is extracted. Section 6.4 describes this work’s approach for the extraction of the transitions.
  - The transition classification module determines the change before and after every transition using the extracted transition windows. Section 6.5 elaborates on the chosen classification approach.
  - The posture derivation module takes the classified sequence of transitions between postures and aggregates the transitions to derive the actual posture after the transitions. Section 6.6 presents the derivation of posture.
- The posture evaluation module assesses the derived posture with respect to the desk work ergonomic risk factors. Section 7.1 explains the calculation of comprehensive posture tendency and activity scores.
- Finally, the user application provides the user immediate action recommendations for improvement based on their assessment as well as the possibility to review their ergonomic assessment which is elaborated in Section 7.3.

This work chooses the approach of modular decomposition for engineering purposes. Each of the modules has clear boundaries and expected behaviors at design time. This allows for the design and evaluation of each individual module which simplifies the overall system design. This also makes tracking system performance easier. In contrast to an end-to-end approach, the modularized approach allows modifying or substituting each module with limited effects for the overall system. This is particularly important for systems using machine learning, as even small changes in the dataset or preprocessing may have an unforeseeable impact on the model performance. With the clear module boundaries, performance changes can be investigated for each individual module and therefore possible issues are easier to detect and address.

As Figure 4.8 indicates, the modules of the processing pipeline may be distributed over three devices: The sensors, which collect movement data, the processing device that collects the data from the sensors, derives the posture from the data and evaluates the posture, and the output device on which the user application is implemented. It is important to note that these three devices do not have to be physically different devices. In particular, this work envisions implementing the posture extraction directly on the sensor. For example, smart glasses may perform both the role of one sensor and the processing device.

## 4.7 Summary

This chapter introduced a sensor solution using two IMUs placed on the frame of wearable glasses and on the armrest of the desk chair to sense posture in the context of desk work ergonomics. The investigation of an example recording showed the general feasibility of the sensing approach to sense relevant movements. An outline of the further processing pipeline to derive and assess posture based on the sensor data was given. The following chapters explain the different modules of the pipeline in-depth.



# Chapter 5

## Data Preprocessing

Data preprocessing is an essential technique for machine learning as the capabilities of the machine learning model to extract information from the dataset strongly depend on the quality of data. The following sections describe different signal processing techniques to improve data quality (Section 5.1 and Section 5.2) and data augmentation as a tool to extend the dataset (Section 5.3).

### 5.1 Denoising

Like all signals measured by real sensors, the data measured by the MEMS sensors contains noise and jitter due to for example thermal effects [48]. Practical experimentation shows that the presence of noise can negatively impact the performance of machine learning models. Therefore, it is important to reduce the noise in the data set. At the same time, noise reduction should preserve the general signal characteristics so that no relevant information is lost.

This work uses a moving average filter to filter noise. Practical experiments have shown that a simple moving average filter with a window size of 20 samples = 400 ms performs best. Other possible noise filtering approaches are using an exponential moving average filter or wavelet denoising. However, in practice these methods performed worse.

The moving average with a window size of  $k$  samples at time  $t$  is calculated as:

$$MA_{k,t} = \frac{1}{k} \sum_{i=t-k+1}^t x_i \quad (5.1)$$

The calculation of successive moving average values can be efficiently realized by using the fact that at every step a new value  $x_{n+1}$  enters the sum and the oldest value  $x_{n-k+1}$  drops out. This allows the cheap real-time computation of the moving average filter. Therefore, this approach is well suited for the application in on-device signal processing where compute and power limitations are relevant.

$$MA_{k,t+1} = MA_{k,t} + \frac{1}{k}(x_{n+1} - x_{n-k+1}) \quad (5.2)$$

## 5.2 Normalization

While it was shown that it is not strictly necessary to normalize inputs for a neural network, practical experience has proven that it can make training faster and minimize the risk of getting stuck in local minima [49]. Normalization of the features also makes comparing the general signal shape of sensors with different scales easier. Therefore, this work performs standardization (also called Z-score normalization) on the input data before feeding it into neural networks. Unless mentioned otherwise, plots in this work show the normalized sensor time series. Therefore, the axis of upcoming plots with the measurements is unitless (indicated with "[1]").

Standardization scales the dataset  $X = (X_i)_{i=1}^n$  by removing the empirical mean of the dataset  $\mu$  and dividing by the empirical standard deviation of the data  $\sigma$ :

$$\mu = \frac{1}{n} \sum_{i=1}^n X_i \quad (5.3)$$

$$\sigma = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (X_i - \mu)^2} \quad (5.4)$$

$$X' = \frac{X - \mu}{\sigma} \quad (5.5)$$

## 5.3 Data Augmentation

In the presence of sufficiently large datasets, neural networks have been shown to perform remarkably well in classification tasks. However, in many real-world tasks, the amount of labeled data available may be limited due to the cost of collecting and labeling data. At the same time, neural networks are prone to overfitting when fed with too few training examples. One way to combat this and make the model more robust to overfitting is to increase the amount of available data by synthetically adding variations of existing data to the dataset, which is called data augmentation.

This work uses a selection of techniques to manipulate existing data to synthetically create data that is representative of actual recorded data. In particular, the data augmentation techniques manipulate the transition data in a way that preserves the identity of the transitions. This allows the model to learn to recognize and correctly classify such invariant transformations. The techniques used in this work are jittering, scaling, magnitude warping, and time warping. Figure 5.1 visualizes the four different techniques applied to example time series.

- **Jittering** adds random noise to the signal. In particular, this work adds zero-mean normally distributed noise as this noise is similar to thermal noise in the sensors.

$$x' = x + \mathcal{N}(0, \sigma^2) \quad (5.6)$$

- **Scaling** takes the original time series and multiplies it with a number that is normally distributed with the mean 1. These kinds of variations may occur when a user performs similar movements but at different speeds.

$$x' = x * \mathcal{N}(1, \sigma^2) \quad (5.7)$$

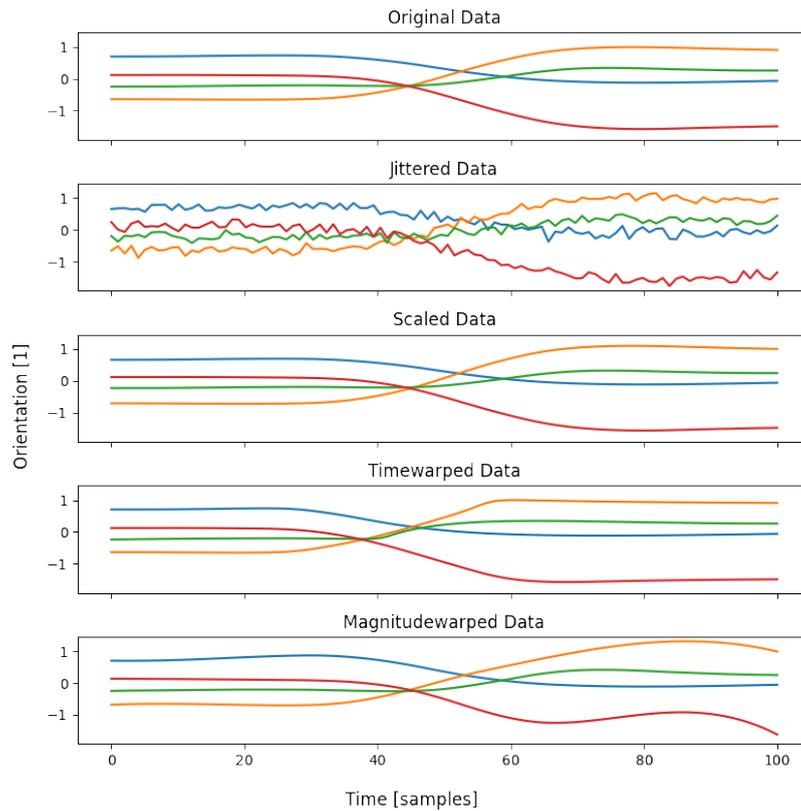


Figure 5.1: Plots of different time series data augmentation techniques for an example time series. The plots show the time series of the orientation sensor on the glasses for one transition window. The original time series (top row) gets manipulated by jittering, scaling, magnitude warping, and time warping. The resulting modified time series are plotted below the original one.

- **Time warping** manipulates the data by smoothly distorting the time intervals between samples using normally distributed distortions. This work uses a cubic spline with randomly generated anchor points to create a smooth distortion curve. Figure 5.2 gives an example of a time distortion curve. The time warped data is obtained by linearly interpolating the original data according to the distorted timescale. Similar modifications as time warping occur when a user makes the same movement at different speeds. Time warping also simulates distortions in time similar to when the time synchronization of sensors is imperfect.
- **Magnitude warping** changes the data by multiplying the original data with a smooth curve that is centered around one. As a curve, this work uses a cubic spline with several generated anchor points at random positions and with random magnitudes determined by a normal distribution centered around one. Figure 5.3 shows an example of such a curve. Such changes to movements may be seen when the same movement gets executed with a slightly varying trajectory.

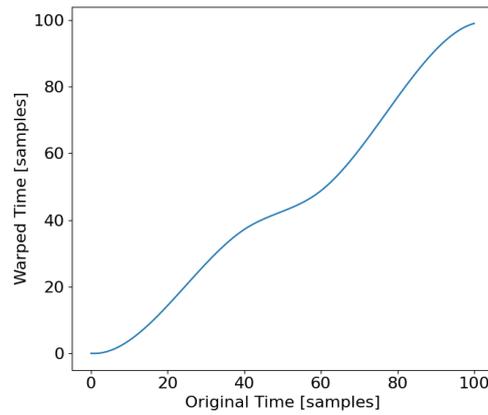


Figure 5.2: Example time warping time distortion curve. The resulting time series is obtained by distorting the time axis according to the time distortion curve and by linearly interpolating the data according to the scaled time.

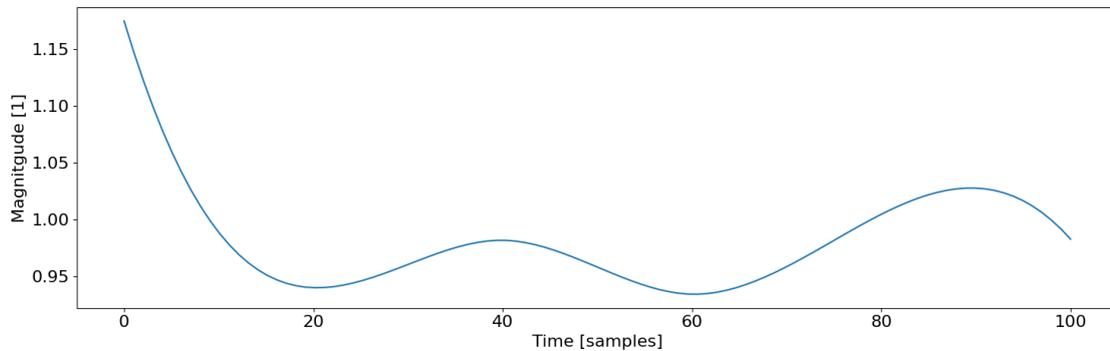


Figure 5.3: Example magnitude warping scaling curve. The resulting time series is obtained by multiplying the original data magnitude with the scaling curve.

## 5.4 Summary

This chapter presented denoising and normalization as relevant signal preprocessing techniques in the context of this work. It also demonstrated data augmentation as a tool to extend the collected dataset in semantically meaningful ways. After the preprocessing, the data is ready for the actual extraction of posture which is explained in the next chapter.

# Chapter 6

## State Transition Model

This work presents a novel domain-specific AI approach for the derivation of posture from the preprocessed sensor data, the state transition model. Section 6.1 explains the fundamental idea of the approach. Section 6.2 clarifies the notion of states and transitions. Section 6.3 explains the technical background for the state transition model. The following three sections then explain the three main steps of the state transition model in detail: The transition extraction detects transitions and extracts small sensor time series windows where a change between posture occurs (Section 6.4). The transition classification distinguishes the extracted transitions to infer the posture before and after each transition (Section 6.5). Finally, the posture derivation derives the user’s posture sequence from the sequence of classified transitions (Section 6.6).

### 6.1 Fundamental Idea

Previous approaches directly derive posture from the sensor time series by moving a window over the sensor time series. They then directly classify the posture for each window. However, as the user sits mostly statically during desk work, they only rarely change posture. This work takes this observation to create a domain-specific machine learning model for posture extraction from the time series data. The fundamental new idea is to only consider the sensor data where the user is actually moving. This approach classifies these movements as transitions between postures. The actual posture is then determined in a second step by aggregating the transitions between the postures. Figure 6.1 visualizes the fundamental idea of the state transition model.

This work refers to this approach as ”state transition model” for postures derivation. The reason for this is that the fundamental idea is to indirectly derive the static state, here posture, by extracting and aggregating the transitions between postures. This approach is inspired by pedestrian dead reckoning (PDR) where also, starting from a known position successive position displacements are added up [50] to derive the position over time.

This approach has a variety of advantages:

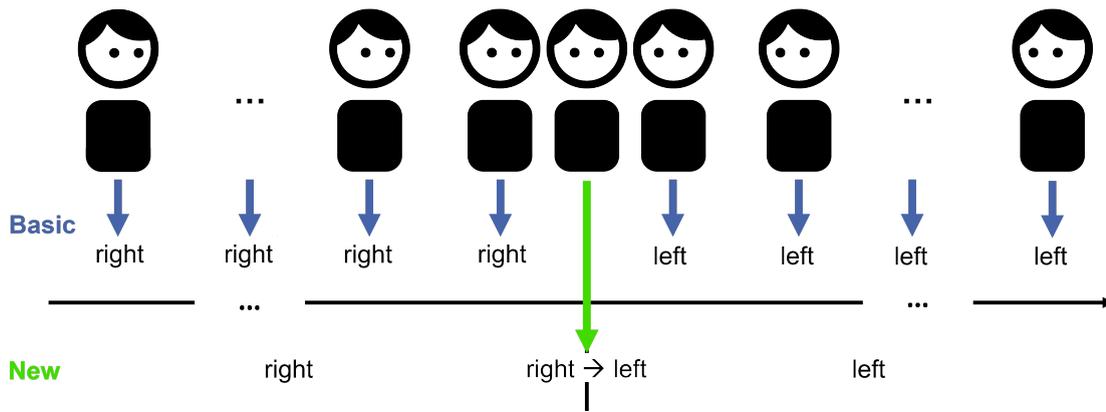


Figure 6.1: Fundamental idea of the state transition model for posture extraction. The image shows an example where a user changes their posture from looking right (from the user’s point of view) to looking left. Previous approaches for posture extraction continually look at windows of the time series data and directly classify the posture for every window (indicated with the blue arrows). This work’s approach instead only looks at the transition between postures and classifies the transition (green arrow). Then, in a second step, the approach derives the actual posture from the extracted transition. Note the broken-up time axis as postures are potentially being held over long periods of time.

1. It is more compute-efficient as the primary signal processing only needs to be active when there is relevant movement and can be inactive otherwise. Therefore, power can be saved by reducing the signal processing when no relevant movement is detected. This helps to implement the system directly on mobile devices where power consumption is relevant.
2. Instead of continuously monitoring the time series the model only investigates the transitions. Consequentially, similar to PDR, the method is more robust against the accumulation of sensor drift over time since the time durations under consideration are much shorter.
3. In contrast to most of the sensor time series where the user sits statically, the accelerometer and gyroscope can measure the dynamic movement during transitions. This means that the transition classification does not have to rely only on the static orientation, but can also use acceleration and angular velocity.
4. Domain knowledge about the posture transitions and common movement sequences can be incorporated into the classification of transitions and derivation of posture from them. This use of domain knowledge improves the overall system performance.

## 6.2 States and Transitions

To better understand the state transition model, this section clarifies the terms state and transition in the context of the state transition model. This work interprets posture as a continuous-time dynamic system with a state that evolves over time and transitions that modify the state at discrete times. Figure 6.2 visualizes the idea of the state-transition understanding of posture.

In terms of this state transition understanding of posture, the state is the overall posture of the user with regard to the four degrees of freedom introduced in Section 3.3. Following that notation, the overall state is defined by the combined state of the four DoFs (see Section 3.3). This work uses two notations to refer to a state: First, as a textual notation,

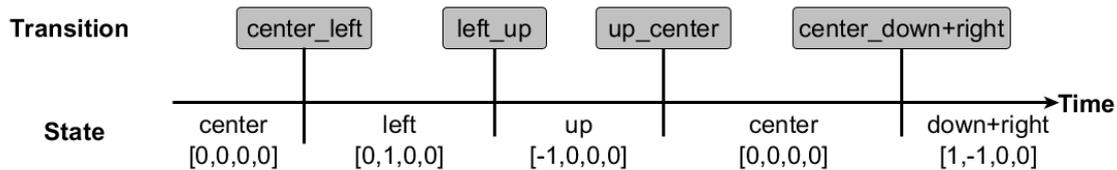


Figure 6.2: Visualization of the state transition understanding of posture. This work interprets posture as a continuous state system with posture as its state. Transitions occur at discrete times and modify the state. The transitions (gray boxes) are characterized by the state before and after the transition. The states are either characterized by a textual notation or by a vector notation indicating the orientation in all four degrees of freedom.

the overall posture is described by joining the DoFs with a plus, e.g. *up+left* describes the posture where the head angle is *up* and the head direction is *left*. For simplicity, *center* is omitted. Instead *center* describes the posture where all four DoFs are neutral. To describe a transition from one posture to another posture the notations of the two postures are joined by an underscore, e.g. *center\_left* is the transition from the neutral posture to the *left* head direction.

And second, the vector notation  $p \in \{-1, 0, 1\}^4$  where each of the four entries describes the state of one DoF. The order of entries is given by the order of mention in Section 3.3 1 and  $-1$  describe the two non-neutral postures and  $0$  describes the neutral posture. As an example, *left* =  $[0, 1, 0, 0]$  and *down+right* =  $[1, -1, 0, 0]$ .

## 6.3 Technical Background

Before going into detail about the state transition model, the following section presents some technical background for the model. This section first introduces convolutional neural networks (Section 6.3.1) and then Hidden Markov Models (Section 6.3.2).

### 6.3.1 Convolutional Neural Networks

Neural networks are used useful in a wide range of applications. Many applications use neural networks to classify data. Convolutional neural networks (CNNs) are a specialized kind of neural network that is designed to process data that has a grid-like topology. CNNs are primarily known for their use in computer vision tasks. However, CNNs also have been successfully applied to time series data which can be thought of as a 2D grid consisting of one dimension for the features per timestep and one dimension for time. The following section introduces the basic structure and properties of CNNs.

Although the exact structure of a CNN varies depending on the application, most CNNs follow a basic structure: They consist of a series of convolutional layers and pooling layers for feature extraction. After that, the extracted features are flattened to one dimension and fed into fully connected layers for the actual classification. Depending on the application the number and size of the individual layers may vary [51].

The basic building block of a convolutional neural network is the convolutional layer which gives CNNs their name. Convolutional layers build upon the mathematic operation convolution [51]. Given functions  $i$  and  $k$  the convolution is defined as:

$$f(t) = (i * k)(t) = \int i(a)k(t - a)da. \quad (6.1)$$

CNNs extend that notation to typically two dimensions and use the discrete convolution. In CNN terminology  $f$  is usually referred to as the input,  $g$  as the kernel, and  $f$  as the feature map. Usually, multiple kernels are used [51]. The output of a convolutional layer can be computed as:

$$f(i, j) = (i * k)(i, j) = \sum_m \sum_n i(m, n)k(i - m, j - n) \quad (6.2)$$

Figure 6.3 visualizes the idea of 2D-convolution.

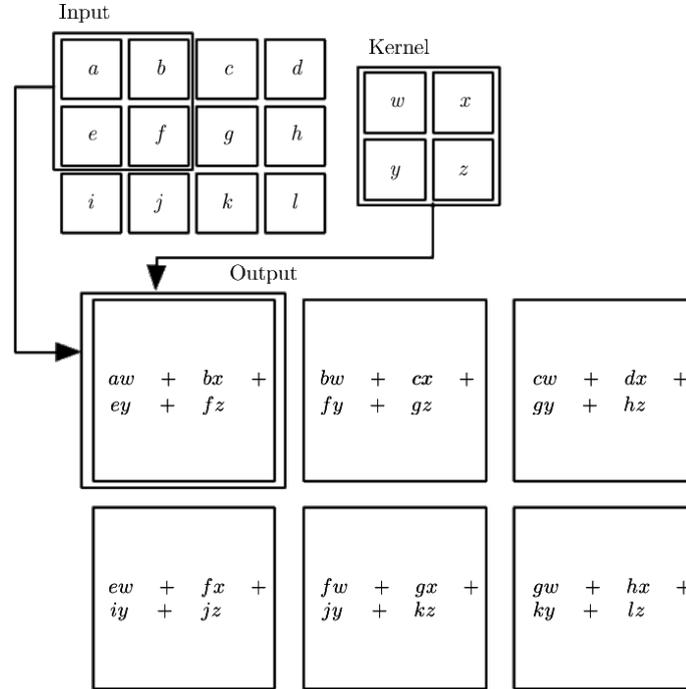


Figure 6.3: Visualization of 2D-convolution (image taken from [51]). The kernel is moved over the input. At each position of the kernel, the output is computed as the sum of the pairwise product of the kernel and input at that position.

The utilization of convolutional layers yields three advantages over basic fully connected neural networks [51]: First, instead of describing the interaction between every input and output unit, CNNs typically use kernels much smaller than the input to detect features in a local neighborhood, which results in sparse representations. This means CNNs need to store fewer parameters and need fewer operations. Second, the same kernel is used to calculate features over the whole input (also called parameter sharing). Therefore, this reduces the storage requirements of the model. And finally, the convolution is invariant to certain transformations which are common in image and time series data, e.g. translation in space or time respectively.

Pooling layers are another building block of CNNs. Pooling performs a similar operation to the convolutional layer in that it moves over the input and investigates rectangular neighborhoods. However, instead of convoluting the neighborhood with a kernel to compute the output, a pooling function reports a summary statistic of the neighborhood. For example, max pooling reports the maximum value within the rectangular neighborhood. Pooling accumulates knowledge from a local neighborhood and reduces the data size for later layers. The summarization of the pooling layer also helps the network to learn to become invariant for certain transformations [51].

The ability of CNNs to detect local dependencies while being memory and computationally efficient results in variations of CNNs being applied for classification tasks on low-power devices [52, 53]. This trend is accompanied by research on optimizing CNNs in particular for the application on low-power devices [54].

### 6.3.2 Hidden Markov Models

A Hidden Markov Model (HMM) is a stochastic model that is used to model processes with unobserved (or hidden) states and observed emissions [55]. One particularly common task of HMMs is to derive the most likely sequence of hidden states from an observed sequence of emissions. A variety of applications such as for example speech recognition [56] and bioinformatics [57] use Hidden Markov models.

The HMM is based on augmenting the Markov chain. A Markov chain models the probabilities of sequences of random variables  $Q = q_1, \dots, q_i$ . Markov chains build upon the assumption that the probabilities of the random state variables only depend on the last state. This assumption is called the Markov assumption [55].

$$P(q_t = a | q_1, \dots, q_{t-1}) = P(q_t = a | q_{t-1}) \quad (6.3)$$

Formally, a Markov chain is defined by three components:

- A set of **states**  $S = \{s_1, \dots, s_n\}$
- A **transition probability** matrix  $T$  where every entry  $T_{i,j} = P(q_t = s_j | q_{t-1} = s_i)$  describes the probability of moving from state  $s_i$  to state  $s_j$
- An **initial probability** distribution  $I$  where every entry  $I_i = P(q_0 = s_i)$  describes the probability that the Markov chain starts in state  $s_i$

The Hidden Markov Model extends that notion by introducing a sequence of observations  $O = o_1, \dots, o_j$  that draw from a vocabulary  $V = \{v_1, \dots, v_m\}$ . The so-called emission probabilities  $E_i(o_t) = P(o_t | q_t = s_i)$  then describe the probability to observe an emission  $o_t$  in state  $s_i$  [55].

The task of deriving the most likely sequence of states  $Q$  from a sequence of observations  $O$  is usually called Decoding. One particularly important algorithm for solving the decoding problem is the Viterbi algorithm. Given an HMM  $\lambda$  the Viterbi algorithm recursively calculates the most probable state sequence [55]:

$$v_t(j) = \max_{q_1, \dots, q_{t-1}} P(q_1 \dots q_{t-1}, o_1 \dots o_t, q_t = j | \lambda) \quad (6.4)$$

To do this, the Viterbi algorithm uses dynamic programming to calculate the current most probable state sequence given the values of the last timestep and the currently observed emission  $o_t$ .

$$v_t(j) = \max_{i=1}^n v_{t-1}(i) T_{i,j} E_j(o_t) \quad (6.5)$$

## 6.4 Transition Extraction

The first step in the state transition model is to detect when the user changes posture. Afterward, a transition window is extracted around each detected transition for further processing. The following section describes the detection mechanism for transitions and how transition windows are extracted around them (Section 6.4.1), how mistakenly detected noise-actions can be filtered (Section 6.4.2) and an approach to separate transitions that are close in time (Section 6.4.3).

