

MASTERARBEIT | MASTER'S THESIS

Titel | Title

Vom Sensor zur Klassifikation: Fourierbasierte
Aktivitätserkennung in Sportuhren

verfasst von | submitted by
Sabrina Gruber BEd

angestrebter akademischer Grad | in partial fulfilment of the requirements for the degree of
Master of Education (MEd)

Wien | Vienna, 2025

Studienkennzahl lt. Studienblatt | Degree
programme code as it appears on the
student record sheet:

UA 199 500 520 02

Studienrichtung lt. Studienblatt | Degree
programme as it appears on the student
record sheet:

Masterstudium Lehramt Sek (AB) Unterrichtsfach
Bewegung und Sport Unterrichtsfach Mathematik

Betreut von | Supervisor:

Mag. Dr. Monika Dörfler Privatdoz.

Abstract (Deutsch)

Die zunehmende Verbreitung von Wearables, insbesondere Sportuhren, hat die sensorbasierte Aktivitätserkennung (Human Activity Recognition, HAR) zu einem zentralen Forschungsgebiet gemacht. Etwa 30% der Europäer nutzen bereits Wearables. Während existierende HAR-Surveys primär Machine Learning-Ansätze fokussieren, fehlt eine systematische mathematische Analyse der Signaltransformationen speziell für ressourcenbeschränkte Wearable-Systeme. Diese Masterarbeit untersucht die mathematischen Transformationen und Algorithmen, die erforderlich sind, um Rohdaten von Beschleunigungssensoren in aussagekräftige Trainingsanalysen zu überführen. Im Gegensatz zu bestehenden Arbeiten, die Deep Learning-Ansätze untersuchen, liegt der Fokus dieser Arbeit auf der Fourier-Transformation als effizientem Analyseinstrument für periodische Bewegungsmuster. Die Ausnutzung der Symmetrie der Fourierreihe spielt in die Recheneffizienz der Fast Fourier Transform (FFT) hinein. Der gesamte Signalverarbeitungsprozess von der Rohsignalerfassung bis zur Aktivitätsklassifikation wird aufgearbeitet. Dabei werden die spezifischen Herausforderungen ressourcenbeschränkter Wearable-Systeme berücksichtigt, die Echtzeitverarbeitung bei minimaler Energieaufnahme erfordern. Die theoretische Analyse wird durch eine praktische Umsetzung der Sensordatenverarbeitung in MATLAB bestätigt. Diese veranschaulicht exemplarisch anhand von Lauferkennung die einzelnen Verarbeitungsschritte von der Datensegmentierung über die Frequenzanalyse und die Merkmalsextraktion bis zur Klassifikation sowie das Training eines Random Forest Klassifikators. Klassifikationsmodelle erreichen mit Frequenzdomänen-Features eine höhere Erkennungsgenauigkeit von 92.38% gegenüber 83.48% bei der Verwendung reiner Zeitbereichsmerkmale. Die Arbeit liefert eine umfassende mathematische Perspektive auf HAR-Systeme in Sportuhren und zeigt auf, wie die FFT trotz begrenzter Rechenkapazität eine präzise Aktivitätserkennung ermöglicht. Die Ergebnisse tragen zum Verständnis der mathematischen Fundierung moderner Aktivitätserkennung bei.

Abstract (English)

The growing popularity of wearable devices, particularly sports watches, has made sensor-based human activity recognition (HAR) a key area of research. Around 30% of Europeans already use wearable devices. While existing HAR surveys primarily focus on machine learning approaches, a systematic mathematical analysis of signal transformations for wearable systems with limited resources remains absent. This Master's thesis investigates the mathematical transformations and algorithms required to convert raw accelerometer sensor data into meaningful training analyses. Unlike existing studies that examine deep learning approaches, this work focuses on the Fourier transform as an efficient analytical tool for identifying periodic movement patterns. The entire signal processing pipeline, from acquiring the raw signal to classifying the activity, is examined in detail. The specific challenges of real-time processing with minimal energy consumption in resource-constrained wearable systems are taken into account. The theoretical analysis is validated by the practical implementation of sensor data processing in MATLAB. The processing steps from data segmentation to frequency analysis, feature extraction and classification, and the training of a Random