### 6.4.1 Transition Detection

The initial step in the state transition model is to detect when a relevant movement happens. Figure 6.4 shows an example of the extracted transition times for a sequence of posture changes.

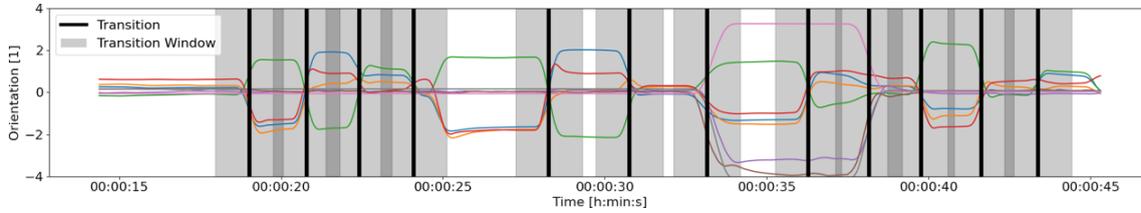


Figure 6.4: Example for transition extraction for a sequence of posture changes. The plot shows the orientation data during a sequence of movements. The vertical lines denote the detected transitions.

Interesting changes of posture express themselves as changes in orientation of the head and the torso along different axes. The gyroscope measures the angular velocity of the sensor. Therefore, utilizing the gyroscope is a natural fit to detect these posture changes in orientation while being robust to other undesired movements. In particular, this work uses the extreme values of the gyroscope to identify transitions. The extreme values indicate the most significant motion and thus indicate where a relevant change of posture is probable.

Let  $g^g = (g_i^g)_{i=1}^n \in \mathbb{R}^3$  and  $g^c = (g_i^c)_{i=1}^n \in \mathbb{R}^3$  be the gyroscope sensor time series of the sensor on the wearable glasses and of the chair respectively. Then for  $i = 1, \dots, n$  calculate the movement signal as follows:

$$\tau(g_i^g, g_i^c)_\alpha = \|g_i^g\|_2 + \alpha * \|g_i^c\|_2 \quad (6.6)$$

The parameter  $\alpha$  adjusts how much significance is given to the chair signal. Practical experimentation determined  $\alpha = 0.5$  to perform well. Figure 6.5 shows an example of the movement signal with annotated transition times.

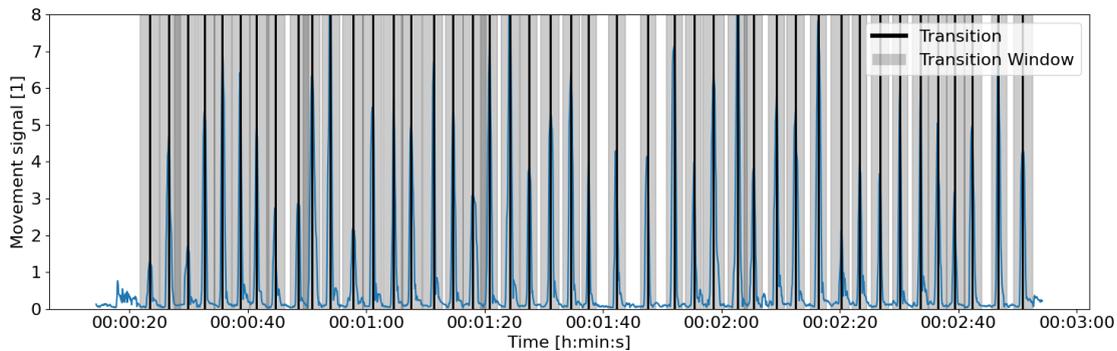


Figure 6.5: Visualization of the transition extraction using gyroscope extreme values. The plot shows the movement signal  $\tau(g_i^g, g_i^c)_\alpha$  with  $\alpha = 0.5$  for a series of random desk work movements. The horizontal lines show the derived transition times. Around each signal transition, the gray area denotes the transition window. The movement signal shows clear peaks whenever the user changes posture.

The local maxima of the movement signal are candidates for transitions. Local maxima are calculated by a simple comparison of neighboring values. Despite signal processing, both gyroscope signals may contain noise. Also, the user may not move smoothly between

different poses. As a result, the transition candidates contain additional local maxima per actual transition. Therefore, filtering of the candidates is necessary to remove potential duplicate transitions. This work uses two different methods of filtering:

1. Transition candidates with a topological prominence of less than 1 are removed. The topological prominence of a peak is defined by the vertical distance between a peak and the lowest contour line that encircles it and no higher peak [58]. The filtering of peaks with low topological prominence only keeps peaks that are significant compared to their surrounding. Peaks with smaller topological prominence are likely caused by noise or unwanted movement. Figure 6.6 visualizes the idea of topological prominence.

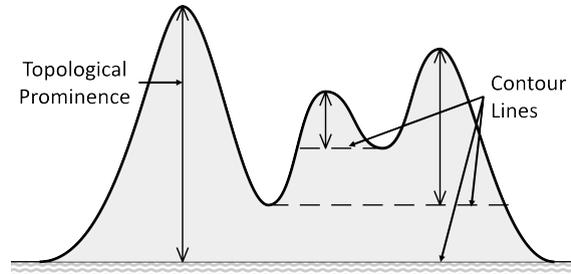


Figure 6.6: Visualization of topological prominence (image taken from [59], modified). The vertical arrows show the topographic prominence of the three visible peaks. The dashed lines show the lowest contours. Peaks with high topological prominence stand out particularly clearly from their surroundings.

2. When two or more transition candidates are closer than 50 samples = 1 s, the smaller peaks are removed until the condition this not true anymore.

After determining the transition times, for each transition a window of 100 samples = 2 s centered around each transition is extracted. These time series windows serve as the basis for further processing.

#### 6.4.2 Noise-Transition Classification

Despite the fit of the transition detection approach in Section 6.4.1 to detect relevant movements, some of the detected transition candidates may still contain non-transition noise actions. This may be the case due to noise in the signal. Another reason is that the user performs movements which result in rotational velocities despite no actual change of posture. To eliminate non-transition noise actions this work introduces a noise classifier. This classifier estimates the probability of a candidate containing an actual true action. Transitions with a low probability of being an actual change of posture can be simply filtered out. Therefore, this approach acts as an optional filtering step after the basic transition extraction. Figure 6.7 illustrates the filtering of noise actions using an example time series.

This work formulates the detection of noise actions as a binary classification problem with one class being "true action" and the other being "noise action". A convolutional neural network (CNN) with a sigmoid output layer performs the classification. The model consists of a single convolutional layer with max pooling before the model flattens the data for classification. The model architecture is intentionally simple to minimize the compute and power requirements of the model. An additional dropout with a dropout frequency of 0.5 reduces the likelihood of overfitting to the training data. For training Adam [60] is used as an optimizer and categorical cross-entropy as a loss function. Early stopping acts

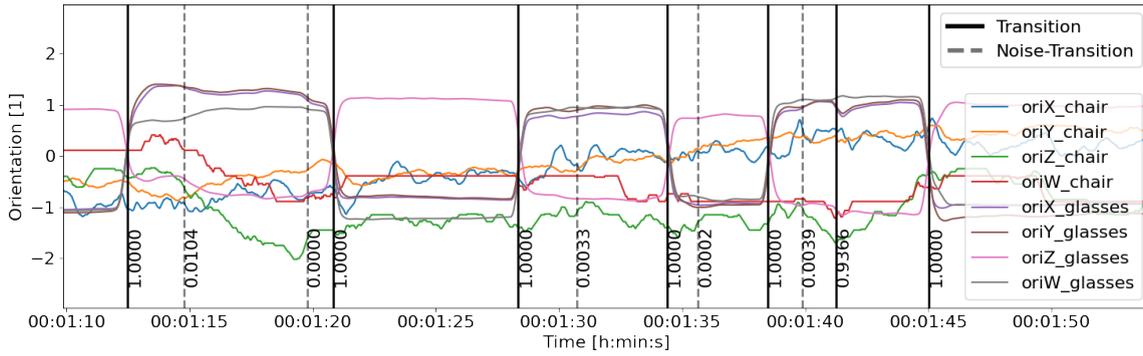


Figure 6.7: Visualization of the filtering of noise-transitions. The plot shows the time series of the orientation sensors during random desk work activities. The solid lines indicate the extracted transitions. Each transition is annotated with the output probability to be a true transition. Transitions with a probability of  $p < 0.5$  are identified as noise-transition and are consequentially masked (dashed lines).

as an additional regularization method. See Figure 6.8 for a visualization of the model architecture.

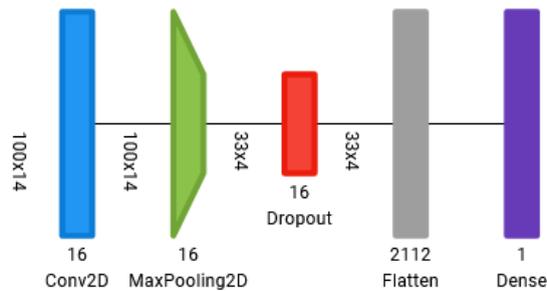


Figure 6.8: Architecture of the noise-transition classifier. The plot shows the layers of the convolutional neural network architecture and the data flow through them. The boxes denote the layers of the network. If applicable, each layer is annotated with the input shape and the number of channels. The noise-transition classifier performs a binary classification of whether an extracted transition is a true transition or not.

### 6.4.3 Transition Separation

Working at their desks, there are passages in which users change between postures in quick succession. Such passages occur, for example, when a user repeatedly looks at the keyboard while typing. Another example is working with multiple contents on different sections of the screen, especially when working with multiple screens. Figure 6.9 shows an example of a time series where the user changes postures in rapid succession.

When the distance of neighboring transitions is less than the size of the transition window, these transition windows overlap. This means that artifacts of a transition may be present in the transition window of a neighboring transition. If the overlap is small, the impact of the artifacts for transition classification is likely to be negligible. However, problems can arise when transitions are so close together that the full neighboring transition spills into the adjacent transition window. The following section discusses an approach that masks transitions when they spill into the transition window of an adjacent transition. The approach tries to find the largest subwindow of a transition that does not contain any

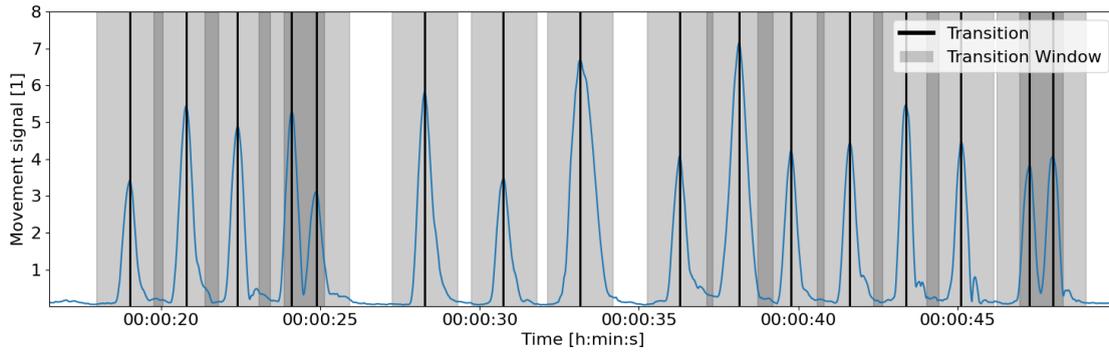


Figure 6.9: Example for transitions where the transition windows overlap. The time series shows the movement signal during a passage where the user repeatedly looks at the keyboard while typing. The horizontal lines mark detected transitions. The transition windows are shown as the gray background around each transition. Note that the background gets darker where windows overlap.

adjacent transitions. The remainder of the window, where there would be artifacts of other transitions gets masked out. Figure 6.10 visualizes the masking of transition windows.

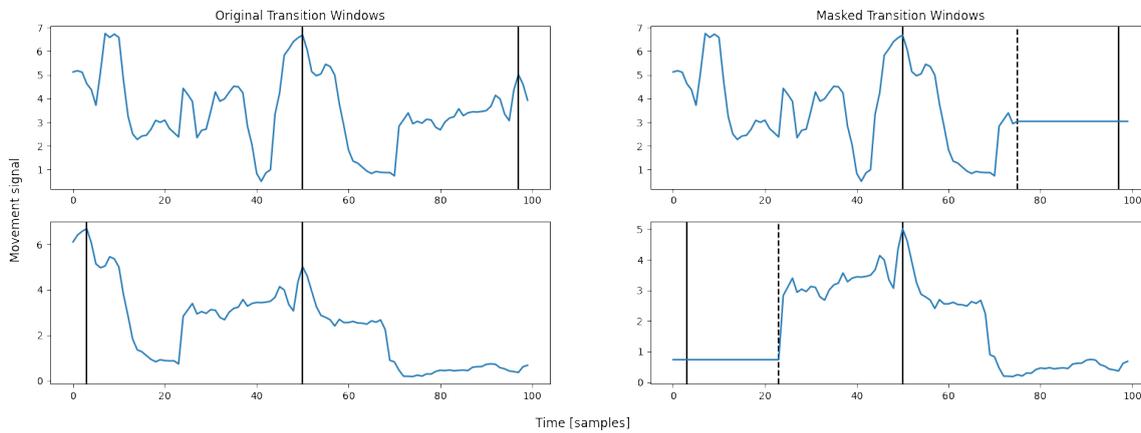


Figure 6.10: Example for the masking of close transitions. The plots show the movement signal time series for two transition windows (upper and lower plots). In these two examples, neighboring transitions are closer than 50 samples. The left plots show the transition windows before the masking and the right plots after masking. The solid horizontal lines mark the transitions and the dashed ones the optimal derived borders for masking.

The following approach can be used to mask overlapping transitions:

1. Find the transitions that overlap. Transitions overlap when their distance is smaller than half the transition window size. Then a transition is included in a neighboring transition's transition window.
2. For all overlapping transitions perform the following steps:
  - a) Calculate the movement signal  $\tau$  for the transition window of the current transition (see Equation 6.6).
  - b) If the transition before the current transition overlaps, determine the optimal lower border for masking. Therefore, calculate the index of the sample with the smallest movement signal between the current and the overlapping transition.

If the index is less than  $\alpha$  samples away from the current transition instead keep at least  $\alpha$  samples.

- c) If the transition after the current transition overlaps, calculate the optimal upper border similar to the step before.
- d) Mask the transition window. Therefore, set all samples before the lower border to the data at the lower border. Intuitively, this masks the overlapping transition in the first half of the transition window. Process the second half of the transition window similarly by setting all samples after the upper border to the value at the upper border. If only one transition overlaps into the window, mask accordingly only on one side.

The parameter  $\alpha$  balances between finding an optimal point for masking and keeping enough characteristics of the transition for further processing. Practical experimentation showed that  $\alpha = 20$  works best with a window size of 100 samples.

## 6.5 Transition Classification

After the extraction of the transitions, the next step of the state transition model is to find out which transition occurred based on the sensor time series. Therefore, this work takes the fixed-sized sensor time series window around a transition and classifies the transition. Figure 6.11 shows an example of a sequence of posture changes annotated with the classified transitions.

In theory, with only the four regarded dimensions with three sub-states each, there are already  $3^4 = 81$  states and  $(3^4)^2 = 6561$  possible transitions. In practice, however, many of the theoretically possible transitions rarely or never occur. This work uses this observation and reduces the number of regarded classes. In particular, a hand-selected subset of 42 transitions is used. The 42 transitions are hand-selected based on manual inspection of the user data. Regardless of how the transitions are chosen exactly, the idea of reducing the actual number of possible states and transitions based on their probability is an important tool for reducing model complexity in practice.

This work uses a Convolutional Neural Network (CNN) with a softmax output to classify the transition windows. CNNs are useful for the classification of time series data particularly for short dependencies as they are present in the only 100 sample long transition windows. A CNN is intentionally chosen over other neural networks such as long short-term memory (LSTM) networks as CNNs are also able to capture local relationships in time while being less computationally complex and easier to train [61]. For a visualization of the transition classifier CNN architecture see Figure 6.12.

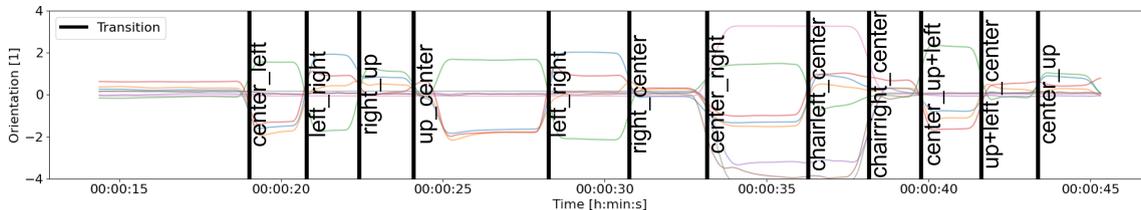


Figure 6.11: Example transition classification for a sequence of posture changes. The plot shows the orientation data during a sequence of movements. The vertical lines denote the detected transitions. Each extracted transition is annotated with the result of the classification.

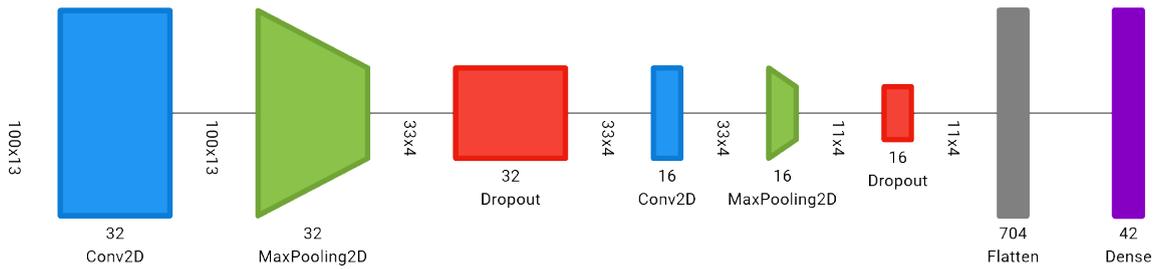


Figure 6.12: Architecture of the transition classifier. The plot shows the layers of the convolutional neural network and the data flow through them. The boxes denote the layers of the network. If applicable, each layer is annotated with the input shape and the number of channels. The transition classifier distinguishes between 42 possible transitions.

The model takes a transition window which is a matrix of size  $[100 \times \#features]$  and outputs a vector of size  $[\#classes]$ . The output denotes the probability of the transition window to be to each of the regarded transitions. The fixed size of the windows allows simply using a CNN without the need for a fixed size embedding. The model architecture uses a series of two convolutional and max-pooling layers before flattening the features and applying a dense layer with softmax activation to get the output probability for each class. This architecture comprising convolutional and max-pooling layers before flattening the features and classifying is a very common CNN architecture [51]. Notably, the chosen architecture is relatively shallow comprising only five computational layers. This choice is intentional to minimize computational and memory requirements for a potential on-device implementation. Additional dropout layers with a dropout frequency of 0.5 act as a regularization measure.

For the training, Adam is used as an optimizer and categorical cross-entropy as the loss function. As an additional regularization method early stopping with a patience of 10 epochs is used.

## 6.6 Posture Derivation

The posture derivation module takes the sequence of extracted postures to extract the sequence of postures from them. Section 6.6.1 describes the use of HMMs to perform this derivation. After that, Section 6.6.2 presents the idea of enriching the model with knowledge about the application.

### 6.6.1 Posture Model

The transition classifier derives which transition between postures occurred in a sensor time series. Thus, applying the transition classifier to all detected transitions extracts the sequence of transitions between postures in the time series. To derive the actual postures, further work is necessary. The following section describes the use of HMMs to derive the posture sequence from the observed transition sequence. Figure 6.13 shows an example where a sequence of posture changes is annotated with the derived postures.

The simplest approach would be to add up the detected transitions. However, if transitions are confused, sequences can occur in which it is not clear which posture is true. For example, if two consecutive transitions are *center\_left* and *center\_right* it is unclear whether the posture between the transitions is *left* or *center*. Therefore, simple aggregation is impossible if the transition classifier does not recognize the transitions 100% correctly, which is unrealistic in practice. To solve this issue, this work uses Hidden Markov Models

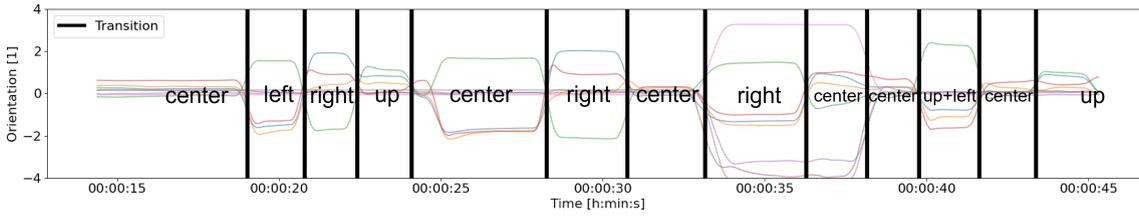


Figure 6.13: Example posture derivation for a sequence of posture changes. The plot shows the orientation data during a sequence of movements. The vertical lines denote the detected transitions. Between two transitions the posture is annotated.

(HMMs) as a stochastic model to derive the sequence of postures from the sequence of transitions under uncertainty about the correctness of the transition sequence. In doing so, the HMM-based approach allows using empiric domain knowledge about common posture sequences during desk work to correct errors in the transition sequence. In particular, this work translates the sequence of observed transitions into the sequence postures such that for each transition the posture after the transition is derived.

In terms of Hidden Markov Model terminology, the setup is the following: The different postures are the states of the HMM. The transitions as classified by the transition classifier provide the observed emission sequence. Using this setup, solving the decoding task with the Viterbi-Algorithm estimates the most probable sequence of postures. As the posture consists of four individual degrees of freedom, this work uses four HMMs, each deriving the sequence of posture states for one degree of freedom. To derive the overall posture the four decoded sequences of the four HMMs are simply stacked along the time axis. This results in the vector notation for the overall posture.

To fully specify the HMMs the transition probability matrix, emission probabilities, and the initial probability distributions need to be defined. By convention, each sequence is expected to start in the center. Therefore, the initial probability distribution is simply one for the *center* state of each HMM and zero for all other states. The transition probability matrix and the emission probabilities are estimated using a maximum likelihood estimation using training data.

Formally, the maximum likelihood estimation of the HMM parameters is described using the following formulas. Let  $A_{kl}$  be the number of transitions from state  $k$  to  $l$  and  $E_k(t)$  be the number of times that a transition  $t$  is observed in state  $k$  in the training data. Then the following equation estimates the transition probabilities  $a_{kl}$  and emission probabilities  $e_k(b)$  [62]:

$$a_{kl} = \frac{A_{kl}}{\sum_{s \in S} A_{ks}} \quad (6.7)$$

$$e_k(b) = \frac{E_k(b)}{\sum_{t \in T} E_k(t)} \quad (6.8)$$

To avoid zero probabilities for transitions and emissions that have not been observed during training additive smoothing is applied [62]. In particular, this work uses Laplace smoothing where  $\alpha = 1$ .

$$a_{kl} = \frac{A_{kl} + \alpha}{\sum_{s \in S} A_{ks} + \alpha} \quad (6.9)$$

$$e_k(b) = \frac{E_k(b) + \alpha}{\sum_{t \in T} E_k(t) + \alpha} \quad (6.10)$$

### 6.6.2 Application Bias

This work chooses the specific application of Hidden Markov Models (HMM) to the translation of a sequence of posture changes back to the underlying sequence of postures. This makes it possible to embed knowledge about the postures and transitions into the model by modifying the transition and emission probability matrices. In particular, it is possible to use knowledge about the similarity of postures and states to influence the emission probabilities of the model in such a way that misclassifications of the transition classifier can be compensated.

First, this work proposes to increase the emission probability of transitions in states that are similar to the state after the transition, e.g. when the classifier falsely classifies *left+lefttilt\_left* as *left+lefttilt\_center*. Increasing the probabilities for transitions with similar endpoints compensates for the fact that the transition classifier may slightly misclassify the endpoint of the transition. To characterize similarity between two transitions, this work calculates the mean difference of the vector representations of the end postures  $p_a, p_b \in \{-1, 0, 1\}^4$ :

$$similarity = \frac{1}{4} \sum_{i=1}^4 |p_{a,i} - p_{b,i}| \quad (6.11)$$

And second, the emission probabilities of transitions are increased in states where there is another transition that goes in the same direction. For example, *left\_center* ends in center and shares the direction with *center\_right*, so increase the emission probability of *center\_right* in the state *center* slightly. This idea compensates for the possibility that the transition classifier detects the direction of a transition but fails to calculate the correct start and endpoint.

The idea of introducing an application bias is particularly interesting for transitions where no knowledge about their occurrence during desk work is available. Therefore, there is also no information to estimate the HMM parameters for these transitions. The application bias allows an estimation of the parameters so even artificially generated transitions can be included.

## 6.7 Summary

This chapter introduced the state transition model-based approach for extracting postures. This approach reconstructs the posture sequence from the collected sensor data by extracting and aggregating transitions between postures. However, knowing the user's posture and how it changes over time is of limited use. Only the interpretation and evaluation of the data concerning ergonomic risk factors give the posture data a meaning. The following chapter demonstrates how to put the posture information into the context of desk work ergonomics and how the user can benefit from it.



# Chapter 7

## Ergonomic Application

With the derived posture sequence of the user, it is possible to assess the posture to evaluate the user's personal ergonomic risk situation. Section 7.1 describes the evaluation of postural tendency and postural activity based on the user's posture. Section 7.2 shows how to also include environmental factors in the ergonomic risk assessment. Finally, Section 7.3 presents notifications and data presentation as two mechanisms to provide the user with the value from their personal ergonomic assessment.

### 7.1 Posture Evaluation

Two ergonomic risk factors are particularly important for the sitting posture: Sustained poor posture and prolonged static periods of sitting. For each of the two risk factors, this work defines a feature that assesses the overall posture.

- The **postural tendency (abbrev. posture)** refers to the long-term trend in the distribution of head and trunk orientations. This work uses the moving average over the orientations to characterize posture. A high postural tendency indicates a poor imbalanced posture while a low postural tendency suggests a good neutral posture. Therefore, this work assesses sustained poor posture with postural tendency.
- **Postural activity (abbrev. activity)** measures the spread in the distribution of head and trunk orientations. Practically, this work uses the moving variance to measure activity. A high postural activity indicates that the user is sitting dynamically while a low activity indicates static periods of sitting. This work uses postural activity to assess prolonged periods of static sitting.

The following calculation rule derives a score between 0 and 100 for both the postural tendency and the postural activity from a user's overall posture. For this purpose, this work evaluates each degree of freedom of the posture (see Section 3.3) individually. This allows a differentiated assessment of the risk situation. The following calculation rule determines the posture and activity score for one particular degree of freedom:

1. Map the orientations to a numbers  $(x_i)_{i=1}^n \in \{-1, 0, 1\}$  such that 1 and  $-1$  describe the two non-neutral postures, and 0 describes the neutral posture. For example, looking at the direction of the head, "left" would be mapped to  $-1$ , "right" to 1, and "center" to 0. This is equivalent to taking one of the four entries of the vector notation from Section 6.2.
2. Calculate the rolling mean and the rolling variance over time windows  $w = (w_j)_{j=1}^m$  of size  $t_{win}$  in time and length  $k = t_{win}/t_{sample}$  in samples where  $t_{sample}$  is the sampling period. Let  $x = (x_i)_{i=1}^n \in \{-1, 0, 1\}$  be the time series of the orientation. Then the rolling mean  $\mu(w_j)$  and the rolling variance  $\sigma^2(w_j)$  can be calculated as follows:

$$\mu(w_j) = \frac{1}{k} \sum_{x_i \in w_j} x_i \quad (7.1)$$

$$\sigma^2(w_j) = \frac{1}{k-1} \sum_{x_i \in w_j} (x_i - \mu_k(w_j))^2 \quad (7.2)$$

3. Scale the rolling mean and rolling variance to the range 0 to 100. For posture, a low mean is desirable as it indicates a neutral posture. Inversely for activity high variance is preferable, as this ensures the presence of activity. Additionally, this work introduces cutoff values  $c_{l,p}$ ,  $c_{h,p}$ ,  $c_{h,a}$ , and  $c_{l,a}$ . A mean below  $c_{l,p}$  is considered perfect posture resulting in a score of 100 (variance above  $c_{h,a}$  as perfect activity similarly). Also, a mean above  $c_{h,p}$  is considered bad enough posture (variance below  $c_{l,a}$  as bad enough activity similarly) to always receive a score of 0. Between the cutoff values, a cosine is used to smooth the distribution. In Figure 7.1a the posture score scaling for posture is shown with  $c_{l,p} = 0.05$  and  $c_{h,p} = 0.3$ . Figure 7.1b similarly shows the scaling for the activity with  $c_{l,p} = 0$  and  $c_{h,p} = 0.5$ . Formally, the following equations calculate the posture and activity scores:

$$posture(w_j) = \begin{cases} 0 & |\mu(w_j)| \geq c_{h,p} \\ \cos\left(\frac{|\mu(w_j)| - c_{l,p}}{c_{h,p} - c_{l,p}}\right) * 50 + 50 & c_{l,p} < |\mu(w_j)| < c_{h,p} \\ 100 & |\mu(w_j)| \leq c_{l,p} \end{cases} \quad (7.3)$$

$$activity(w_j) = \begin{cases} 0 & \sigma^2(w_j) \leq c_{l,a} \\ 100 - \left(\cos\left(\frac{\sigma^2(w_j) - c_{l,a}}{c_{h,a} - c_{l,a}}\right) * 50 + 50\right) & c_{l,a} < \sigma^2(w_j) < c_{h,a} \\ 100 & \sigma^2(w_j) \geq c_{h,a} \end{cases} \quad (7.4)$$

Based on empiric evaluation this work proposes  $t_{win} = 30mins$  as the size of the regarded time windows and  $c_{h,p} = 0.3$ ,  $c_{l,p} = 0.05$ ,  $c_{h,a} = 0.3$  and  $c_{l,a} = 0$  as initial starting values for the score cutoff. Figure 7.2 shows an ergonomic assessment of an example recording using the above approach.

The ergonomic risk situation of a user is highly individual and subject to a variety of personal factors. Therefore, personalization of the ergonomic risk assessment is necessary. Adjusting the regarded time window  $t_{win}$ , as well as the introduced cutoffs  $c_{h,p}$ ,  $c_{l,p}$ ,  $c_{h,a}$  and  $c_{l,a}$  allows the personalize the assessment to the personal precondition of a user. In addition, individual adjustment of the cutoff values for each of the regarded dimensions provides a more differentiated assessment of the various postures. The configuration of these parameters can be created by the user, adjusted based on medical assessments, or automatically derived by an algorithm based on the measurements made for a user.

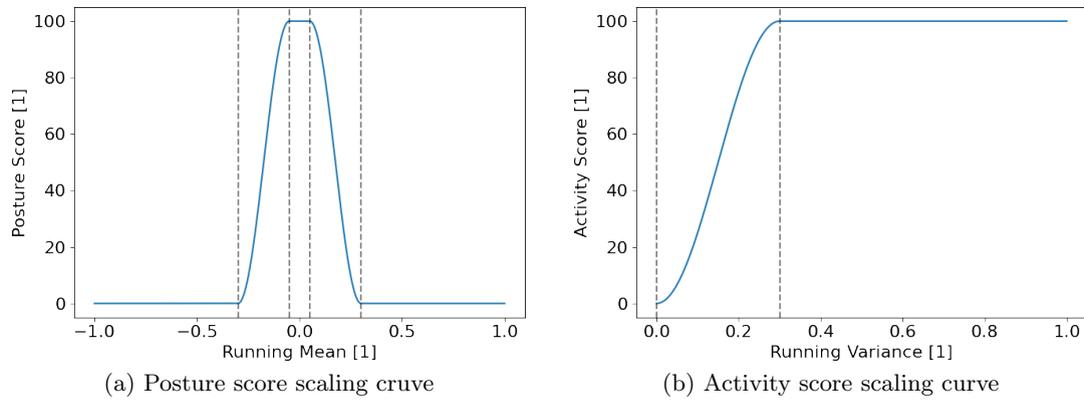


Figure 7.1: Plots of the scaling curves for the posture and activity scores. The left plot shows the scaling curve for the posture ( $c_{h,p} = 0.3$  and  $c_{l,p} = 0.05$ ). The right plot shows the activity scaling curve ( $c_{h,a} = 0.3$  and  $c_{l,a} = 0$ ). The posture score falls off with increasing mean as a high mean indicates an unbalanced posture. For the activity score, a high variance denotes a high score as a high variance indicates an active posture. Note the symmetry around the  $x = 0$  of the posture curve.

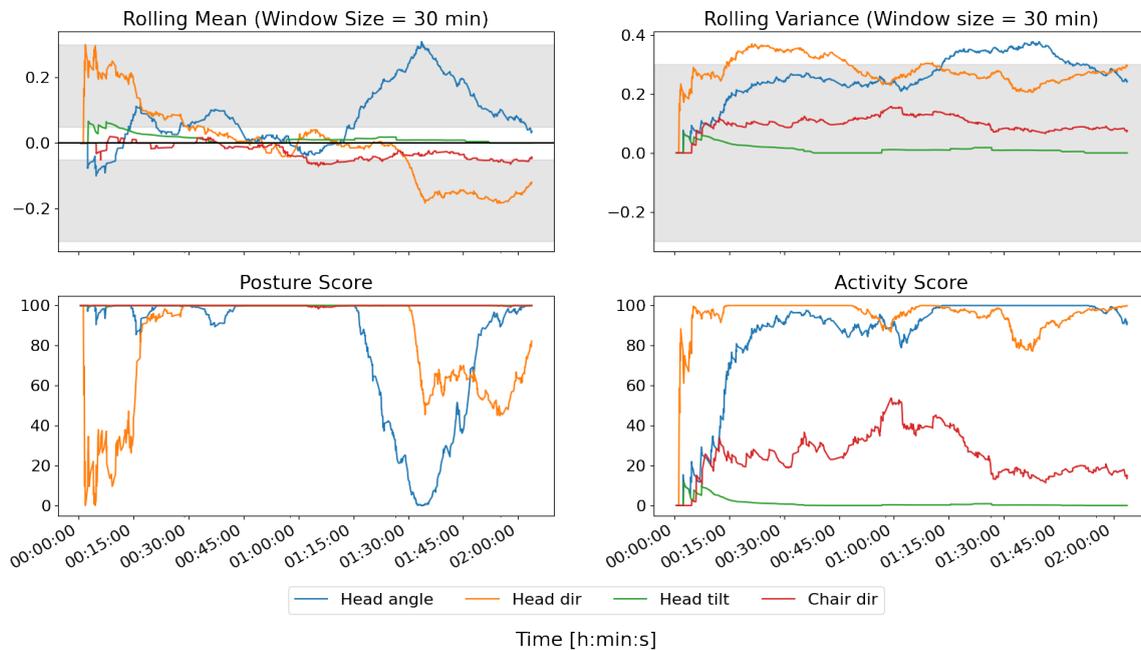


Figure 7.2: Example ergonomic scoring. The upper plots show the rolling mean and variance for an example recording while the lower plots show the corresponding posture and activity score after scaling ( $c_{h,p} = 0.3$ ,  $c_{l,p} = 0.05$ ,  $c_{h,a} = 0.3$  and  $c_{l,a} = 0$ ). The white areas in the upper plots denote the areas above and below the cutoff values while the gray hue indicates the area between the cutoff values.

## 7.2 Environmental Evaluation

Apart from measuring the user's movement and the derivation of their posture, the sensor setup also measures various environmental factors that are relevant for office ergonomics. This work also includes these factors in an overall ergonomic assessment of the desk workplace. As noted in Section 4.1 the sensors used in this work measure humidity, air pressure, and volatile organic compounds (VOC). The desk workplace setup can be estimated using the virtual orientation sensors. The absolute head angle at the default position estimates the head-to-monitor view angle. Combined with knowledge about the used screens, the maximum head direction angle can be used to estimate the eye-to-monitor distance. Adding additional sensors for noise and light to the sensing setup is possible to provide an even broader environmental assessment. The evaluation approach listed below can be applied to any other factors as desired.

Table 7.1: Guide values for desk environmental and desk setup ergonomic factors [37].

Factor	Guide Value
Noise	<55 dB
Light	≈500 lx
Relative Humidity	50 %–60 %
Temperature	20 °C–27 °C
VOC	<3.0 mg/m <sup>3</sup>
Eye-to-monitor View Angle	0°–30°
Eye-to-monitor Distance	500 mm–700 mm

This work compares the measured values of the environmental factors to guide values to assess ergonomic risk. Table 7.1 gives an overview of typical guide values for the desk setup and environment [37]. Small deviations indicate low ergonomic risk while abnormalities mark high potential risk. Similar to the posture score, the assessment assigns a score between 0 and 100, depending on the match with the guide values. Again, the sliding mean with a window size of 30 minutes of the sensor time series is used (Equation 7.3). In this context, the sliding window approach smoothes the assessment of the environmental factors as many environmental factors undergo large short-term changes, e.g. light changes when someone walks by the desk or there is sudden noise during printing.

Let  $c_l$  be the lower limit of the guide value range and  $c_h$  the higher limit. Note that some guide values only have one limit. Others like noise even only have a singular target value. In that case, the missing limit is simply set to the maximum value, or both limits are set to the same value respectively. The following equation then calculates the environmental score:

$$environment(w_j) = \begin{cases} 100 - \frac{c_l - \mu(w_j)}{\alpha * c_l} & \mu(w_j) \leq c_l \\ 100 & c_l < \mu(w_j) < c_h \\ 100 - \frac{\mu(w_j) - c_h}{\alpha * c_h} & \mu(w_j) \geq c_h \end{cases} \quad (7.5)$$

The parameter  $\alpha$  adjusts how harshly deviations from the guide value should be punished. For different environmental factors, different values for  $\alpha$  are possible.

## 7.3 User Application

The assessment of a user's ergonomic risk factors already offers some insights into their personal ergonomic risk. However, the real value is in providing this information to the

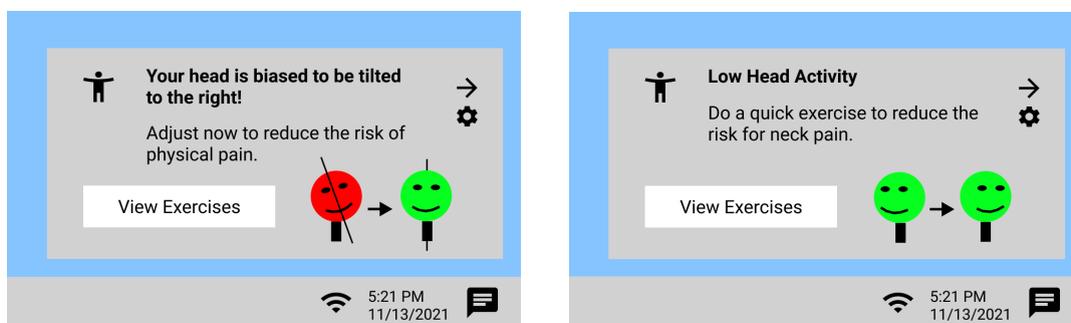
user, helping them to better understand and positively change their personal ergonomic risk factors in the workplace. This section presents two mechanisms to make the information gained by the ergonomic assessment accessible to the user. Notifications provide immediate action propositions to the user whenever their ergonomic scores suggest arising issues (Section 7.3.1). And an application presents the data to the user (or a medical professional) in a way that allows the user to gain a detailed understanding of his personal ergonomic risk factors (Section 7.3.2). These approaches enable the user to identify ergonomic problems earlier and correct them directly, improving their quality of life.

### 7.3.1 User Notification

A first mechanism to provide value to the user is to make immediate recommendations for action as soon as ergonomic problems occur. For this purpose, the user receives notifications on his screen or cell phone as soon as his posture scores fall below defined threshold values and thus indicate an ergonomic risk. This approach has two goals: First, the user is informed about problems directly during the execution of his work and not afterward. This allows them to understand their behavior directly on the spot and not to look back on it retrospectively. And more importantly, the action suggestions provide the user with immediate and professional instructions on how to improve their posture immediately. Therefore, the user can eliminate posture problems before problems or diseases arise from them.

To identify when to send out notifications, this work proposes a thresholding approach on the ergonomic scores from Section 7.1. Whenever either the activity or posture score falls below a certain threshold a notification is sent. The threshold can be defined in an application by the user or a medical professional who supervises the user. Additionally, the thresholds can be individually defined for each of the different posture dimensions. This allows for the personalization of the notification mechanism to the personal conditions of the user.

This work proposes two output devices for the notifications. Since this work considers work at a computerized desk workstation, a natural option is to display notifications on the user's computer. This approach integrates into the user's workflow and minimizes the barrier to interacting with the notification because there is no need to change the device. Figure 7.3 shows two examples of notifications on a desktop application.



(a) Desktop notification issued because of poor sitting posture. (b) Desktop notification issued because of low sitting activity.

Figure 7.3: Examples for desktop notifications. The notifications contain both the cause for the notification as well as an immediate action proposition. Note the use of visual cues to indicate the proposed action.

Another option is to provide notifications on a smartphone. This method increases the barrier of interaction with the notification by crossing device boundaries. However, users

are more accustomed to receiving and interacting with notifications on the smartphone, which could in turn encourage interaction. Figure 7.4 examples for notifications on a smartphone.

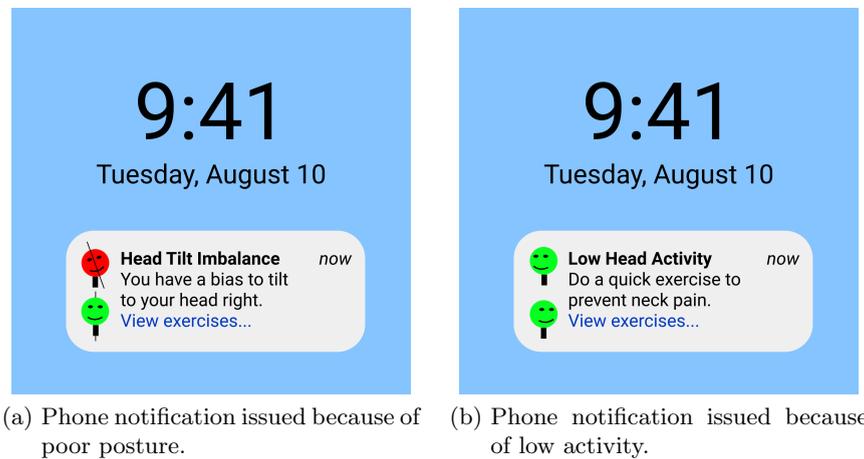


Figure 7.4: Examples for smartphone notifications. The notifications contain both the cause for the notification as well as an immediate action proposition. Note the use of visual cues to indicate the proposed action.

The notification itself contains both the cause for the notification and an immediate action proposition. By showing the user not only the problem but also the immediate action, they can improve their posture without having to stop and think. An additional proposition may take the form of suggestions on how to change posture or what body parts to move. The notification may refer to additional optional exercises for affected body parts. These exercises are specifically tailored to the user's personal disposition based on the posture evaluation.

This work proposes to not only use textual notifications but also to provide sketches or small videos as visual cues for suggested actions. The visual cues help the intuitive understanding of the action proposition. Again, the goal is to improve posture while minimizing the necessary mental distraction from the actual work.

### 7.3.2 Data Presentation

In addition to immediate action propositions, presenting the collected data to the user also provides value. Thus, this work proposes an application that presents the user with his current ergonomic risk situation and shows time trends of his data. This gives the user a holistic picture of their current situation and helps them to identify both positive and negative trends. This additional information further helps uncover additional problems and reveal sources of certain phenomena. The application for data presentation is inspired by very popular fitness applications that collect fitness data and present it in a similar way [30, 31, 32]. The application consists of a home screen, which displays the user's current situation, and detail screens, where the user can click to view the development of his values over time.

The home screen shows the scores calculated in Section 7.1 to give the user an intuitive understanding of his situation. See Figure 7.5a for a mockup of the home screen. The scores are separated into posture and activity to indicate the relevance of both factors for a good posture. The home screen also shows the scores determined by the environmental evaluation in Section 7.2. For simplicity, the home screen only shows a selection of environmental factors, and different factors are cumulated such as the desk setup or air

quality. Therefore, the home screen gives the user a holistic picture of their current personal postural and environmental ergonomic risk situation. The home screen also presents notifications generated by the mechanism in Section 7.3.1, to remind the user of the action proposition. A coloring of the scores similar to a traffic light visualizes the associated ergonomic risk of every score. This helps the user to intuitively distinguish between high and low ergonomic risk.

When the user clicks on a score, they see a graph that visualizes the development of their posture over time. Figure 7.5b shows a mockup of such a detail screen. The time scale is adjustable so the user can review the development both short-term (e.g. one day) and long-term (e.g. months). The detail screen adopts the traffic light coloring of the home screen to distinguish between periods of high ergonomic risk and unproblematic periods.

In addition to the usefulness of this information for the user himself, it may also be of interest to healthcare professionals. The long-term nature of work ergonomics makes it difficult to collect relevant data, as observation over such long periods of time in controlled studies is practically impossible. Similarly, treating physicians can only rely on a snapshot of the patient and their reports. In particular, a physical therapist interviewed in the context of this work stated that in some cases it can be very difficult to deduce the actual cause of the problems only from the information given by the patients and the physically recognizable characteristics. Therefore, an application that collects and presents the user's data over long periods can help both medical researchers and healthcare practitioners gain objective and long-term insights into their patient's ergonomic risk factors.

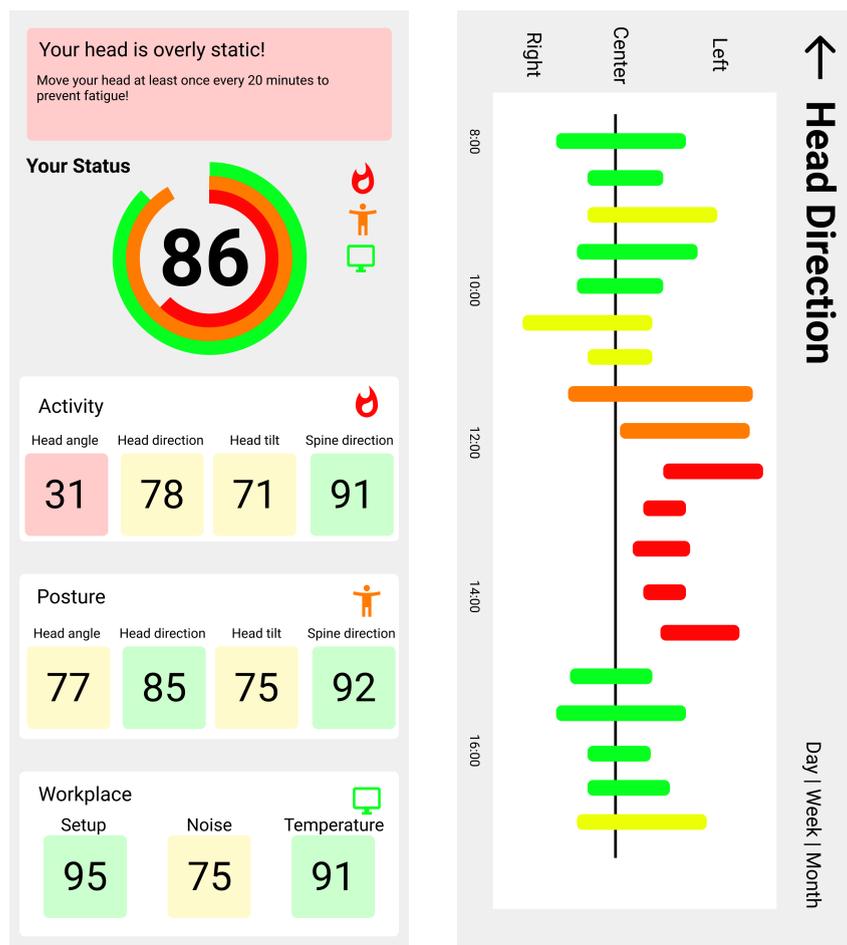


Figure 7.5: Mockups of the data presentation application. The application consists of a home screen that shows the current situation of the user and detail screens that show the development of his situation over time. The home screen shows the scores for the user's posture, activity, and workplace environment. Clicking on the scores on the home screen access the detail screens. The detail screens show the time development of the user's activity and posture. The height of the bars indicates the activity of the user, while the center of the bars indicates the posture tendency of the user.

## 7.4 Summary

This chapter demonstrated the assessment of the user's posture. An application presents the assessment to the user by recommending actions to improve their posture and presenting them their evaluation. This enables the user to actively improve their posture and understand their personal ergonomic risk. Now that this work presented the entire pipeline from sensing to an ergonomic assessment of posture, the next chapter explores the application of manifolds to ergonomic motion data as an additional technique that further improves the understanding of ergonomic motion.



# Chapter 8

## Ergonomic Movement Manifold

The sensor setup proposed in this work allows collecting large volumes of high-dimensional data. This complexity of the dataset makes it hard to find semantically interesting structures in the data and thus hinders understanding of the data. Manifold learning is an important mathematical tool that helps find semantically meaningful low-dimensional structures in high-dimensional data [63]. Previous research has shown that applying manifold learning to human motion offers promising results [64]. This work uses manifolds of ergonomically relevant movements for reasoning and explainability. Section 8.1 first explains the basic idea of manifolds. Section 8.2 explains the manifold learning approach of this work. Finally, Section 8.3 and Section 8.4 demonstrate how to use the manifold in explaining and reasoning about ergonomic movement data.

### 8.1 Manifold Basics

A manifold is a topological space that is locally Euclidean. This means that around every point, there is a neighborhood that locally appears to be the euclidean space. This abstract definition allows for a variety of very different manifolds. There are many different approaches for creating manifolds ranging from statistical approaches to using machine learning. However, typically manifolds with specific additional structures are of interest [63]:

- The so-called manifold hypothesis assumes that the data distribution over real-life data is highly concentrated. Therefore, manifold learning tries to find the low-dimensional structures in the overall data space where interesting variations of the data lie on the directions of the manifold. Such manifolds allow representing the data in terms of coordinates on the low-dimensional manifold rather than coordinates in the high-dimensional original space.
- There is interest in manifolds where traversing the locally euclidean neighborhood has semantic meaning. These manifolds allow traversing the manifold to obtain meaningful new results.

## 8.2 Learning the Movement Manifold

This work uses principal component analysis (PCA) as the manifold learning method. PCA is a mathematical method that successively decomposes a multivariate data set into a set of orthogonal components, the principal components which are linear combinations of the original dimensions. Each principal component explains the maximum variance in the dataset while being orthogonal to the previous principal components. This is equivalent to successively fitting lines that minimize the squared distance from the data points to the line. Therefore, the first principal component is the direction of greatest covariance in the data, the second the next uncorrelated direction of greatest covariation, etc. The principal components are calculated by determining the eigenvectors of the covariance matrix that correspond to the largest eigenvalues. Each principal component explains the fraction of variance in the dataset equal to the corresponding eigenvalue [65]. Figure 8.1 illustrates the fundamental idea of the PCA.

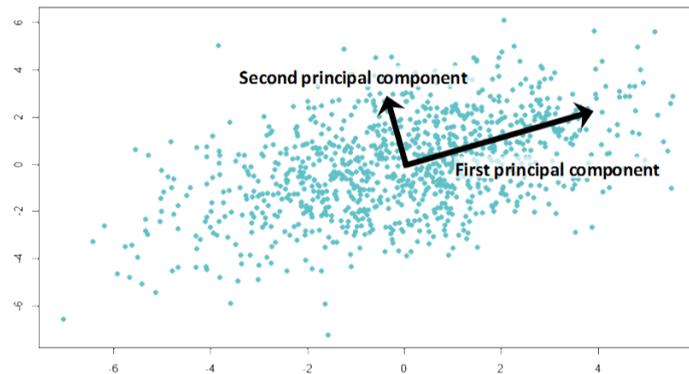


Figure 8.1: Illustration of the principal component analysis (image taken from [66]). The plot shows a toy dataset with its first two principal components. The principal component analysis iteratively determines new directions in space, the principal components. At each step, the next principal component is the linear combination of the original dimensions that explains the most variance while being orthogonal to the previous principal components.

This work projects the transition window data on the manifold spanned by the first  $n$  components of the PCA that explain at least 70% of the original dataset's variance. Similar fractions of explained variance are common in other applications [67]. To accommodate the fact that the collected data is two-dimensional (time  $\times$  features), this work flattens the data to one dimension to be able to apply PCA.

There is a variety of different existing manifold learning approaches. Despite the fact that this thesis uses PCA as the manifold learning approach, the following ideas and approaches can generally be applied to other manifold learning algorithms as well. However, some ideas may have to be modified. For example, if the chosen manifold learning approach does not create linear structures, the structures may need to be interpreted differently.

## 8.3 Explainability of Movement Data using the Manifold

The ability of manifolds to find semantically relevant structures in high-dimensional data allows using them to explain data. For this purpose, the data is mapped into the manifold. Then this work visually investigates symmetries and clusters in the projected data. These structures can reveal semantic information hidden in the original dataset. Clusters on the manifold reveal similarities between the data. Striking symmetries between different patterns suggest special relationships between the data. Depending on what patterns show symmetry, the relationships between the patterns can be explained.

Some manifold learning approaches also provide interpretability of the created manifold, e.g. the principal components of a dataset are ordered in order of decreasing explained variance. This additional information further explains the dataset structure. For example, if certain semantic structures are particularly noticeable in a projection on the first principal components these semantic structures can be responsible for the most variance in the dataset.

## 8.4 Reasoning and Data Generation using the Manifold

One of the factors that complicate the correct derivation of a user's overall posture is the fact that a user's posture consists of many independent dimensions. To cover all possible overall postures, the number of possible postures, and even more so the number of transitions between two postures is already notable with the four considered dimensions ( $3^4 = 81$  possible postures and  $(3^4)^2 = 6561$  possible transitions). Although many of the transitions are rarely or never observed in practice, collecting sufficient training data only for relevant transitions is still costly. This work proposes a novel approach that reduces the effort to collect training data by generating data for combined movements from existing data using manifolds.

The proposed approach is inspired by work on word embeddings in natural language processing. This work found that it is possible to produce semantically meaningful and comprehensible results by vector composition when linear substructures in a manifold contain relevant semantic information [68]. This work takes this idea and applies it to desk work ergonomically relevant movements. This work shows that on a manifold in which the vector combination of movements leads to semantically meaningful movements, this information can be used both to reason about the structure of data as well as to generate data. In particular, this thesis presents an approach that uses a vector combination on the manifold to combine two transitions and obtain a movement that resembles the movement when both transitions are performed simultaneously. Thus, this work explores the possibility of applying the idea of vector composition on manifolds to motions that are relevant for ergonomics.

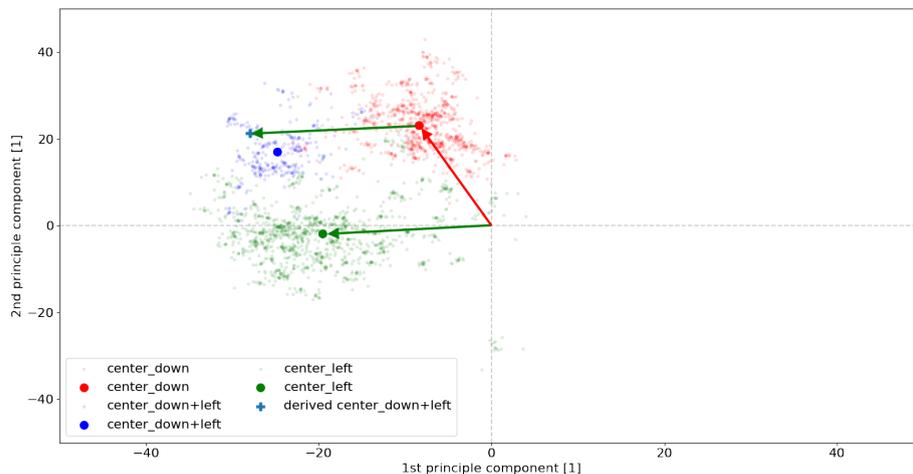


Figure 8.2: Visualization of vector combination on the manifold. The plot shows a projection of the transition data onto the first two principal components of the dataset. The scatter points denote the projections of actual recorded data (small points) and the centroids for the classes (large points). The arrows visualize the idea of using vector composition to derive the centroid of a combined transition.

Figure 8.2 provides a visualization of the fundamental idea of vector combination to obtain semantically meaningful movements. The approach performs the following steps:

1. Find a manifold where linear substructures in space represent semantically meaningful characteristics of the motions under consideration. Depending on the manifold learning approach used, interesting substructures that are not linear may appear as well.
2. Project the recorded transition window data onto the manifold.
3. Find the centroid of each transition class on the manifold as a representative of this class.
4. Combine transitions by vector combination of the vectors from the origin to the centroids on the manifold. Depending on the found substructures the necessary vector combination may be different.
5. Transform the resulting derived centroid back into the original space. This allows to obtain the transition window of the derived centroid.
6. Create a dataset from the back-transformed centroid using data augmentation techniques. The data augmentation ensures the reintroduce variation into the created data that is similar to the variation found between actual recordings of the same transition.

When the projection of the data on the manifold produces semantically interesting linear substructures, the arithmetics of the vector combination to obtain meaningful results are particularly simple. As Figure 8.2 illustrates simply adding or subtracting vectors yields meaningful results. Other manifold learning approaches may generate non-linear substructures. Then, vector combination may still produce meaningful results on these manifolds. However, different arithmetics of the vector combination may be necessary to account for the non-linearity.

## 8.5 Summary

The application of manifolds to the movement data offers the potential for explainability and reasoning about desk workplace ergonomic movement data. Interesting substructures of the data on the manifold allow explaining characteristics of the data. In particular, linear substructures allow reasoning about the data structure and for vector combination on the manifold to obtain semantically meaningful information. After Chapter 4- 8 conceptually presented various intelligent sensing techniques, the next chapter presents the evaluation of the approaches with a practical implementation using recorded data.

# Chapter 9

## Evaluation

This chapter evaluates the approaches based on a practical implementation using data recorded in a real-world desk workplace setup. Section 9.1 describes the overall data collection setup and process. After that, the different steps of the posture extraction pipeline, the transition extraction (Section 9.2), transition classifier (Section 9.3), and posture derivation (Section 9.4) are evaluated individually. Section 9.5 compares the performance of different variations of the end-to-end pipeline. Finally, Section 9.6 investigates the application of manifolds to the recorded data.

### 9.1 Data Collection

The collection of a comprehensive dataset was necessary to train and test the various approaches in the posture extraction pipeline, and to investigate the application of the manifold to ergonomic movement data. The following subsections describe the details of the data collection performed in this work. The first subsection describes the controlled desk setup for data collection (Section 9.1.1). Then Section 9.1.2 shows the general data collection process. The last section provides detailed information about the collected dataset (Section 9.1.3).

#### 9.1.1 Desk Setup

For the collection of data, this work uses a controlled desk workplace setup. The workplace consists of a dual monitor setup similar to ones found in offices. Figure 9.1 shows a sketch of the desk setup. The sensors are placed as shown in Figures 4.3a and 4.3b.

This work uses two monitors with a screen diagonal of 22 inches equating to a screen size of  $56\text{ cm} \times 34\text{ cm}$ . The seam between the screens is aligned with the central axis of the chair. There is no gap between the monitors. The monitors are 107 cm away from the user. The upper end of the monitor lines up with the user's view angle at  $0^\circ$ . To see the lower border of the monitor the user has to look  $25^\circ$  down. To see the left and rightmost areas of the screen the user has to move his head at most  $32^\circ$ . The desk is a simple desk that is not height-adjustable. The desk is 76 cm high, 200 cm wide, and has a depth of

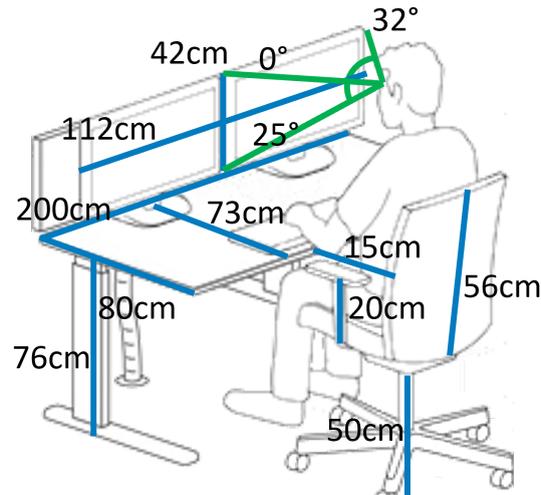


Figure 9.1: Sketch of the desk setup for data collection (image taken from [69], modified). Blue lines indicate distances, green lines indicate angles. Note that the sketch is not true to scale.

80 cm. The distance between the front of the desk to the desktops is 73 cm. The office chair is height adjustable and has an armrest and a backrest. The seat height is 50 cm high and the armrest is 20 cm above the seating platform. The armrest does not touch the desk when the chair turns. While seated the user is roughly 15 cm away from the desk. The chair also has rollers and can rotate freely around its axis.

### 9.1.2 Data Collection Process

This work uses two programs in the data collection process. The first program is an Arduino sketch for sensor readout and BLE communication on the sensors. The second program is a python script on the processing device that collects the data from the sensors. For wireless communication between the sensors and the processing device, this work uses Bluetooth low energy (BLE). This work interfaces the BLE module and the sensors on the Arduino Nicla board with libraries provided by Bosch Sensortec and Arduino. The provided cheat sheet [70] gives an impression of how to utilize the libraries. On the processing device, bleak is used as the library to utilize BLE. Figure 9.2 visualizes the overall data collection process.

With respect to the roles defined by BLE, the processing device acts the BLE central and the sensors as BLE peripheral devices respectively. Each of the sensors provides a BLE service with three characteristics. The movement and the environmental characteristics utilize the notification mechanism of BLE and can be subscribed to by the processing device to receive notifications whenever new sensor data is available. The third characteristic is the time synchronization characteristic which can be written to by the processing device for global time synchronization with the sensor (see Section 4.2).

The Arduino sketch runs continuously on the sensors. In its fixed frequency loop, the sketch reads out the sensor values. The sketch writes the recorded sensor data to the movement and environmental characteristics. The processing device periodically sends its timestamp to the sensors so they can update the estimate of their timedelta to the processing device. Starting the python script initiates the actual data collection. For each sensor, the script starts a data collector subscript that subscribes to both the movement and environmental characteristics of the sensor. When a processing device is subscribed, the BLE notification mechanism automatically notifies the device whenever new data is



classifier, and a test set to evaluate the models. Since the number of classes in the dataset is relatively large in relation to the number of samples, stratified sampling was used to split the datasets so that the distribution of classes in the different sets is approximately equal. This prevents underrepresenting rarer classes in either dataset. To do so, each time series was split into 70% that are included in the training set, 15% in the validation set, and 15% in the test set. The data is not shuffled to preserve the order of transitions to learn for the posture model.

This work collected an additional four recordings with 204 labeled transitions and a total duration of 11min 33s for the end-to-end testing of the state transition model. The data collection for the end-to-end test data took place on a different day than the model training and test dataset. This data is not included for the training and evaluation of the individual modules of the state transition model. The dataset contains around 10% of transitions where two transitions neighboring transitions are closer than 50 samples = 1s.

For the evaluation of the ergonomic evaluation over the long term, an additional three time series were recorded. In contrast to the other time series, these recordings are not labeled. The unlabeled recordings comprise a total duration of 04h 56min 42s.

## 9.2 Transition Extraction

The first step in the state transition model is the transition extraction. The performance of the overall model depends on whether all transitions between postures that the user performs can be detected. Therefore, this section serves to provide insight into the performance of the transition extraction algorithm and its extensions. In particular, this work compares the basic transition extraction algorithm, the extraction algorithm with additional filtering of non-action transitions, the algorithm allowing close transitions, and the algorithm combining allowing close transitions and filtering noise-transitions. Section 9.2.1 introduces the false extraction rate as a quantitative measure to compare the models. Section 9.2.2 compares the transition extraction variations by visual inspection and by comparing the false extraction rates.

### 9.2.1 False Extraction Rate

This work uses the false extraction rate (FER) to evaluate the transition extraction capabilities of the different variations of the transition extraction algorithm. The FER denotes the fraction between the falsely added or missed transitions and the overall true number of transitions. This gives an intuition for how well the different approaches extract the correct transitions and only the correct transitions. Concretely, the FER is defined as follows [71]:

$$FER = \frac{\#Missing\ transitions + \#Additional\ transitions}{\#True\ transitions} \quad (9.1)$$

$\#Missing\ transitions$  and  $\#Additional\ transitions$  denote the number of actual transitions that the transition extraction algorithm doesn't correctly extract and the number of additionally extracted transitions that are not actual transitions.  $\#True\ transitions$  denotes the total number of actual transitions in the dataset.

### 9.2.2 Transition Extraction Performance

The chosen example for visual inspection is in total 3min 18s long and contains 56 transitions, 7 of which are closer than 500ms to each other. Figure 9.4 shows the movement signal of the recording for the four different variants of the transition extraction pipeline.

Each time series is annotated with the correctly identified transitions, the falsely detected additional transitions, and the missed actual transitions.

As Figure 9.4a shows, the basic algorithm can detect most transitions. However, whenever transitions are close to each other, not all transitions can be correctly identified due to the distance constraint. The missed transitions (dashed lines) are close to other correctly identified transitions. One additional noise-transition is extracted at the beginning of the recording.

When the noise-classifier filters out transitions with a significant probability of not being actual transitions ( $p < 0.5$ ), the falsely identified transition at the start of the recording is removed (see Figure 9.4b). However, some correct transitions are also falsely filtered out. Applying the filtering of noise-transitions to unsupervised desk work the approach appears to be much more effective at filtering out many additional transitions (Figure 9.3). This suggests that for unsupervised desk work the noise-filtering yields an advantage. Note that for unsupervised desk work a higher threshold ( $p < 0.9$ ) provides better results.

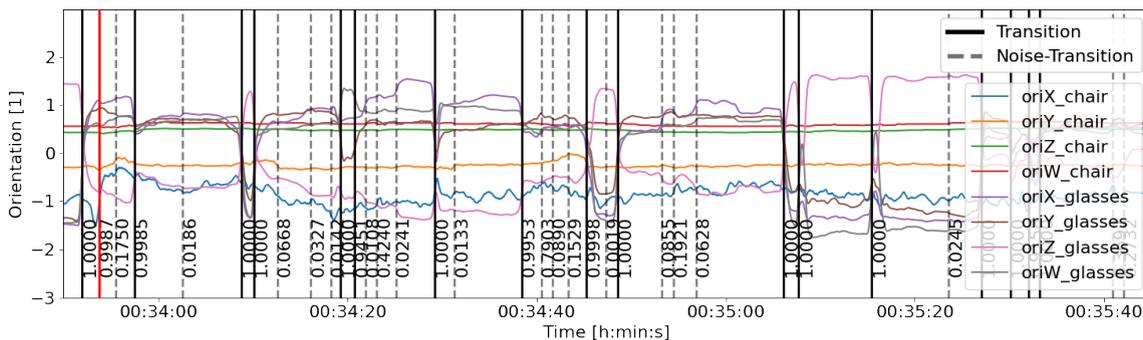


Figure 9.3: Noise-transition removal for example unsupervised desk work. The plot shows the virtual orientation sensor time series for both the glasses and the chair sensor. Extracted transitions are marked as solid lines. Each transition is annotated with the output of the noise-transition classifier. Transitions with an output  $< 0.9$  are masked (dashed lines). The approach masks all noise-transitions correctly, except one in the beginning (red line).

Removing the distance constraint allows detecting close transitions. If done, all previously missed transitions get extracted (see Figure 9.4c). On the other hand, more additional false transitions are detected as well.

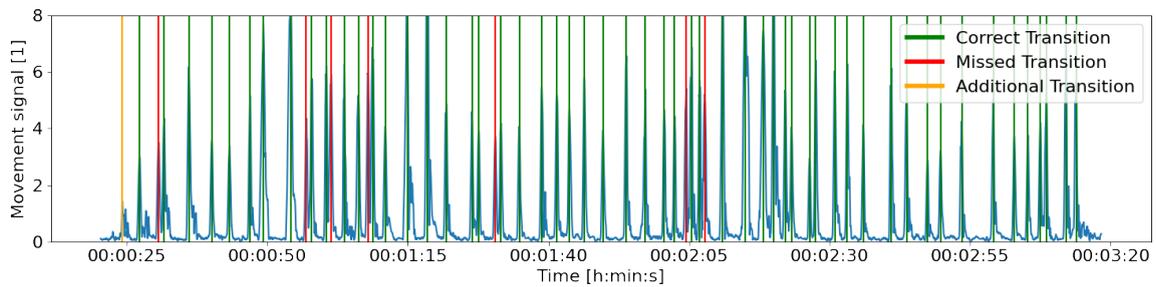
Allowing close transitions and simultaneously filtering noise-transitions at the same time allows for filtering of the close transitions. As seen in Figure 9.4d all close transitions that are missed by the basic approach are detected. Importantly also all falsely added transitions are removed by the noise-transition classification. This shows that the combination of allowing close transitions but also filtering them is well suited for practical transition extraction.

Table 9.1 shows the false extraction rates and the number of missing and additional transitions on the end-to-end test recordings for all algorithm variations. Following the findings of the visual inspection, the basic transition extraction algorithm already performs well ( $\text{FER} = 0.0539$ ). The noise-transition removal filters all additional transitions, but also some correct transitions. Allowing close transitions enables detecting all missed transitions, but also falsely detects additional transitions. In both cases, this results in slightly worse scores than the basic algorithm (avg.  $\Delta\text{FER} = -0.0123$ ). Combining both approaches yields the best result ( $\text{FER} = 0.0196$ ). This indicates that combining both approaches balances between allowing additional transitions but also filtering noise-transitions which in combination resolves the overshooting of the individual goals of the extensions.

Table 9.1: Transition extraction performance. FER = False extraction rate. The basic transition extraction algorithm already performs well but detects one additional transition and misses 10 transitions. Removing noise-transitions removes all additional transitions but also some correct ones. Allowing close transitions detects all transitions but falsely detects additional transitions. The combination of both methods performs best with missing no transitions and filtering all but four additional transitions.

<b>Model</b>	<b>Actual Transitions</b>	<b>Additional Transitions</b>	<b>Missing Transitions</b>	<b>FER</b>
Basic transition extraction	204	10	1	0.0539
Basic transition extraction + Noise-transition removal	204	14	0	0.0686
Basic transition extraction + Allowing close transitions	204	0	13	0.0637
Basic transition extraction + Noise-transition removal + Allowing close transitions	204	4	0	0.0196

This section concludes that the basic transition extraction algorithm is already well suited for transition extraction apart from missing close transitions. The combination of allowing close transitions and filtering additional noise-transitions performs even better than the basic algorithm by providing a balance between detecting possibly missed transitions and filtering additional ones. The noise-transition classification appears particularly effective for filtering out transitions during unsupervised desk work where the user moves less conscious than during the labeled recordings.



(a) Basic transition extraction

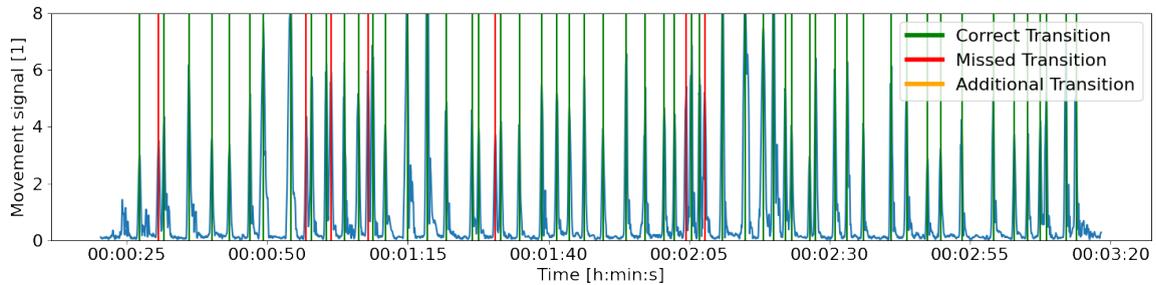
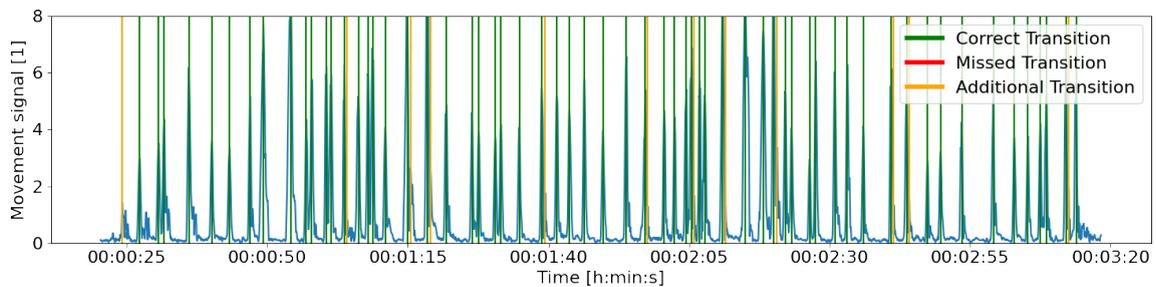
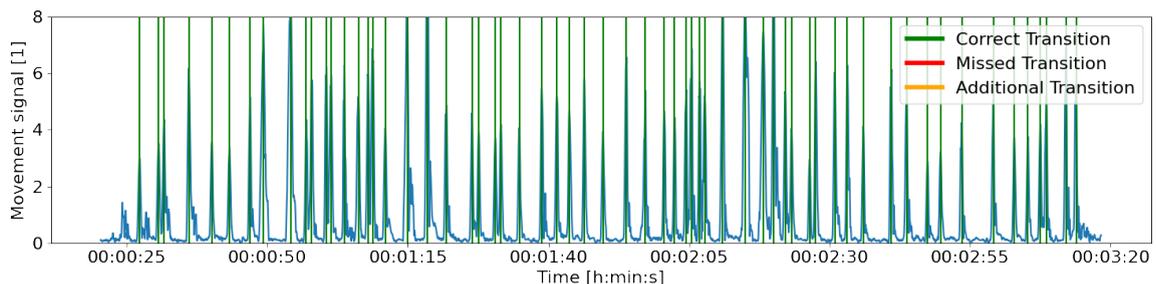
(b) Transition extraction with filtering of non-action noise-transitions ( $p < 0.5$ )(c) Transition extraction with allowing close transitions ( $d < 500$  ms)(d) Transition extraction with allowing close transitions ( $d < 500$  ms) and filtering of non-action noise-transitions ( $p > 0.5$ )

Figure 9.4: Transition extraction examples. The transition extraction algorithm finds most transitions but misses some transitions that are very close in time (dashed lines) due to the distance constraint. One additional transition is falsely detected (red line). The noise-transition filtering filters out the additional transition but also some correct transitions. When close transitions ( $d < 500$  ms) are allowed, additional transitions including the ones the normal algorithm falsely misses but also other, noise-action transitions are detected. Allowing close transitions and also filtering with the noise model provides a balance between both effects.

## 9.3 Transition Classification

After transition detection, the classification of the transitions is necessary to be able to derive the posture. Section 9.3.1 introduces metrics for the performance assessment of the transition classifier. Section 9.3.2 investigates how the transition classifier performs under various feature selections. Section 9.3.3 evaluates how data augmentation and the introduction of data generated using the manifold approach impact the transition classifier. Finally, Section 9.3.4 takes a detailed look at typical classification errors of the model.

### 9.3.1 Classification Metrics

This work uses two commonly used classification metrics, the accuracy, and the F1-score, for evaluating the transition classifier. These metrics can be derived from the confusion matrix. Table 9.2 explains the confusion matrix for a binary classification task. The concept can be extended to a multiclass classification task by adding columns and rows for all classes and viewing all but the true class as negatives.

Table 9.2: Confusion matrix for a binary classification task. TP = True Positives, FN = False Negatives, FP = False Positives, and TN = True Negatives.

		Predicted	
		Positive	Negative
Actual	Positive	$TP$	$FN$
	Negative	$FP$	$TN$

The accuracy is a basic score that describes the proportion of data that is correctly labeled which gives an intuitive understanding of a classifier's performance.

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (9.2)$$

While accuracy is an intuitive metric it does not take the distribution of the data into account which might lead to incorrect conclusions. Therefore, this work also utilizes the F1-score which in turn combines the measures of precision and recall.

Precision denotes how much of the positive labeled data is positive.

$$precision = \frac{TP}{TP + FP} \quad (9.3)$$

Recall answers the question of how much of the positive labeled data is predicted correctly.

$$recall = \frac{TP}{TP + FN} \quad (9.4)$$

The F1 score finally combines both recall to balance between the two concurrent metrics.

$$F_1 = \frac{2 * precision * recall}{precision + recall} \quad (9.5)$$

### 9.3.2 Feature Selection

Both the sensor on the glasses, as well as the sensor on the chair collect a total of 10 different motion features: Three features for each of the accelerometer axes, three for the gyroscope axes and four for the orientation quaternion of the virtual orientation sensor. The following section compares the performance of the transition classifier depending on the selection of features for classification. It thereby aims to find the optimal feature selection for deriving the transitions from the time series. In addition, this work compares the model size with the goal to be able to use the model on mobile devices or directly on the sensor. Note that the transition classifier was trained on a GPU. Therefore, despite seeding all used libraries results may not be fully reproducible [72]. Table 9.3 shows the model size, loss, accuracy, and F1-Score of the transition classifier with varying feature selection.

The results reveal that no matter the feature selection the transition classifier performs well on the training data (Accuracy, F1-Score  $> 0.9$ ). However, if not all ten features of the sensor on the glasses are selected, the classifier does not generalize well on the test set. In that case, the test set Accuracy is at least 0.26, and the F1-Score 0.19 worse than the performance on the training set. Selecting only the gyroscope features for both sensors results in the worst performance (Accuracy = 0.5687, F1-Score = 0.6109). The best-performing feature selection that selects the same features for both the glasses and the chair sensor is using the full 20 features (Accuracy = 0.9866, F1-Score = 0.9932 for training and Accuracy = 0.8879, F1-Score = 0.9094 for test set respectively). In particular, this feature selection is able to shrink the generalization gap between training and test set performance significantly compared to the other symmetric feature selections.

Interestingly, similar and even better performance can be achieved with an asymmetric feature selection where for the glasses sensor the full ten features are selected and for the chair sensor the selected features are filtered. In particular, the transition classifier performs best when for the glasses sensor only the gyroscope features are selected. This feature selection achieves the best performance for both the training set (Accuracy = 0.9952, F1-Score = 0.9941) and for the test set (Accuracy = 0.9070, F1-Score = 0.9210) of the tested feature selections. Additionally, this model has 15,680 parameters less ( 32%) than the model with all 20 selected features. This means that the memory footprint of the model is also smaller and computation cheaper. Therefore, this feature selection does not only perform better but also is more suitable for low-power devices with smaller memory availability.

The main findings of the comparison of the transition classifier under different feature selections are: If not all 10 features are selected for the glasses sensor, the generalization of the classifier on the test data is significantly worse. The classifier with the full 10 features of the glasses sensor but only the gyroscope features from the chair sensor performs best and requires less model size. In the following, only the best performing feature selection (accelerometer, gyroscope, and orientation sensor of the glasses and gyroscope of the chair sensor) is considered if not mentioned otherwise.

Table 9.3: Transition classifier performance per selected features. Acc = Accelerometer, Gyro = Gyroscope, Ori = Virtual orientation sensor. L = Categorical cross-entropy loss, A = Accuracy, F1 = F1-Score. If not mentioned otherwise the same features are selected for both sensors. Asymmetric feature selection was also performed for the features of the glasses sensor. The results were however omitted as the models performed significantly worse.

Features	#Features	Params	Training	Test
Acc	6	17,690	L: 0.2960 A: 0.9494 F1: 0.9489	L: 1.2776 A: 0.7336 F1: 0.7308
Gyro	6	17,690	L: 0.4125 A: 0.9179 F1: 0.9344	L: 1.8585 A: 0.5687 F1: 0.6109
Ori	8	17,946	L: 0.2886 A: 0.9513 F1: 0.9579	L: 1.4537 A: 0.7548 F1: 0.7620
Acc, Gyro	12	33,242	L: 0.1821 A: 0.9885 F1: 0.9853	L: 1.1882 A: 0.7674 F1: 0.7630
Acc, Ori	14	33,498	L: 0.1840 A: 0.9780 F1: 0.9729	L: 1.8064 A: 0.7844 F1: 0.7908
Gyro, Ori	14	33,498	L: 0.1722 A: 0.9914 F1: 0.9932	L: 2.0017 A: 0.7315 F1: 0.7620
Acc, Gyro, Ori	20	49,050	L: 0.1622 A: 0.9866 F1: 0.9884	L: 0.7276 A: 0.8879 F1: 0.9094
Acc, Gyro, Ori Glasses Acc Chair	13	33,370	L: 0.1637 A: 0.9828 F1: 0.9851	L: 0.8178 A: 0.8795 F1: 0.8935
Acc, Gyro, Ori Glasses Gyro Chair	13	33,370	L: 0.1257 A: 0.9952 F1: 0.9941	L: 0.5809 A: 0.9070 F1: 0.9210
Acc, Gyro, Ori Glasses Ori Chair	14	33,498	L: 0.1653 A: 0.9914 F1: 0.9894	L: 0.8124 A: 0.8753 F1: 0.8842
Acc, Gyro, Ori Glasses Acc, Gyro Chair	16	41,146	L: 0.1475 A: 0.9876 F1: 0.9874	L: 0.9165 A: 0.8689 F1: 0.8816
Acc, Gyro, Ori Glasses Acc, Ori Chair	17	41,274	L: 0.1445 A: 0.9885 F1: 0.9866	L: 1.0912 A: 0.8753 F1: 0.8732
Acc, Gyro, Ori Glasses Ori, Gyro Chair	17	41,274	L: 0.1430 A: 0.9914 F1: 0.9891	L: 0.7799 A: 0.8943 F1: 0.9060

### 9.3.3 Data Augmentation & Manifold

This work proposes data augmentation and data generation with manifolds as methods to create synthetic data. Therefore, this section investigates how the addition of this data impacts the performance of the transition classifier. For evaluation of data augmentation, each transition window in the training set got modified with the four different data augmentation techniques mentioned in Section 5.3: Jittering, scaling, time warping, and magnitude warping. For the normal distributions of the four augmentation techniques variances  $\sigma \in \{0.05, 0.01\}$  were used. For time and magnitude warping  $k \in 2, 4$  warping points were used. In total, the data augmentation adds 12 variations for each transition window to the dataset. The manifold-based data generation approach (Section 8.4) was used to create in total 72 transition windows for 8 previously not regarded transitions (combinations of head angle and head tilt). Table 9.4 shows the model performance of the transition classifier for different datasets.

Table 9.4: Transition classifier performance for different data. L = Categorical cross-entropy loss, A = Accuracy, F1 = F1-Score.

Data	Dataset size	Params	Training	Test
Basic Data	1047	33,370	L: 0.1257 A: 0.9952 F1: 0.9941	L: 0.5809 A: 0.9070 F1: 0.9210
Basic Data + Manifold	1119	39,010	L: 0.1507 A: 0.9902 F1: 0.9924	L: 0.5682 A: 0.9089 F1: 0.9194
Augmented Data	13,611	33,370	L: 0.1186 A: 0.9885 F1: 0.9881	L: 0.7442 A: 0.9239 F1: 0.9255
Augmented Data + Manifold	14,547	39,010	L: 0.1188 A: 0.9904 F1: 0.9912	L: 0.7837 A: 0.9248 F1: 0.9324

Comparing the classifier performance using the basic dataset to the one with the generated data using the manifold, it is observable that the classifier needs an additional 5640 parameters to account for the additional classes. Also, the training for the models using the manifold data needed more epochs to converge. The model performance of the model with the added generated data is similar to the performance of the model without the added data (avg.  $\Delta$ Accuracy = +0.0024,  $\Delta$ F1-Score = +0.0033). This means that the transition classifier is able to discriminate between the generated data similarly to the actual recorded data.

Comparing the models using augmentation of the dataset with the ones without augmentation it is observable that the models perform on average worse on the training set and better on the test set. The decrease of performance on the training set (avg.  $\Delta$ Accuracy = -0.0034,  $\Delta$ F1-Score = -0.0024) is smaller than the increase of performance on the test set (avg.  $\Delta$ Accuracy = +0.0748,  $\Delta$ F1-Score = +0.0043). Therefore, the augmentation of the dataset reduces the generalization gap between the training and test set slightly. Interestingly the performance increase of the accuracy metric is much bigger than the increase in F1-Score.

This section concludes the following main findings: Adding data generated using the manifold vector combination approach does not impact the ability of the transition classifier to discriminate between the transitions significantly. Augmentation of the dataset allows

slightly closing the generalization gap between training and test set. The overall best performing model is the transition classifier with the augmented data and the data generated using the manifold.

### 9.3.4 Transition Classifier Errors

This section takes a closer look at the performance of the transition classifier using the best-performing feature selection. Therefore, this work compares the true and derived transition for one concrete example recording. This section also takes a closer look at the confusion matrices of the transition classifier for the training and test dataset to identify common errors of the model.

Figure 9.5 shows a comparison of the detected transitions of the transition extraction algorithm and the transition classifier and the actually labeled transitions for an example recording. For this example, the transition classifier using the augmented dataset was used. Note that the chosen recording is part of the dataset not used for either training or testing the transition classifier. While one of the transitions is not correctly extracted by the extraction algorithm, most of the correctly extracted transitions are also correctly classified. The only wrongly classified transition in the example is the second to last transition. There, *right\_down* is misclassified as *right\_center*. Therefore, the transition classifier misclassified the end posture of the transition in one degree of freedom.

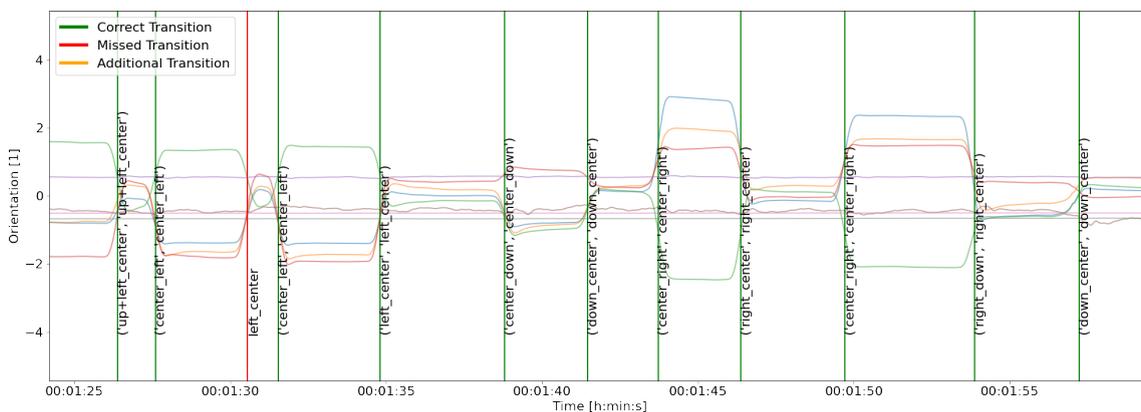


Figure 9.5: Comparison of true and derived transitions. Each transition (lines) is annotated with a tuple (true transition, derived transition). Some transitions were not correctly identified by the transition extraction algorithm. Actual transitions that the extraction algorithm missed (dashed lines) are annotated with the true transition. Falsely identified additional transitions are annotated with the derived transition.

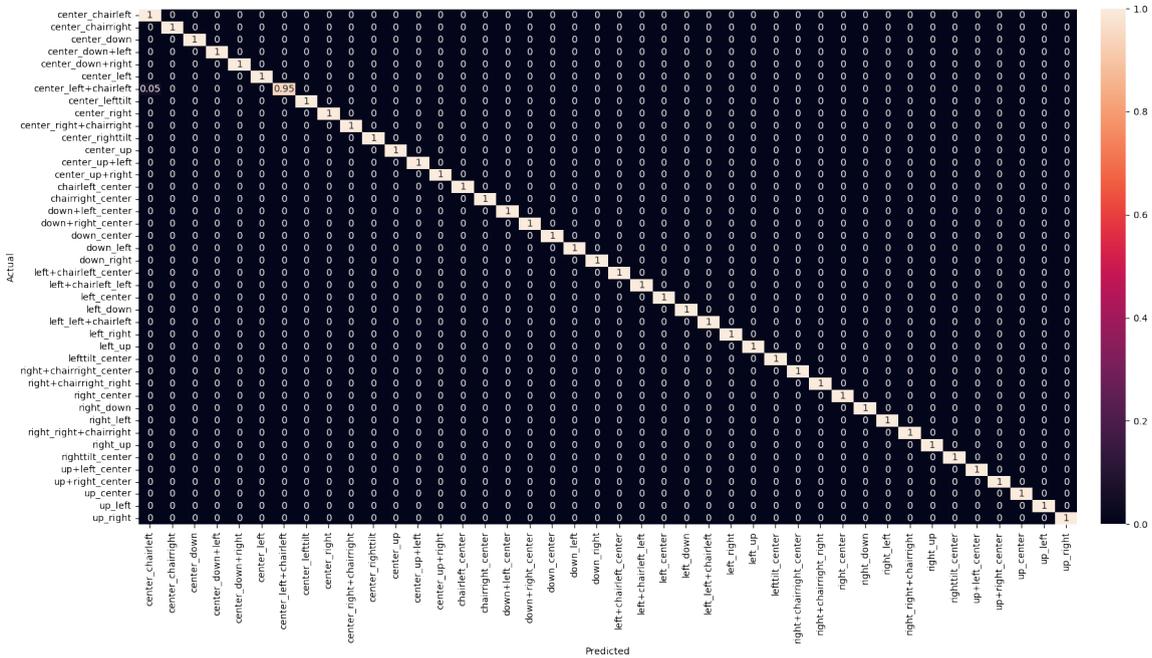
To identify common errors of the transition model this work takes a closer look at the confusion matrices of the transition classifier (Figure 9.6) for the basic dataset and the data set after data augmentation (Figure 9.7). The cell in row  $i$  and column  $j$  of the confusion matrix indicates the number of observations that are known to be transition  $i$  and predicted to be transition  $j$ . The diagonal of the confusion matrix denotes the correctly classified transitions.

The confusion matrices for the training sets (Figures 9.6a and Figure 9.7a) indicate a good performance as almost all entries are on the diagonal. This is consistent with the good accuracy and the good F1-score as noted in Section 9.3.3. The transition classifier for the augmented dataset misclassified more, but still only a few, transitions in the training set. One observable error is the confusion of two transitions in the same direction, e.g. *up\_center* and *center\_down*. This is unsurprising since both transitions move the head from

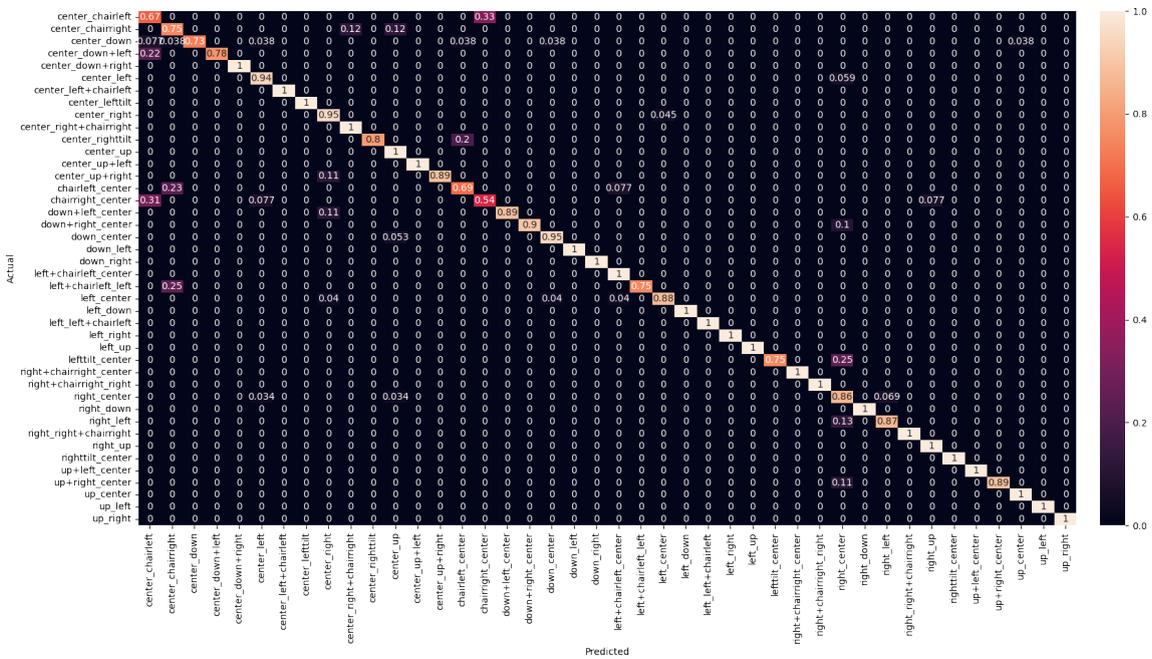
downwards. Another observable error is the confusion of *down+left\_center* and *down\_center*. There the start-posture is misclassified in one degree of freedom. This error is also explainable by the similarity of the two transitions which differ only in one degree of freedom of the postures. Similar errors also occur for other transitions.

For the test sets (Figure 9.6b and Figure 9.7b) similar errors are observable but overall with higher frequency than the training set. Again, errors where either two transitions with the same general direction but different start- and end-postures got confused or one degree of freedom of either start- or end-posture is not get recognized correctly can be observed. Additionally, occasional confusions of transitions of the trunk and the head in the same direction, e.g. *center\_chairleft* and *center\_left* can occur. Errors of the classifier using the basic and the augmented dataset are similar. The confusion matrix for the augmented dataset also reflects the slightly higher accuracy compared to the classifier only using the basic dataset.

The important findings of this section are: The observed errors are mostly confusions between two transitions with the same direction but different start- and end-postures and misclassifications of one degree of freedom in the start- or end-postures of a transition. Therefore, typical errors of the transition classifier are due to the confusion of two similar transitions. The observed errors are consistent for both the basic and the augmented dataset.

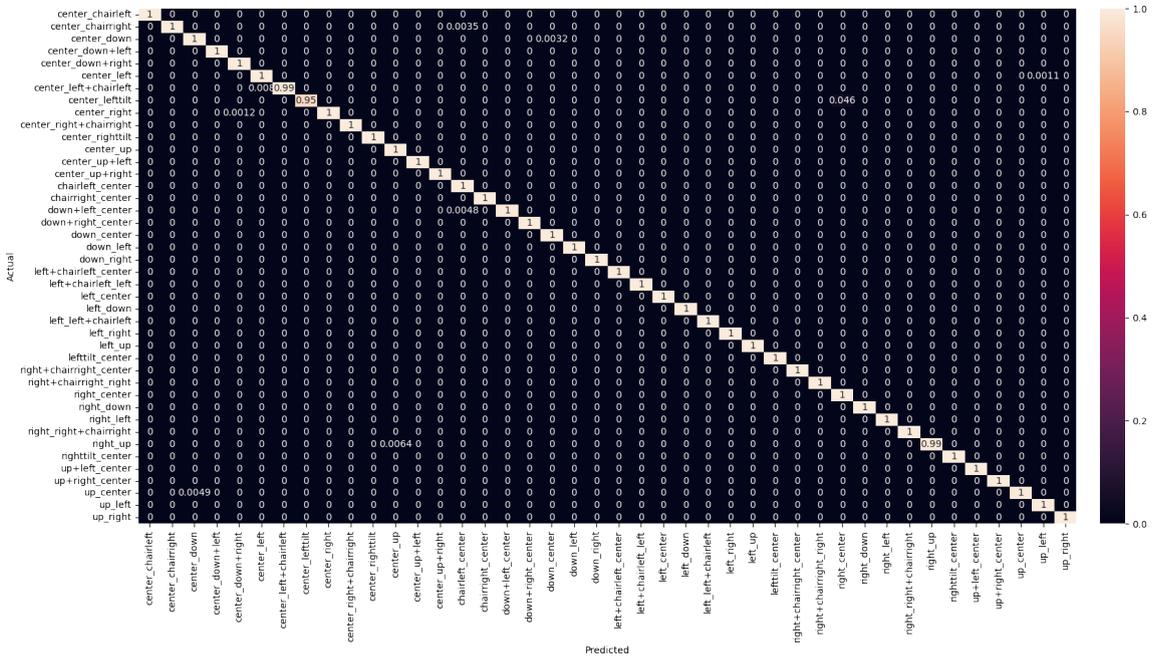


(a) Confusion matrix for training data

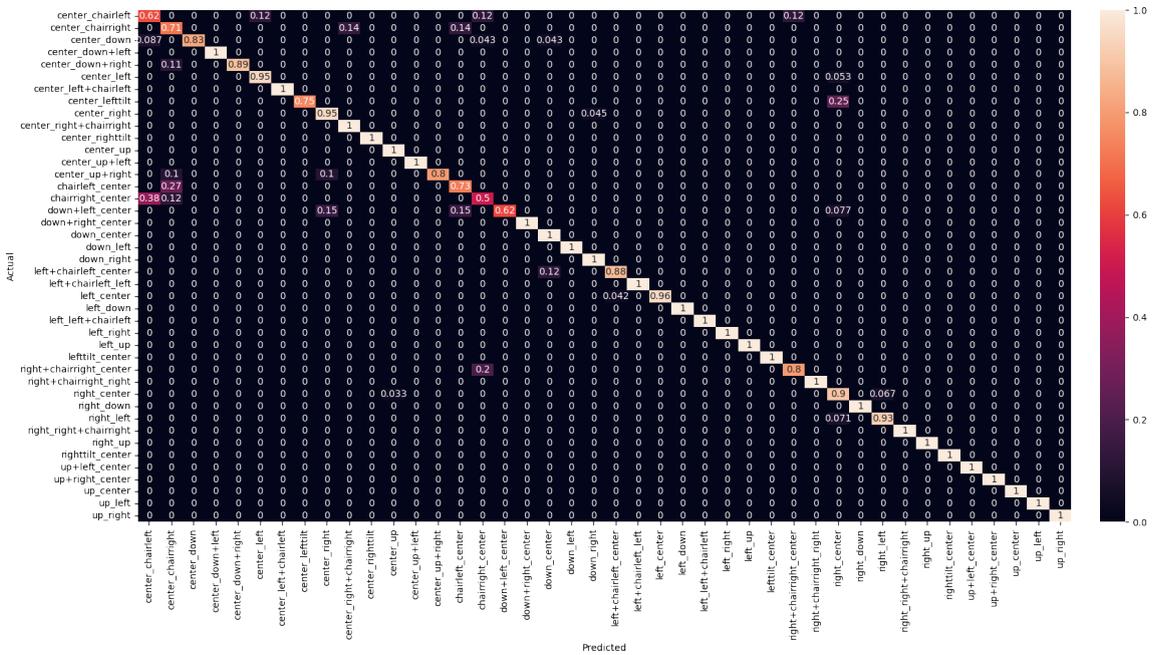


(b) Confusion matrix for test data

Figure 9.6: Confusion matrices of the transition classifier. The rows of the confusion matrix are normalized.



(a) Confusion matrix for training data



(b) Confusion matrix for test data

Figure 9.7: Confusion matrices of the transition classifier after data augmentation. The rows of the confusion matrix are normalized.

## 9.4 Posture Extraction

The last step of the state transition model is the posture extraction. Section 9.4.1 introduces the BLEU-score as the performance metric for the posture extraction. Section 9.4.2 examines the performance of the posture model under different dataset variations in detail.

### 9.4.1 BLEU-Score

In theory, it would be possible to evaluate transition sequences to posture sequences using the classification metrics introduced in Section 9.3.1. Instead of evaluating the classification of single transitions, these metrics would then evaluate the correct translation of a whole posture sequences. Consequentially, the metrics would classify a posture sequence as incorrect no matter if the complete sequence or only the derivation of one posture is wrong. However, this contradicts the intuition to assess an almost correctly derived posture sequence where only one of  $n$  is misclassified as still very good and a completely random sequence as bad. Therefore, a metric that accounts for this intuition is desirable. Consequentially, this work uses a metric from the area of natural language translation that is designed to correlate well with the human judgment of translation, the BLEU-score [73].

The basic idea of the BLEU score is to compare a query text with one or more reference translations and assign a score between 0 and 1 indicating how similar the query is to the reference texts. Values closer to 1 represent more similar texts. BLEU computes its precision metric by comparing the n-grams of the query with the n-grams of the reference translation and count the number of matches. Matches are independent of the positions where they occur. An additional factor, the brevity penalty, is introduced to punish short translations. Equation 9.6 shows the calculation rule for the BLEU-score [73]. Depending on the maximum order of n-grams the BLEU-score is also called BLEU-n, i.e. BLEU with 1-grams, 2-grams, 3-grams, and 4-grams is called BLEU-4 [74].

$$\text{BLEU-n} = \underbrace{\min\left(1, \exp\left(1 - \frac{\text{reference-length}}{\text{output-length}}\right)\right)}_{\text{brevity penalty}} \underbrace{\left(\prod_{i=1}^n \text{precision}_i\right)^{1/n}}_{\text{n-gram precision}} \quad (9.6)$$

With  $m_i$  being the number of matched  $i$ -grams between translation  $T$  and query  $Q$  and  $l_i$  being the total number of  $i$ -grams in  $T$  the  $i$ -gram precision can be calculated as

$$\text{precision}_i = \frac{m_i}{l_i} \quad (9.7)$$

There are two additional typical modifications to the BLEU score: The scores for each of the individual n-gram orders are aggregated using the geometric mean. If there is no n-gram overlap for any order the geometric mean and consequentially the BLEU-score is 0. To avoid this harsh behavior the value 0 is replaced with a small value  $\epsilon$ . And also, the precision is modified not to allow more matches of a query n-gram than there are occurrences of that n-gram in the reference. This avoids translations that just repeat probable words [73].

The variation of BLEU used in this work uses the BLEU-4 score with  $\epsilon = 0.1$ . Instead of word n-grams, the BLEU score compares n-grams of posture sequences. The BLEU score is calculated individually for each of the four degrees of freedom.

### 9.4.2 Posture Model Performance

The following section evaluates the performance of the posture model. In particular, this work compares the performance of the posture model on the training data, the test

data, and the transitions classified by the transition classifier. Additionally, this section investigates how the performance changes depending on whether the data generated using the manifold approach is used and the application bias is applied and whether the dataset is augmented or not. Table 9.5 shows the BLEU-scores for each of the variants. The score is also broken down into the scores for each of the degrees of freedom.

Table 9.5: Posture model performance for different datasets. The scores denote the BLEU-score for each of the regarded degrees of freedom and  $\mu$  is the average over all degrees of freedom. "Combined" denotes the performance of the posture model when fed with the classified transitions from the transition classifier.

Data	Training	Test	Combined
Basic Data	[1.0, 0.9816, 1.0, 1.0] $\mu = 0.9954$	[1.0, 0.9770, 0.9977, 0.9977] $\mu = 0.9931$	[0.9549, 0.9060, 0.9844, 0.9229] $\mu = 0.9421$
Basic Data + Bias & Manifold	[1.0, 0.9415, 0.9314, 0.9598] $\mu = 0.9582$	[1.0, 0.9392, 0.9333, 0.9631] $\mu = 0.9589$	[0.9600, 0.8647, 0.9333, 0.8962] $\mu = 0.9135$
Augmented Data	[1.0, 1.0, 1.0, 1.0] $\mu = 1.0$	[1.0, 1.0, 0.9977, 0.9977] $\mu = 0.9989$	[0.9595, 0.9501, 0.9977, 0.9143] $\mu = 0.9554$
Augmented Data + Bias & Manifold	[0.9986, 1.0, 0.9307, 1.0] $\mu = 0.9823$	[0.9986, 1.0, 0.9307, 1.0] $\mu = 0.9823$	[0.9670, 0.9526, 0.9428, 0.9376] $\mu = 0.9500$

Looking at the basic posture models without applying the application bias, the hidden Markov model (HMM) can almost perfectly learn the translation from transitions into states for the training set. This is also true for the model using data augmentation (avg. BLEU-Score 0.9977). Thus, the HMM is able to model the underlying process. The performance for the test set is almost identical to the performance on the training set for all regarded models (avg.  $\Delta$ BLEU-score = -0.0007). This is not surprising as the previous result showed that the HMM is able to model valid sequences and the test set is made up of valid sequences as well.

The performance of the posture model on the transitions classified by the transition model does slightly drop compared to the training and test sets (avg.  $\Delta$ BLEU-score = -0.0430). A drop in performance is expected as the posture model has to handle the misclassifications of the imperfect transition classifier. However, the fact that the performance drop is relatively small shows that the model is able to correct mistakes made by the transition classifier.

When comparing the biased HMMs with the HMMs without application bias it is observable that the biased models perform worse for all datasets (avg.  $\Delta$ BLEU-score = -0.0233). The difference between the biased and unbiased models is larger for the basic dataset (avg.  $\Delta$ BLEU-score = -0.0333) than for the augmented dataset (avg.  $\Delta$ BLEU-score = -0.0132). In this thesis it was observed, that the difference is larger for the training and test set than for the combined model which uses the classified transitions. The fact that the biased model performs worse on the training set is expected as the bias forces the HMM slightly away from the distributions estimated on the training data. However, the biased model also performs worse on the classified transitions. In particular, for the head tilt the application bias appears to reduce the overall score the most. Interestingly, all of the manifold-generated data includes the head tilt. In isolation, the introduction of bias worsened model performance. However, since the result is obtained through the interaction of the transition and posture model, several factors influence the result. It remains to be seen whether other approaches to bias the HMMs will yield performance improvements.

Figure 9.8 shows a comparison of the derived postures using the posture model and the labelled postures for an example recording. For this example, the posture model with the

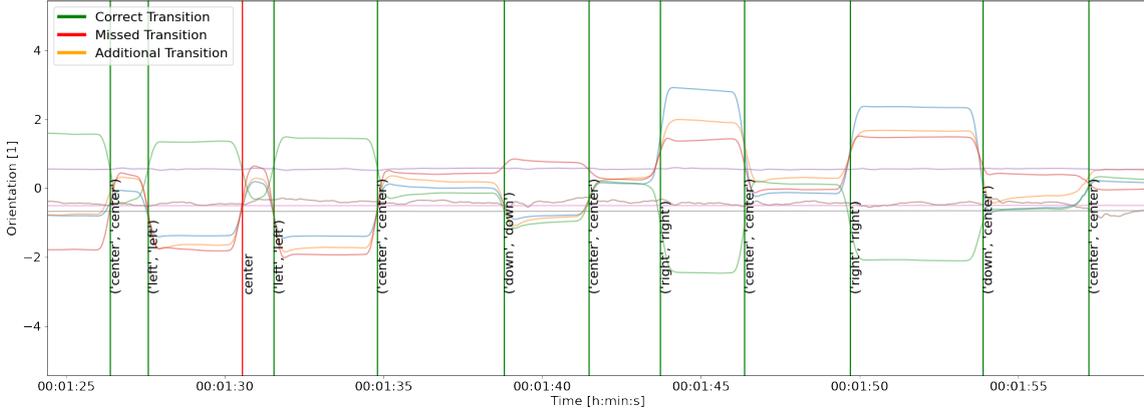


Figure 9.8: Comparison of true and derived postures. Each transition (lines) is annotated with a tuple (true posture, derived posture). Two transitions were not correctly identified by the transition extraction algorithm. These transitions (dotted lines) are annotated with the true posture.

augmented training data was used. Note that the chosen recording is part of the dataset not used for either training or testing the transition classifier.

This section concludes that the posture model is able to capture the underlying process well and is even able to correct errors of the transition classifier. The posture model for the augmented dataset performed. Using an application bias does not immediately yield better performance which leaves the question open if the bias helps generalization of the posture model to unknown data.

## 9.5 End-to-End Pipeline comparison

This section evaluates the end-to-end performance of the different variants of the complete pipeline. Section 9.5.1 introduces the quantitative evaluation metric for the end-to-end performance. Section 9.5.2 compares the different pipeline variations quantitatively and discusses the effect of the different pipeline variations on the overall posture extraction performance.

### 9.5.1 Metric

The end-to-end performance metric grasps how well a model is able to extract the correct postures over a recording. The metric evaluates how much of the time the extracted posture is similar to the true posture and scale this by how similar the two postures are at each moment in time. This penalizes if the detected posture differs from the true posture at any time and takes the duration of each posture into account. Thereby, it also penalizes incorrect transition extraction.

The metric compares the posture sequence extracted using the pipeline (query) and the labeled postures (reference). For multiple recordings, the metric is calculated by averaging over all individual scores, weighted by the length of the recording. The following calculation sequence calculates the metric for one recording:

1. Let  $T_q = (t_{q,i})_{i=1}^n$  and  $T_r = (t_{r,i})_{i=1}^m$  be the timestamps of the transitions and  $P_q(t)$  and  $P_r(t)$  the postures at time  $t$  of query  $q$  and reference  $r$  respectively. Additionally, let  $t_0$  be the start time of the recording and  $t_{max}$  be the end time of the recording.
2. The query and reference posture must be compared whenever either one changes. Therefore, calculate moments in time where either the query or the reference posture

sequences changes. Therefore, sort the sequence of the combined timestamps of the query and the reference posture sequence. Include the start of the recording as the first timestamp  $t_0$ .

3. Compare query and reference posture for each determined timestamp. Therefore, calculate the subset accuracy of query and reference at each point of time. The subset accuracy extends the accuracy by allowing partially correct labeling. By convention both sequences are expected to be in the center at the start. Square the resulting score to punish large deviations more. The reason for this alteration of the subset accuracy is that in practice most of the time only one or two degrees of freedom are not neutral. Therefore, deviations are more punished to lower the score of classifications which just randomly guess "mostly center".  $I$  is the indicator function.

$$A_i = \left(\frac{1}{4} \sum_{d=1}^4 I(P_q(t_i) = P_r(t_i))\right)^2 \quad i = 0, \dots, m+n \quad (9.8)$$

4. Each posture is held after a transition until another transition changes the posture again. The score for each posture is scaled by how long the posture lasts to account for the duration of each posture. Therefore, the subset accuracy is multiplied by the difference between the timestamp of a transition and the next timestamp. The last extracted posture is held until the end of the recording  $t_{max} = t_{m+n+1}$ . The score is finally normalized by dividing by the total length of the recording.

$$S = \frac{\sum_{i=0}^{m+n} A_i * (t_{i+1} - t_i)}{t_{m+n+1} - t_0} \quad (9.9)$$

### 9.5.2 Comparison of Pipelines

This section aims to evaluate the overall end-to-end performance of different variations of the state transition model processing pipeline for posture extraction. In total 16 different processing pipelines are compared. The various pipelines utilize different transition extraction mechanisms and some of the pipelines utilize the application bias and manifold data generations while others don't. As a baseline, this work also evaluates the performance of the model that simply always predicts the user's posture to be in the center. Table 9.6 shows the end-to-end performance metric for the different models.

All different state transition model pipeline variations (avg. Metric = 0.8929) manage to score significantly higher than the baseline (Metric = 0.7159). The worst-performing pipeline utilizes the basic dataset, filters noise-transitions, and applies the application bias. This pipeline achieves a score of 0.8592. In contrast, the best-performing pipeline achieves a score of 0.9123. This pipeline utilizes the dataset with augmented data, allows close transitions that are masked, and filters noise-transitions in addition to the basic state transition model. This demonstrates that the wearable approach with the state transition model for posture extraction is able to match the ground truth posture well. Also, the different proposed additions to the basic state transition model pipeline overall increase the score with varying success.

Every pipeline with the augmented dataset performs better than the same pipeline trained on the basic dataset (avg.  $\Delta$ Metric = 0.0170). Although the difference is small, the consistent increase of performance across all models suggests that data augmentation is able to help the pipeline to generalize better on the unseen test data. This confirms the findings of Section 9.3.3 that concluded that data augmentation improves the transition classifier performance on the test dataset. These observations overall conclude that the augmentation of the dataset is useful for regularization of the state transition model although the increase in performance is limited.

The removal of noise-transitions (see Section 6.4.2) lowers the pipeline metric in all but one case. The examination of the extracted transitions on the test recordings shows that the basic transition extraction mechanism misses very few and detects very few additional transitions. This means that overall the basic transition extraction algorithm already performs very well for the test recordings. The noise-transition classifier is then able to correctly filter out the additional transitions. However, it also filters out some of the correct transitions which explains the worse score. Simply increasing the threshold for noise-classification has not proven useful as the falsely misclassified transitions receive similar scores than the actual true transitions. As mentioned in Section 9.2, during visual inspection of the effect of the noise-transition removal on the unsupervised recordings it was found that the removal of noise-transitions appears effective. This raises the suspicion that when the user moves more consciously during the predefined sequences of the test data recording the basic extraction algorithm is sufficient. During unsupervised desk work, there are more unconscious movements that the transition classifier correctly masks. This work concludes that the idea of filtering noise-transitions is promising but its actual performance impact depends on the application.

Allowing close transitions and masking transitions that enter a neighboring transition (see Section 6.4.3) consistently improves model performance (avg.  $\Delta$ Metric = 0.0157). Similar to data augmentation the difference is not very big but the consistent increase of the metric shows that the approach does indeed yield a performance increase. This work concludes that allowing close transitions helps the overall model performance especially paired with the masking of transitions that appear in neighboring transition windows.

The addition of the application bias and the manifold generated data overall decreases performance slightly (avg.  $\Delta$ Metric = 0.0087). As this decrease is very small it could be the result of statistical effects. A possible explanation for lower scores could also be the fact that the addition of new transitions to the transition classifier may have a negative impact on the performance of the classifier for the unknown end-to-end test dataset. Also, the positive effect of introducing new transitions may be less than expected as the most often occurring transitions are already covered by the recorded dataset. This work concludes that the generation of data using the manifold approach in itself is viable and does not significantly negatively impact model performance. However, also no improvement in performance could be seen.

The results of the end-to-end pipeline evaluation indicate an overall good match of the extracted posture with the ground truth. Data augmentation allows for better generalization of the pipeline to unknown data. Filtering noise-transitions did not yield a performance improvement but the application to unsupervised movement data appears promising. Allowing close transitions and masking overlapping transitions does consistently improve performance. Adding data generated by the manifold method does not improve or decrease performance significantly.

Table 9.6: End to end model performance comparison. Metric denotes the score from Section 9.5.1. The "Baseline" model classifies the posture to be permanently neutral.

Model	Data Augmentation	Noise-transition removal	Transition separation	Bias & Manifold	Metric
Baseline					0.7159
State transition model					0.8949
		X			0.8681
			X		0.9050
		X	X		0.8792
				X	0.8878
		X		X	0.8592
			X	X	0.8961
		X	X	X	0.8704
	X				0.9114
	X	X			0.8953
	X		X		0.9119
	X	X	X		0.9123
	X			X	0.9008
	X	X		X	0.8835
	X		X	X	0.9090
	X	X	X	X	0.9015

## 9.6 Ergonomic Movement Manifold

This section investigates using a manifold for explaining and reasoning about ergonomic movement data. This work uses a principal component analysis of the data as a manifold. The dataset used for the PCA comprises the combined training and test set. 9 principal components are necessary (see Figure 9.9) to explain at least 70% of the variance of this dataset. With data augmentation, instead 13 components are necessary. In the following, this work uses a projection on the first two principal components to explain substructures on the manifold. The first two components of the PCA explain the most variance ( $\approx 50\%$ ). While they explain not all of the used variance, it is easier to understand the projection to this two-dimensional subspace of the PCA. In addition to the projection of each transition window, this work uses the centroids for each transition class as representatives for that class.

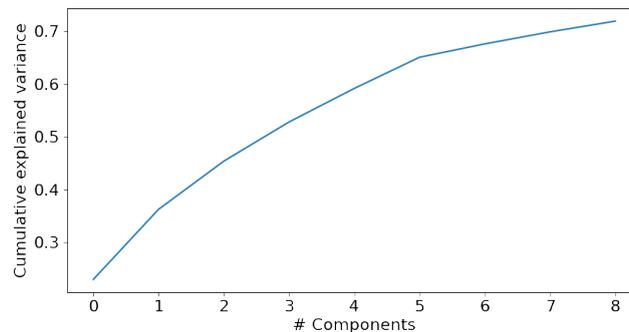


Figure 9.9: Cumulative explained variance of the first principal components. At least 9 components are necessary to explain 70% of the dataset's variance.

The projection of the dataset onto the manifold spanned by the first two principal components reveals three interesting substructures:

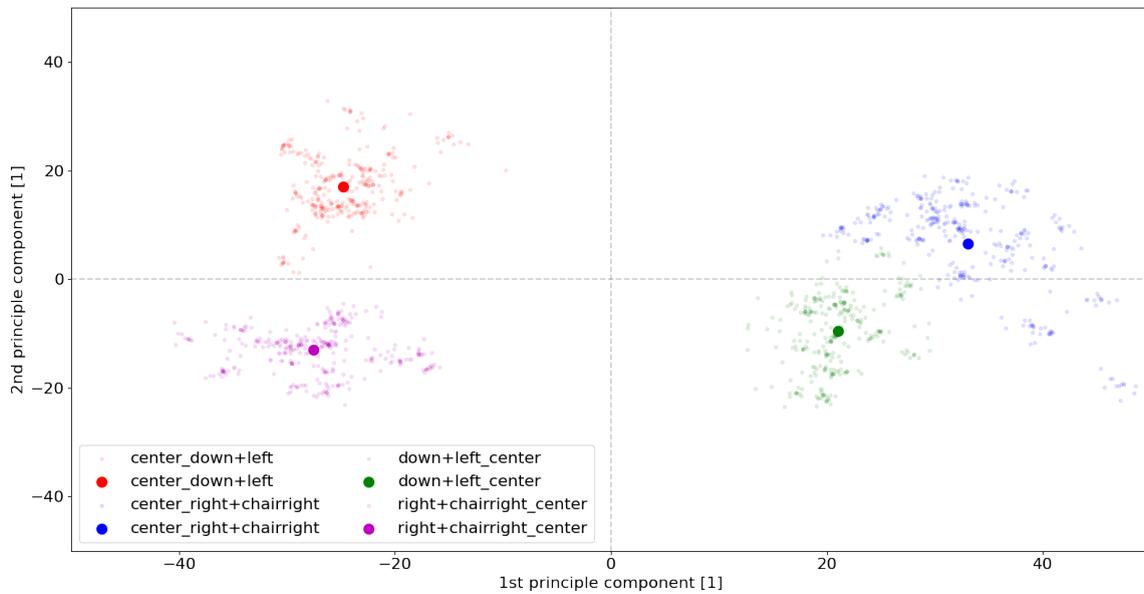
First, inverse transitions, which are transitions from a posture  $A$  to posture  $B$  and the transition back from  $B$  to  $A$  are projected approximately opposite on the manifold. Figure 9.10a illustrates this for the transitions *center\_down+left* and *center\_right+chairright*. Similar observations can be made for other transitions as well. Note that the symmetry of each transition is not perfect with respect to the origin. This observation shows that the idea of inverse transitions translates onto the manifold.

The second observation is that transitions with the same direction are projected to the same area on the manifold, independent of their start and end posture. As Figure 9.10b illustrates, the transitions *center\_up+left*, *down+right\_center*, *down\_left*, and *right\_up* are all located in a similar substructure of the manifold. They all describe movements simultaneously up and left. Again this observation is consistent for other transitions such as for example *center\_left* and *right\_center*. This observation explains that the direction of the movement plays an important role in the structure of the dataset.

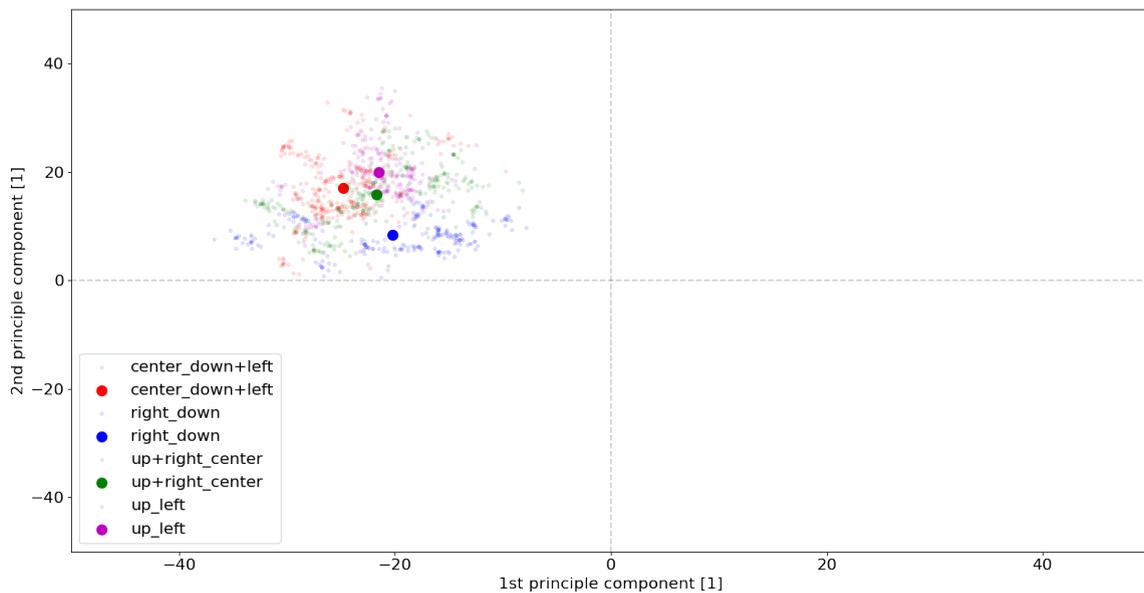
The third observation is that a combined transition, i.e. a transition in two degrees of freedom is projected close to the vector combination of the two transitions in one degree of freedom. For example vector combination of the transition *center\_down* and *center\_left* lies close to the projection of *center\_down+left* on the manifold. Figure 9.11 visualizes two examples of this observation. This observation allows two conclusions: First, it enables reasoning in that combined movements are the result of combining the individual movements. And the second conclusion is that vector combinations on the manifold create semantically meaningful combined movements. As the principal component analysis defines a back-transformation into the original space, this enables obtaining the sensor time series corresponding to the point on the manifold. Therefore, this observation allows the generation of data for combined transitions.

To confirm that the transitions created using vector combination on the manifold are representative of real combined movements, this work compares the time series that is created by transforming the combined point back into the original space with actually recorded combined movements. Figure 9.12 shows the comparison of the sensor time series of the back-transformed derived centroid, the actual centroid of the recorded combined movement's transitions, and three examples of actually recorded transitions. It is observable that the derived time series shows a close similarity with the actual centroid of the combined movement and also with the recordings. Compared with the actual example recordings the difference to the derived transition is overall larger which is not surprising as the examples vary among each other and compared to the centroid as well. This means variance has to be reintroduced to the dataset in order to obtain data that is similar to the recorded dataset.

This work concludes that an ergonomic movement manifold allows explaining and reasoning about ergonomic movements. First, the idea of inverse transitions translates onto the manifold. Second, the direction of movements is important for the manifold structure. And finally, the vector combination of transitions on the manifold allows for reasoning about combined transitions as well as the generation of semantically meaningful movement data. Therefore, the application of manifolds to ergonomic movement data proves a valuable technique for understanding ergonomic motion.



(a) Manifold structure for inverse transitions



(b) Manifold structure for transitions with the same direction

Figure 9.10: Manifold structure for inverse transitions and transitions with the same direction. The plot shows a projection of the recorded transition windows on the first two principal components of the data. The large scatter points are the centroids of each data class. On the manifold inverse transitions (transitions from one posture  $A$  to posture  $B$  and the transition back from  $B$  to  $A$ ) are projected opposite to each other (upper plot). Transitions with the same direction are projected to the same region of the manifold independent of the start- and end-posture (lower plot). Note that for visual clarity, all the projections of other transitions are not plotted.

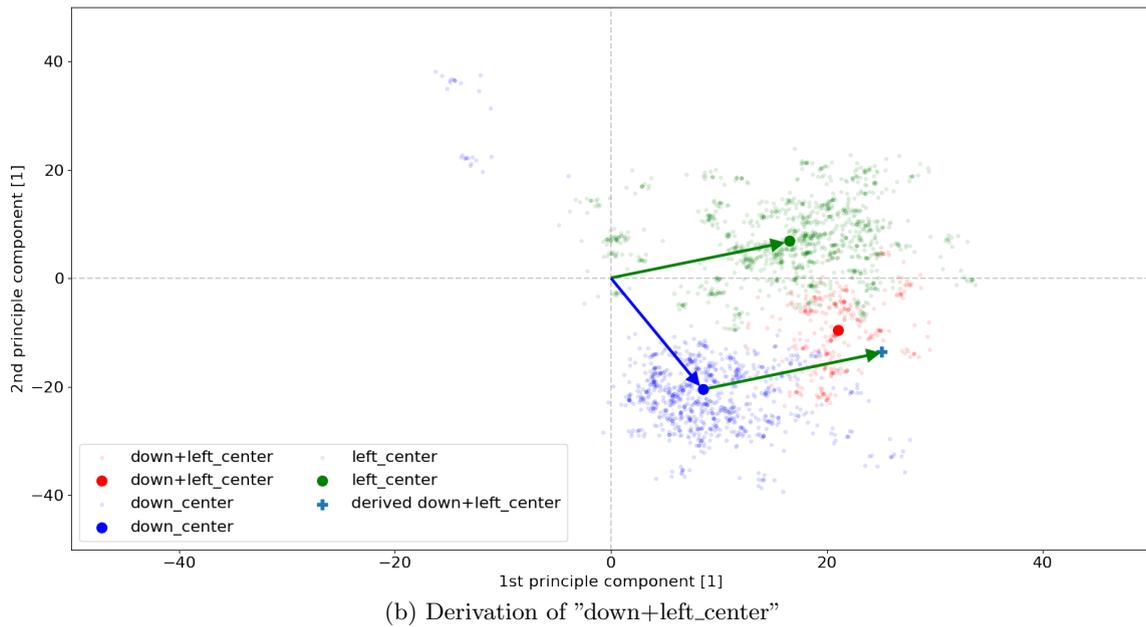
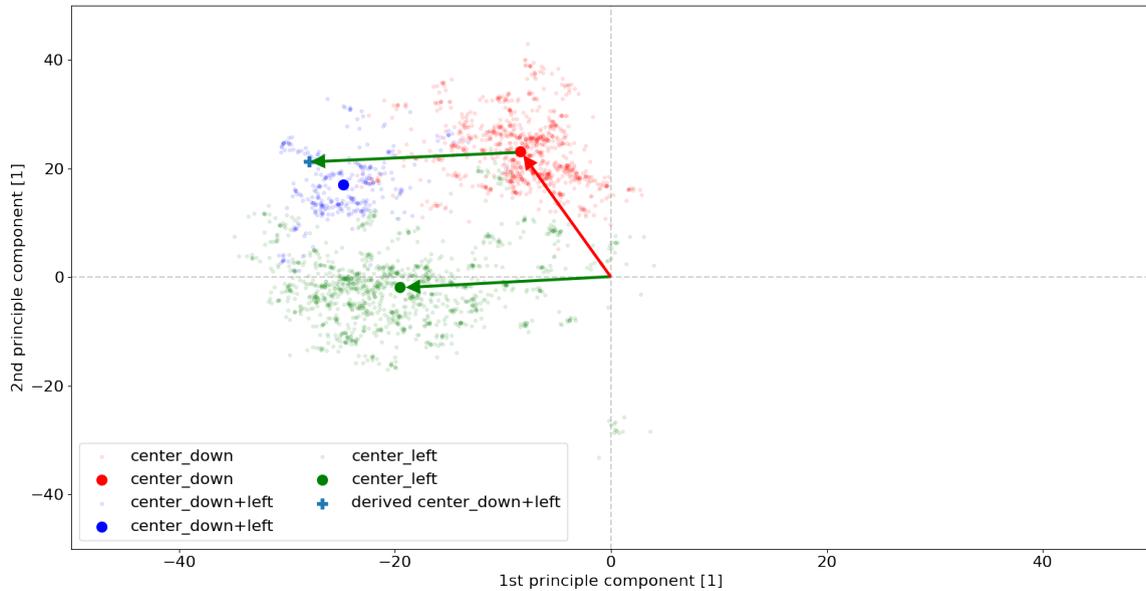
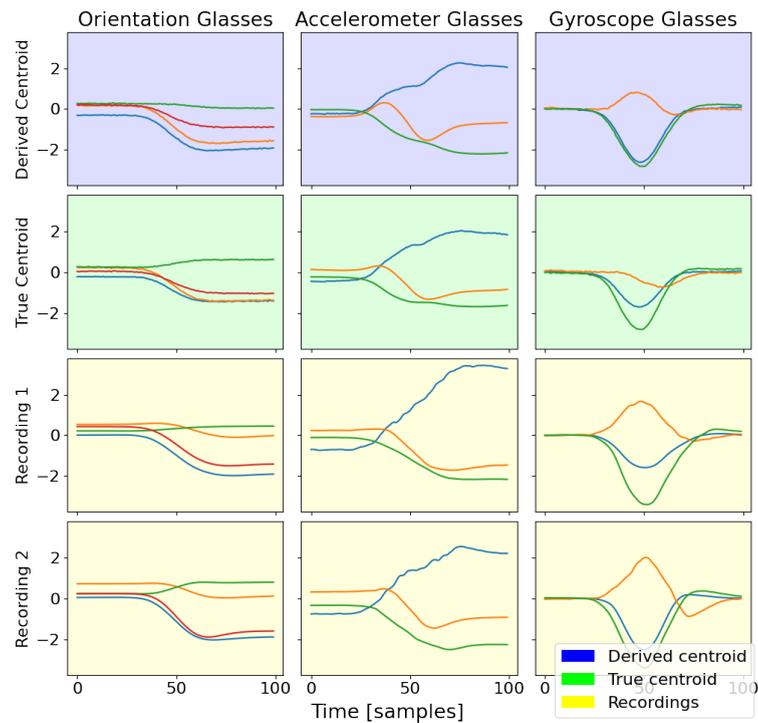
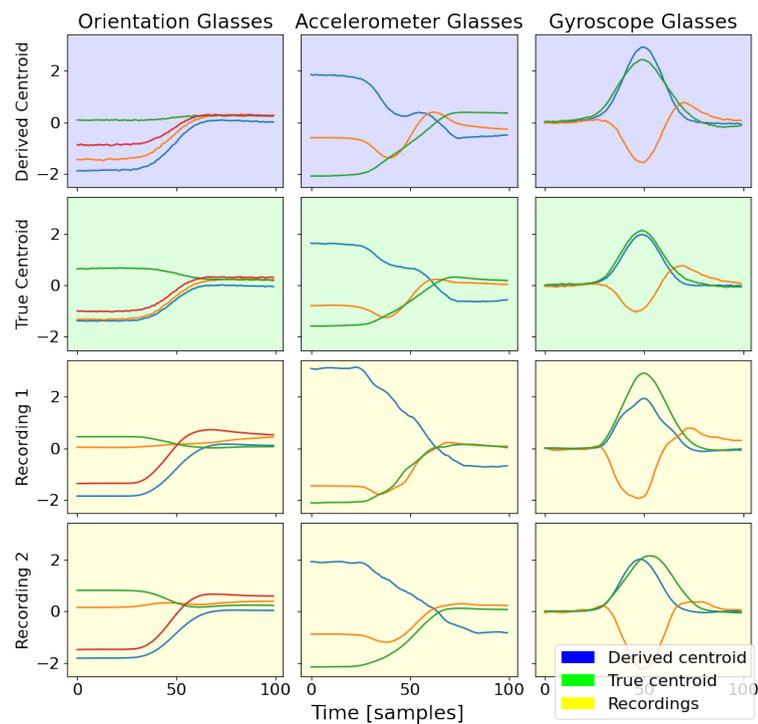


Figure 9.11: Derivation of two example transitions on the manifold. The plots show a projection of the recorded transition windows on the first two principal components of the data. The large scatter points are the centroids of each data class. The plus denotes the derived centroid using vector combination of the vectors from the origin to the centroids of the simple transitions in one direction. The derived centroid lies close to the centroid of the recorded combined movement. Note that for visual clarity, all the projections of other transitions are not plotted.



(a) Data for "center\_down+left"



(b) Data for "down+left\_center"

Figure 9.12: Comparison of derived transition windows using the manifold method and actual recordings. The plots show a comparison of the transition window of the back-transformed derived centroid, the back-transformed centroid time series of the true centroid, and two example actual recordings (rows). For each transition, the orientation, the accelerometer, and gyroscope data for the glasses sensor are plotted (columns). The chair sensor data has been omitted as the regarded movements only affect the head. For easier comparison, all features are standard scaled and are thus unitless.

## 9.7 Summary

This chapter presented a detailed evaluation of the different intelligent sensing techniques the work proposes using data recorded in a controlled environment. The evaluation concludes that the individual steps of the posture extraction pipeline are able to perform their respective purpose. In addition, an end-to-end test of the processing pipeline showed that the posture extraction method this work proposes matches well with the ground truth. With the viability of the approach is proven, the next chapter discusses the results of this work and uses the discussion to outline possible future work.

# Chapter 10

## Discussion

This chapter discusses the main findings of this work. Therefore, Section 10.1 looks at the IMU-based sensor setup, Section 10.2 at the state transition model for posture extraction from the sensor data, and Section 10.3 at the ergonomic movement manifold for explaining and reasoning about ergonomic movement data in detail. This is followed by a discussion of ethical considerations (Section 10.4) and the personalization of the ergonomic assessment (Section 10.5).

### 10.1 Sensing Posture with IMUs

Using inertial measurement units to sense human posture provides a number of advantages over other sensing approaches like cameras. These advantages are lower processing power and battery requirements, and higher user privacy. With the goal of implementing the signal processing on-device, the use of IMUs is a natural choice in this work.

Nevertheless, the use of IMUs also presents inherent difficulties, such as the accumulation of sensor drift during sensor fusion, which make them difficult to use in long-term, mostly static environments such as desk work. However, by using the state transition model, this work provides an approach that addresses sensor drift. It also allows leveraging the dynamic sensor data from the accelerometer and gyroscope in addition to sensor fusion. The state transition model is thus an important contribution to the use of IMUs, as it can mitigate the difficulties and better exploit the advantages of their use for the application.

This work uses a sensor setup with one sensor placed on the frame of wearable glasses and one sensor on the armrest of the desk chair. The placements of the sensors were specifically chosen so that they can sense the posture while impeding the user as little as possible. Using this setup to record data, it was shown that the state transition model can derive the four regarded postures well. More and differently placed sensors might be useful to detect more postures and enhance accuracy. In particular, the attachment of additional sensors to the user's backrest and seating platform looks promising for providing even more details about the user's sitting posture.

## 10.2 State Transition Model

Typical motion data at a desk workstation is mostly static with short dynamic periods where the user transitions between postures. This work leveraged these characteristics of the data to create a domain-specific machine learning method, the state transition model. Instead of extracting the posture directly from the raw data, this model first extracts and classifies the transitions between postures and then aggregates the transitions to derive the posture.

Focusing only on the transitions where the user is moving instead of constantly inferring the posture optimizes power consumption by only classifying when something is happening. It also eliminates the effect of sensor drift in the IMU data by reducing the time spans under consideration. To make this approach work, transitions between postures must be identifiable. Looking at real-world data, there are clear transitions that are well detected using the presented transition detection approach. However, in rare cases, if the user moves very slowly and drifts from one posture to another, it may not always be possible to detect the movement. To solve issues when transitions have not been detected for a long time but the sensor orientations changed significantly, this work artificially inserts transitions.

The classification for the posture transitions in each transition window plays an important role in the state transition model. When developing the state transition model further and including even more postures, one particular challenge is that the number of possible transitions grows quadratically with the number of possible postures. This work reduces the number of regarded transitions by filtering out transitions that don't occur often in practise. Nevertheless, it is worth thinking about other approaches to reduce the number of regarded transitions. One idea would be to only classify and aggregate the relative movement, but not the start- and end-posture of a transition.

This work uses Hidden Markov Models to derive the posture sequence from the classified transitions sequence. The evaluation shows that this approach is able to resolve errors of the transition classifier well whenever a single transition is misclassified. When multiple transitions are misclassified in a row, the error-correction capabilities of the HMM are not sufficient. This is due to HMMs only considering the current posture when determining the probability for the next posture. In practice, this is not very often the case due to the good classification performance of the transition classifier. Nevertheless, when sufficient sequence data is available, it would be interesting to investigate if other sequence models that consider more timesteps, e.g. RNNs can correct these errors as well.

The evaluation of the model using recorded data reveals that the extracted posture using the state transition model matches well with the ground truth. Also, extensions and variations of the submodules of the state transition model show the potential to improve the model performance. Although the results are already promising, it is expected that additional tuning and testing of the different approaches can further improve performance. In particular, biasing the posture model with domain knowledge was shown to be less beneficial than expected in this work. This leaves the open question of whether other approaches to bias may offer greater performance improvements. Overall, the state transition model offers great potential for the extraction of posture from long-term IMU data under low-power constraints.

## 10.3 Ergonomic Movement Manifold

The application of manifolds to ergonomic movements at the desk workstation is mostly unexplored. This work investigates the explainability and reasoning of movements through projecting the transition windows onto the first principal components of the data. The

evaluation of this approach shows that interesting structures of the dataset are indeed uncovered.

The opposite projection of inverse transitions and the grouping of transitions with the same direction represent notable structures on the manifold that help explain the motion data. Apart from explaining the movement data, these observations also suggest the feasibility of other interesting approaches. For example, the already mentioned grouping of similarly directed transitions suggests the feasibility of focusing only on the direction instead of classifying start- and end-posture to transition classification.

Using vector combination, this work demonstrates the application of manifolds to the movement data to reason about combined movements and generate semantically meaningful data. Such a vector combination was only shown for transitions that start or end in the same posture, or for two transitions in the same direction. This leaves open whether and how similarly promising vector combinations exist for other combinations of motions on the manifold.

This work also examined if the generated data affects the posture extraction performance. While the evaluation showed that the addition of generated data doesn't negatively impact the transition classifier performance, the overall performance was reduced slightly. One possible reason is the disability of the application bias to create reasonable probabilities for the HMM rather than the quality of the created data itself. Further investigation into using the generated data may be required to use the approach.

## 10.4 Ethical Considerations

The creation of personal health information and private data that can be traced back to a person makes the consideration of ethical issues indispensable. Wearable devices allow the collection of a vast amount of highly sensitive data such as health-related workplace ergonomic data. The collection and transmission of highly personal data raise serious concern over the privacy and data misuse for the users [75]. To address this issue, this work actively considers privacy in the development of the solution.

This work specifically considers the on-device processing of data. The entire processing pipeline is envisioned to be implemented directly on the device, without any data having to leave the system and only the final assessment scores being transmitted. Therefore, the collected critical data is confined to the system, and only the absolutely necessary information is transmitted to the user. Thus, on-device signal processing is of critical importance for a privacy-conform wearable solution for desk ergonomics. Consequentially, on-device assessment of desk workplace ergonomic risk factors provides a great opportunity to reduce the inherent tension between privacy and automatic ergonomic assessment.

Besides computing and power considerations for on-device processing, the use of cameras is very sensitive with respect to the user's privacy. Apart from body posture, the video image contains a lot of highly sensitive and personal information about the user and their environment. The problem is amplified by the fact that most of the information in the video is not necessary for assessing the user's posture. Using IMUs with motion sensors to capture the user's body movements is thus a more purpose-built and privacy-friendly sensing approach.

Finally, the user has to actively consent to the use of the system and is able to dissent the use at any time. Since the user themselves is the only subject of both the wearable sensor system and the resulting assessment, they are in complete control of using the system. A future solution must resolve the trade-off between the benefits of ergonomic assessment and the protection of the user's private data.

## 10.5 Personalization

The ergonomic disposition of each individual user varies greatly. A user's physical disposition, their psychosocial situation, and the possibility of compensating for physical inactivity and poor posture through sports are just a few examples of personal factors. The approach presented in this work offers advancements for personalized ergonomic risk factor assessment.

The IMU-based sensing approach allows for an automatic assessment of the ergonomic risk situation of the user based on their personal sensor data. Therefore, the sensing-based assessment approach is inherently better personalized than using for example standardized checklists. This work establishes the feasibility of the proposed sensing approach for one particular dual monitor setup with a typical desk chair. To be able to provide such a personalized assessment to a wide range of users, a larger variety of desk setups have to be considered in the future.

Another place where this work explicitly addresses personalization is in the scaling curves of mean and variance to determine the posture and activity scores. By applying different evaluation curves between users or even for each degree of freedom, the individual predisposition of the user can be precisely incorporated into the evaluation. While the practical benefits of personalized scoring are clear, it is required to define the curves in the first place. In addition to the proposal that this work makes, it is envisioned to automatically generate the scoring curves based on the sensor data. An influence of previous medical findings would also be desirable.

This work uses machine learning in the state transition model for posture extraction. This intentional choice over threshold or rule-based approaches was made because machine learning has proven successful in evaluating human activity data and adapts well to different users and setups. Due to the limited scope of this work, the impact of the use of the system by different users was not further investigated. One logical next step towards the practical is to further study the functionality with many different test subjects and desk setups.

## 10.6 Summary

The discussion showed how the intelligent sensing techniques contribute to a privacy-informed and personalized desk workplace ergonomic risk assessment. The IMU-based sensing solution allows for purpose-bound and low-power sensing. The state transition model then extracts the posture from the sensor data while minimizing the influence of sensor drift and further minimizing the power consumption. Using the ergonomic motion manifold helps to better understand the motion data and provides further insights to improve the state transition model. The efforts to optimize power consumption and reduce computational complexity enable on-device processing of data, which in turn allows local privacy of sensitive data. Finally, the personalized assessment of the user's posture provides them with the ability to improve their personal desk workplace posture and thus their health. The next chapter concludes this work by summarizing the results and providing an outlook for future work.

# Chapter 11

## Conclusion and Outlook

At the desk workstation, many people are exposed to a particularly high ergonomic risk due to potential sustained poor sitting posture and lack of activity while sitting. Since desk work will continue to increase and the associated health risks will become more dramatic, the ergonomic assessment of the desk workplace is becoming increasingly important. However, the long-term and highly individual nature of desk work ergonomics makes assessing the risk situation very difficult. Previous work has not been able to create an established solution for the continuous assessment of ergonomic risk factors. The goal of this work was to explore intelligent sensing techniques for the continuous assessment of desk work ergonomic risk factors under considerations of low-power devices and long-term sensing. This chapter summarizes the main findings of this work and provides an outlook on future work in the area.

Poor posture and low sitting activity are particularly important ergonomic risk factors at the desk workplace. To provide an assessment of the personal risk situation of a user this work created a sensor setup using IMUs placed on the frames of wearable glasses and the armrest of the desk chair that sense relevant movements of the user. A domain-specific AI-approach, the state transition model, derives the posture from the raw sensor data while utilizing the typical mostly-static nature of desk workplace data to optimize power consumption and long-term accuracy. Instead of deriving the posture directly, this approach first extracts only the transitions between postures from the sensor data and then aggregates the transitions in a second step to derive the posture. To assess the posture for the ergonomic risk factors of poor posture and low sitting activity, this work derives the postural tendency score and the postural activity score. The ergonomic assessment is used to provide the user action propositions to improve their posture whenever the scores indicate high ergonomic risk. Finally, this work creates an ergonomic movement manifold and demonstrates its use for explainability and reasoning about ergonomic movement data.

A practical implementation is used to evaluate the intelligent sensing techniques with data collected in a controlled desk workplace environment. The state transition model for posture extraction achieved promising results with an overall time-scaled partial accuracy of 0.9123. In addition to comparing different end-to-end posture extraction pipelines, this work evaluates the transition extraction, transition classification, and posture derivation in

detail. The results demonstrate that the posture derived using the state transition model on the data recorded with the proposed sensor setup matches the ground truth well. The projection of the recorded movement data onto the ergonomic movement manifold reveals interesting structures of the dataset: Inverse transitions are projected opposite, transitions with the same direction share the same region, and the combination of two movements on the manifold is similar to the recorded combined movement. These results show the applicability of manifolds to ergonomic movement data.

This work presented a variety of novel contributions to the area of intelligent sensing techniques for desk workplace ergonomics. The sensor setup provides a joint sensing approach for assessing head and trunk posture. The use of IMUs to measure posture allows for higher user privacy as well as lower compute and battery power requirements compared to previous camera-based approaches. Special emphasis was put on developing a sensor setup that does not impede the user in contrast to previous approaches using IMUs. The state transition model does not only alleviate inherent issues of the use of IMUs over long periods of time but also further reduces power usage while extracting posture. Due to the low computational and battery power requirements of the processing pipeline developed in this work, the processing is envisioned to be performed directly on the device. This enables local privacy of the user's sensitive health data. With the application of manifolds to ergonomic movement data, this work investigated a technique that is barely explored in previous work on desk work ergonomics. This technique does not only allow a better understanding of ergonomic movements but also new approaches to reason about or even generate data for them. Overall, this work provides valuable insights into the personalized assessment of desk workplace ergonomic risk factors and provides non-invasive supporting technology with local privacy.

The proposed solution enables providing a personalized ergonomic risk assessment to individual users and can be used as a supporting technology to enable long-term medical studies on ergonomics at the desk workstation. However, there is still much to research and develop in the field of intelligent sensor technology for the ergonomics of desk workstations. Moving forward, an in-depth study of the approach for a wide range of users and desk configurations is important to demonstrate general applicability. Another logical next step is to implement the processing pipeline on-device evaluate the power and processing requirements in practice. Also, adding sensors to the backrest and seat of the desk chair promises to collect more detailed data on the user's posture. This could enable looking at other postures that were not previously regarded. Beyond ergonomics at the desk workstation, there are a variety of application areas with similar characteristics for which transfer learning might be possible. In particular, a transfer to ergonomics for professional drivers and people who read a lot, such as school children, is possible since they are also strongly affected by prolonged sitting postures.

# Bibliography

- [1] M. Rana and V. Mittal, “Wearable sensors for real-time kinematics analysis in sports: a review,” *IEEE Sensors Journal*, vol. 21, no. 2, pp. 1187–1207, 2020.
- [2] D. Nahavandi, R. Alizadehsani, A. Khosravi, and U. R. Acharya, “Application of artificial intelligence in wearable devices: Opportunities and challenges,” *Computer Methods and Programs in Biomedicine*, vol. 213, p. 106541, 2022.
- [3] P. Picerno, M. Iosa, C. D’Souza, M. G. Benedetti, S. Paolucci, and G. Morone, “Wearable inertial sensors for human movement analysis: a five-year update,” *Expert Review of Medical Devices*, vol. 18, no. sup1, pp. 79–94, 2021, pMID: 34601995. [Online]. Available: <https://doi.org/10.1080/17434440.2021.1988849>
- [4] Parent-Thirion, Agn s and Biletta, Isabella and Cabrita, Jorge and Vargas, Oscar and Vermeulen, Greet and Wilczynska, Aleksandra and Wilkens, Mathijn, “Sixth European Working Conditions Survey – Overview report,” Eurofound, 2017, accessed 22-December-2021. [Online]. Available: [https://www.eurofound.europa.eu/sites/default/files/ef\\_publication/field\\_ef\\_document/ef1634en.pdf](https://www.eurofound.europa.eu/sites/default/files/ef_publication/field_ef_document/ef1634en.pdf)
- [5] Eurostat, the statistical office of the European Union, “Use of computers and the internet by employees,” 2021, accessed 10-January-2022. [Online]. Available: [http://appsso.eurostat.ec.europa.eu/nui/show.do?dataset=isoc\\_ci\\_cm\\_pn2&lang=en](http://appsso.eurostat.ec.europa.eu/nui/show.do?dataset=isoc_ci_cm_pn2&lang=en)
- [6] G. Ari ns, P. Bongers, M. Douwes, M. Miedema, W. Hoogendoorn, G. van der Wal, L. Bouter, and W. van Mechelen, “Are neck flexion, neck rotation, and sitting at work risk factors for neck pain? results of a prospective cohort study,” *Occupational and environmental medicine*, vol. 58, no. 3, pp. 200–207, 2001.
- [7] L. Ortiz-Hern ndez, S. Tamez-Gonz lez, S. Mart nez-Alc ntara, and I. M ndez-Ram rez, “Computer use increases the risk of musculoskeletal disorders among newspaper office workers,” *Archives of medical research*, vol. 34, no. 4, pp. 331–342, 2003.
- [8] Stacey, Nicola and Ellwood, Peter and Bradbook, Sam and Reynolds, John and Williams, Huw, “Key trends and drivers of change in information and communication technologies and work location,” European Agency for Safety and Health at Work (EU-OSHA), 2017, accessed 04-January-2022. [Online]. Available: [https://osha.europa.eu/sites/default/files/Key\\_trends\\_drivers\\_%20information\\_communication\\_technologies.pdf](https://osha.europa.eu/sites/default/files/Key_trends_drivers_%20information_communication_technologies.pdf)
- [9] Irastorzan, Xabier, “Third European Survey of Enterprises on New and Emerging Risks (ESENER 3) - First Findings,” European Agency for Safety and Health at Work (EU-OSHA), 2019, accessed 04-January-2022. [Online]. Available: [https://osha.europa.eu/sites/default/files/ESENER\\_3\\_first\\_findings.pdf](https://osha.europa.eu/sites/default/files/ESENER_3_first_findings.pdf)
- [10] P. Paliyawan, C. Nukoolkit, and P. Mongkolnam, “Prolonged sitting detection for office workers syndrome prevention using kinect,” in *2014 11th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, 2014, pp. 1–6.

- [11] Y.-C. Wu, T.-Y. Wu, P. Taelle, B. Wang, J.-Y. Liu, P.-s. Ku, P.-E. Lai, and M. Y. Chen, *ActiveErgo: Automatic and Personalized Ergonomics Using Self-Actuating Furniture*. New York, NY, USA: Association for Computing Machinery, 2018, pp. 1–8. [Online]. Available: <https://doi.org/10.1145/3173574.3174132>
- [12] B. Prueksanusak, P. Rujvapatand, and K. Wongpatikaseree, “An ergonomic chair with internet of thing technology using svm,” in *2019 4th Technology Innovation Management and Engineering Science International Conference (TIMES-iCON)*, 2019, pp. 1–5.
- [13] H. Jeong and W. Park, “Developing and evaluating a mixed sensor smart chair system for real-time posture classification: Combining pressure and distance sensors,” *IEEE Journal of Biomedical and Health Informatics*, vol. 25, no. 5, pp. 1805–1813, 2020.
- [14] D. Jun, V. Johnston, S. M. McPhail, and S. O’Leary, “Are measures of postural behavior using motion sensors in seated office workers reliable?” *Human Factors*, vol. 61, no. 7, pp. 1141–1161, 2019. [Online]. Available: <https://doi.org/10.1177/0018720818821273>
- [15] I.-C. Severin, “Head posture monitor based on 3 imu sensors: Consideration toward healthcare application,” in *2020 International Conference on e-Health and Bioengineering (EHB)*, 2020, pp. 1–4.
- [16] V. K. K. Sinha, K. K. K. Patro, P. Plawiak, and A. J. J. Prakash, “Smartphone-based human sitting behaviors recognition using inertial sensor,” *Sensors*, vol. 21, no. 19, p. 6652, 2021.
- [17] Inwerk GmbH, “DE202018006259U1 - Sensorsystem für interaktive Büroergonomie,” 2018, accessed 28-January-2022. [Online]. Available: <https://worldwide.espacenet.com/patent/search?q=pn%3DDE202018006259U1>
- [18] Wilkhahn Wilkening + Hahne GmbH & Co. KG, “US2016095442A1 - Combination comprising a set, a portable device which comprises a movement sensor and an evaluation device and a method which can be carried out using the combination,” 2015, accessed 28-January-2022. [Online]. Available: <https://worldwide.espacenet.com/patent/search?q=pn%3DUS2016095442A1>
- [19] F. Lin, C. Song, X. Xu, L. Cavuoto, and W. Xu, “Sensing from the bottom: Smart insole enabled patient handling activity recognition through manifold learning,” in *2016 IEEE first international conference on connected health: applications, systems and engineering technologies (CHASE)*. IEEE, 2016, pp. 254–263.
- [20] S. Chen, S. S. Bangaru, T. Yigit, M. Trkov, C. Wang, and J. Yi, “Real-time walking gait estimation for construction workers using a single wearable inertial measurement unit (imu),” in *2021 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*. IEEE, 2021, pp. 753–758.
- [21] International Ergonomics Association, “What Is Ergonomics?” International Ergonomics Association, 2021, accessed 07-January-2022. [Online]. Available: <https://iea.cc/what-is-ergonomics/>
- [22] Council of the European Union, “Council directive 89/391/eec on the introduction of measures to encourage improvements in the safety and health of workers at work,” Directorate-General for Employment, Social Affairs and Inclusion, DG05/F, 2008, accessed 10-January-2022. [Online]. Available: <https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A01989L0391-20081211>
- [23] F. Tosi, *Ergonomics and Design*. Cham: Springer International Publishing, 2020, pp. 3–29. [Online]. Available: [https://doi.org/10.1007/978-3-030-33562-5\\_1](https://doi.org/10.1007/978-3-030-33562-5_1)

- [24] N. Jaffar, A. Abdul-Tharim, I. Mohd-Kamar, and N. Lop, "A literature review of ergonomics risk factors in construction industry," *Procedia Engineering*, vol. 20, pp. 89–97, 2011, 2nd International Building Control Conference. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1877705811029511>
- [25] P. K. Nag, "Fundamentals of office ergonomics," in *Office Buildings*. Springer, 2019, pp. 129–148.
- [26] R. Zemp, M. Fliesser, P.-M. Wippert, W. R. Taylor, and S. Lorenzetti, "Occupational sitting behaviour and its relationship with back pain—a pilot study," *Applied ergonomics*, vol. 56, pp. 84–91, 2016.
- [27] de Langen, Nicolien and Peereboom, Kees, "Musculoskeletal disorders and prolonged static sitting," vhp human performance, 2020, accessed 07-Januaray-2021. [Online]. Available: [https://oshwiki.eu/wiki/Musculoskeletal\\_disorders\\_and\\_prolonged\\_static\\_sitting](https://oshwiki.eu/wiki/Musculoskeletal_disorders_and_prolonged_static_sitting)
- [28] Gregory, Nicole, "Office Ergonomics And How It Affects Your Health," Forbes Health, 2021, accessed 03-January-2022. [Online]. Available: <https://www.forbes.com/health/body/office-ergonomics/>
- [29] M. J. Pereira, V. Johnston, L. M. Straker, G. Sjøgaard, M. Melloh, S. P. O’Leary, and T. A. Comans, "An investigation of self-reported health-related productivity loss in office workers and associations with individual and work-related factors using an employer’s perspective," *Journal of occupational and environmental medicine*, vol. 59, no. 7, pp. e138–e144, 2017.
- [30] Google, "Google Fit," Google, 2022, accessed 11-January-2022. [Online]. Available: <https://www.google.com/fit/>
- [31] Apple, "Apple Fitness+," Apple, 2022, accessed 11-January-2022. [Online]. Available: <https://www.apple.com/de/apple-fitness-plus/>
- [32] Samsung, "Enhance your life with Samsung Health," Samsung, 2022, accessed 11-January-2022. [Online]. Available: <https://www.samsung.com/global/galaxy/apps/samsung-health/>
- [33] Apple, "The health app. With a healthy dose of updates." Apple, 2022, accessed 11-January-2022. [Online]. Available: <https://www.apple.com/de/ios/health/>
- [34] N. Akkarakittichoke and P. Janwantanakul, "Seat pressure distribution characteristics during 1 hour sitting in office workers with and without chronic low back pain," *Safety and health at work*, vol. 8, no. 2, pp. 212–219, 2017.
- [35] G. A. Ariens, W. Van Mechelen, P. M. Bongers, L. M. Bouter, and G. Van Der Wal, "Physical risk factors for neck pain," *Scandinavian journal of work, environment & health*, pp. 7–19, 2000.
- [36] K. C. Wong, R. Y. Lee, and S. S. Yeung, "The association between back pain and trunk posture of workers in a special school for the severe handicaps," *BMC musculoskeletal disorders*, vol. 10, no. 1, pp. 1–8, 2009.
- [37] A. A. Shikdar and M. A. Al-Kindi, "Office ergonomics: deficiencies in computer workstation design," *International Journal of Occupational Safety and Ergonomics*, vol. 13, no. 2, pp. 215–223, 2007.
- [38] L. Dimberg, J. Goldoni Laestadius, S. Ross, and I. Dimberg, "The changing face of office ergonomics," *The Ergonomics open journal*, vol. 8, no. 1, 2015.

- [39] Wikipedia contributors, “Inertial measurement unit — Wikipedia, the free encyclopedia,” 2022, accessed 25-January-2022. [Online]. Available: [https://en.wikipedia.org/w/index.php?title=Inertial\\_measurement\\_unit&oldid=1067073686](https://en.wikipedia.org/w/index.php?title=Inertial_measurement_unit&oldid=1067073686)
- [40] M. Kok, J. D. Hol, and T. B. Schön, “Using inertial sensors for position and orientation estimation,” *arXiv preprint arXiv:1704.06053*, 2017.
- [41] L. E. Meyer, L. Porter, M. E. Reilly, C. Johnson, S. Safir, S. F. Greenfield, B. C. Silverman, J. I. Hudson, and K. N. Javaras, “Using wearable cameras to investigate health-related daily life experiences: A literature review of precautions and risks in empirical studies,” *Research Ethics*, vol. 18, no. 1, pp. 64–83, 2022. [Online]. Available: <https://doi.org/10.1177/17470161211054021>
- [42] Ryan Lee, “Flickr - How to Sit at your Desk Properly,” 2021, accessed 21-December-2021. [Online]. Available: <https://www.flickr.com/photos/190784293@N05/50859720417>
- [43] Bosch Sensortec, “Arduino Nicla Sense ME - A tiny board for sensing the big world.” 2021, accessed 21-December-2021. [Online]. Available: <https://www.bosch-sensortec.com/software-tools/tools/arduino-nicla-sense-me/>
- [44] —, “Software: Sensor fusion software - Smart. Dynamic. Distortion-free,” 2021, accessed 21-December-2021. [Online]. Available: <https://www.bosch-sensortec.com/software-tools/software/sensor-fusion-software/>
- [45] —, “BME688 - Datasheet,” p. 11, 2021, accessed 21-December-2021. [Online]. Available: <https://www.bosch-sensortec.com/media/boschsensortec/downloads/datasheets/bst-bme688-ds000.pdf>
- [46] Y.-C. Wu, Q. Chaudhari, and E. Serpedin, “Clock synchronization of wireless sensor networks,” *IEEE Signal Processing Magazine*, vol. 28, no. 1, pp. 124–138, 2010.
- [47] Pandas Library Authors, “pandas.merge\_asof - pandas 0.25.0 documentation,” 2021, accessed 16-December-2021. [Online]. Available: [https://pandas.pydata.org/pandas-docs/version/0.25.0/reference/api/pandas.merge\\_asof.html](https://pandas.pydata.org/pandas-docs/version/0.25.0/reference/api/pandas.merge_asof.html)
- [48] F. Mohd-Yasin, D. J. Nagel, and C. E. Korman, “Noise in mems,” *Measurement Science and Technology*, vol. 21, no. 1, p. 012001, 2009.
- [49] Sarle, Warren S, “comp.ai.neural-nets faq, part 2 of 7: Learning,” 2002, accessed 25-January-2022. [Online]. Available: <http://www.faqs.org/faqs/ai-faq/neural-nets/part2/>
- [50] S. Beauregard and H. Haas, “Pedestrian dead reckoning: A basis for personal positioning,” in *Proceedings of the 3rd Workshop on Positioning, Navigation and Communication*, 2006, pp. 27–35.
- [51] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, <http://www.deeplearningbook.org>.
- [52] S. Ha and S. Choi, “Convolutional neural networks for human activity recognition using multiple accelerometer and gyroscope sensors,” in *2016 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2016, pp. 381–388.
- [53] A. N. Mazumder, H. Ren, H.-A. Rashid, M. Hosseini, V. Chandraredy, H. Homayoun, and T. Mohsenin, “Automatic detection of respiratory symptoms using a low power multi-input cnn processor,” *IEEE Design & Test*, 2021.
- [54] Y. Wang, L. Xia, T. Tang, B. Li, S. Yao, M. Cheng, and H. Yang, “Low power convolutional neural networks on a chip,” in *2016 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2016, pp. 129–132.

- [55] Jurafsky, Daniel and Martin, James H., “Hidden Markov Models,” 2021, accessed 08-February-2022. [Online]. Available: <https://web.stanford.edu/~jurafsky/slp3/A.pdf>
- [56] R. S. Chavan and G. S. Sable, “An overview of speech recognition using hmm,” *International Journal of Computer Science and Mobile Computing*, vol. 2, no. 6, pp. 233–238, 2013.
- [57] J. Söding, “Protein homology detection by hmm–hmm comparison,” *Bioinformatics*, vol. 21, no. 7, pp. 951–960, 2005.
- [58] Maizlish, Aaron, “Orometry: Introduction to Prominence,” 2004, accessed 22-February-2022. [Online]. Available: [http://www.peaklist.org/theory/orometry/article/Orometry\\_1.html](http://www.peaklist.org/theory/orometry/article/Orometry_1.html)
- [59] Wikipedia contributors, “Topographic prominence — Wikipedia, the free encyclopedia,” 2021, accessed 7-December-2021. [Online]. Available: [https://en.wikipedia.org/w/index.php?title=Topographic\\_prominence&oldid=1054822805](https://en.wikipedia.org/w/index.php?title=Topographic_prominence&oldid=1054822805)
- [60] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.
- [61] S. Bai, J. Z. Kolter, and V. Koltun, “An empirical evaluation of generic convolutional and recurrent networks for sequence modeling,” 2018.
- [62] S. Boodidhi, “Using smoothing techniques to improve the performance of hidden markov’s model,” Ph.D. dissertation, University of Nevada, Las Vegas, 2011.
- [63] Y. Ma and Y. Fu, *Manifold learning theory and applications*. CRC press Boca Raton, FL, 2012, vol. 434.
- [64] A. Elgammal and C.-S. Lee, “The role of manifold learning in human motion analysis,” in *Human Motion*. Springer, 2008, pp. 25–56.
- [65] L. Cayton, “Algorithms for manifold learning,” *Univ. of California at San Diego Tech. Rep*, vol. 12, no. 1-17, p. 1, 2005.
- [66] Karthe, “Pca: A practical guide to principal component analysis in r & python,” 2016, accessed 25-January-2022. [Online]. Available: <https://www.analyticsvidhya.com/blog/2016/03/pca-practical-guide-principal-component-analysis-python/>
- [67] I. T. Jolliffe and J. Cadima, “Principal component analysis: a review and recent developments,” *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 374, no. 2065, p. 20150202, 2016.
- [68] J. Pennington, R. Socher, and C. D. Manning, “Glove: Global vectors for word representation,” in *Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1532–1543. [Online]. Available: <http://www.aclweb.org/anthology/D14-1162>
- [69] Interior business association, “Quality criteria for office desks,” 2021, accessed 27-January-2022. [Online]. Available: <https://iba.online/site/assets/files/2013/doppelarbeitsplatz.640x0.jpg>
- [70] Sebastian Romero, “Arduino Nicla Sense ME Cheat Sheet,” 2021, accessed 21-December-2021. [Online]. Available: <https://docs.arduino.cc/tutorials/nicla-sense-me/cheat-sheet>
- [71] C. L. R. Wieland, “Domain knowledge infusion in machine learning for digital signal processing applications : An in-depth case study on table tennis stroke recognition,” Master’s thesis, Karlsruher Institut für Technologie (KIT), 2021.

- 
- [72] Keras Team, “Keras FAQ - How can I obtain reproducible results using Keras during development?” 2022, accessed 12-January-2022. [Online]. Available: [https://keras.io/getting\\_started/faq/#how-can-i-obtain-reproducible-results-using-keras-during-development](https://keras.io/getting_started/faq/#how-can-i-obtain-reproducible-results-using-keras-during-development)
- [73] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, “Bleu: a method for automatic evaluation of machine translation,” in *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, 2002, pp. 311–318.
- [74] Lee, Chin Yee and Li, Hengfeng and Hou, Ruxin and Lim, Calvin Tanujaya, “Documentation - Source code for nltk.translate.bleu\_score,” 2022, accessed 18-February-2022. [Online]. Available: [https://www.nltk.org/\\_modules/nltk/translate/bleu\\_score.html](https://www.nltk.org/_modules/nltk/translate/bleu_score.html)
- [75] V. G. Motti and K. Caine, “Users’ privacy concerns about wearables,” in *International Conference on Financial Cryptography and Data Security*. Springer, 2015, pp. 231–244.

# Appendix

## A Expert Interview

In the context of this work, a qualitative expert interview with a physiotherapist with over 20 years of experience in treating conditions associated with desk work was conducted. The interview consisted of a number of open questions. The goal of the interview was to get an impression of relevant factors of desk ergonomics and the treatment of affected patients from the medical point of view. The following section shows a summary of the interview. Clarifications of the answers are inserted in square brackets.

- *Q: In your day-to-day work, how much of your work relates to issues caused by prolonged sitting / sitting posture?*  
A: Roughly 20-40% of patients. There are differences between patients with issues due to sitting at the computer and other reasons for prolonged sitting.
- *Q: What are the most frequent medical conditions that you see caused by work ergonomic issues?*  
A: In severe cases, mainly herniated discs. Of course, also muscle tensions in the neck and cervical muscles. There is also the elbow problem due to mouse usage which is often a muscular malposition. There are also issues with the hip joint due to bent sitting. The dominant issues are muscle tensions.
- *Q: Is there any distribution in people that are particularly very/less likely to be affected?*  
A: There is a slight increase for the group of people of age 40-50. However, also many younger people are affected. Younger people are more likely to be affected by muscle tensions. For the older group issues are more likely already degenerative problems. Less affected are people who have compensation, through exercise, relaxation techniques, yoga, sports, or similar. However, I also have athletes as patients that have problems due to prolonged sitting.
- *Q: How relevant are issues to the people you treat?*  
A: This is really highly variant. For some patients, it is only an inconvenience due to muscle tensions which they can compensate for with treatment and heat. For others, they have paresthesias [abnormal sensations of the skin and radiating pain], numbness, and possibly even motor limitations. With herniated discs, there is often reflex attenuation. In that case, operations might be necessary. In particularly severe

cases, it is necessary to take painkillers and antidepressants during particularly long periods of illness.

- *Q: How established is the connection between bad posture and musculoskeletal disorders (MSD) in the medical community?*

A: The connection is established. The same is true for the connection between lack of activity and MSDs. However, the actual state of studies is highly problematic. One would really have to accompany people for eight hours during their work. One would have to create a summary of how long the subject remained in which posture. In practice, the connection is very tangible.

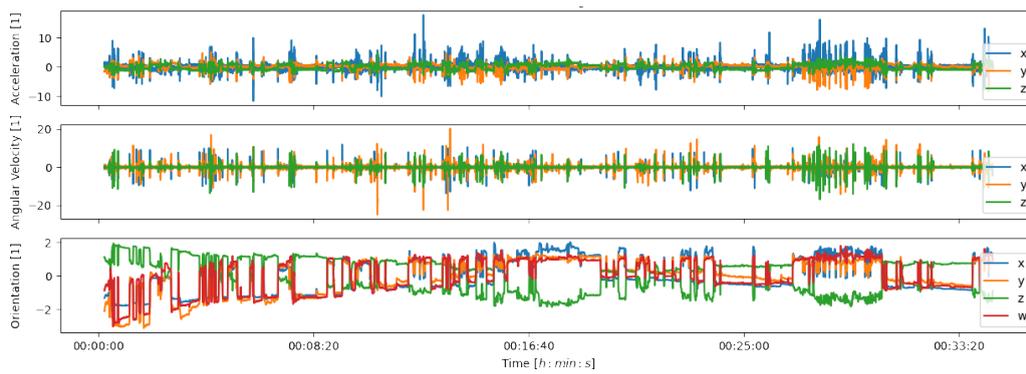
- *Q: Are the issues different between patients?*

A: Yes, highly. The general muscle tone and basic posture of each patient's body are individual. As noted before, how people move outside of the desk strongly affects their risk for postural issues. Nevertheless, even very active people are affected by the problems.

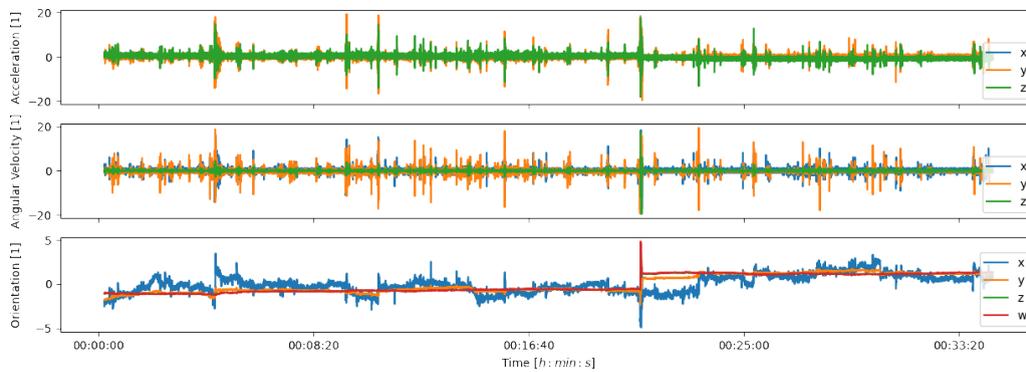
- *Q: What are particular challenges in assessing desk workplace-associated issues?*

A: I am not there with the patient at the desk. Patients have to describe their situation to me. Otherwise, the assessment is based on different physical factors: Head posture, spine, muscle expression, shoulder blades, spreading of the arms, from the side: double S-shape, from the front: belly shape (belly button), collarbone position, etc. From there I have movement patterns performed actively, passively, and isometrically. This provides information about the mobility of the skeleton and the muscles. However, all this has to be seen in relation to the precondition of the person. For example, when a man does a lot of sports is totally normal that his shoulder blades are further apart. For women with narrower shoulders and a wider pelvis, their arm posture is also different.

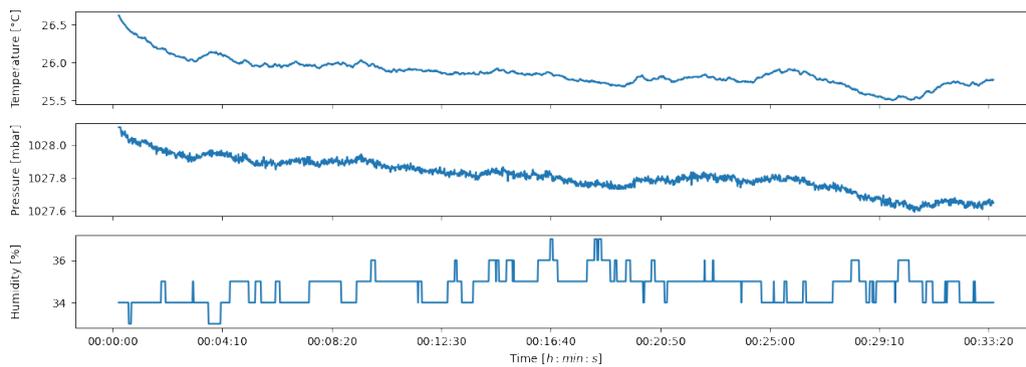
## B Example Raw Data



(a) Raw movement data of the glasses sensor



(b) Raw movement data of the chair sensor



(c) Raw environmental data of the glasses sensor

Figure B.1: Example raw sensor data. Note that the movement sensor data is standard scaled.

## C Dataset Information

Table C.1: Total recorded transitions per transition type and data set

<b>Transition</b>	<b>Training Set</b>	<b>Validation Set</b>	<b>Test Set</b>
center_chairleft	23	5	5
center_chairright	22	5	6
center_down	48	10	11
center_down+left	17	3	4
center_down+right	17	4	4
center_left	72	11	12
center_left+chairleft	19	4	4
center_lefttilt	24	4	5
center_right	62	15	15
center_right+chairright	18	4	5
center_righttilt	22	5	6
center_up	47	11	11
center_up+left	18	3	4
center_up+right	17	4	4
chairleft_center	22	5	6
chairright_center	22	4	4
down+left_center	16	4	4
down+right_center	16	4	5
down_center	46	11	12
down_left	12	2	3
down_right	11	2	2
left+chairleft_center	19	4	4
left+chairleft_left	8	1	2
left_center	63	13	14
left_down	11	3	3
left_left+chairleft	8	1	2
left_right	45	10	10
left_up	13	3	3
lefttilt_center	23	5	5
right+chairright_center	17	4	5
right+chairright_right	7	2	2
right_center	71	18	19
right_down	12	3	3
right_left	32	7	8
right_right+chairright	7	2	2
right_up	12	2	3
righttilt_center	22	5	6
up+left_center	18	3	3
up+right_center	17	4	4
up_center	47	11	11
up_left	11	3	3
up_right	13	3	3
<b>Total</b>	<b>1047</b>	<b>226</b>	<b>247</b>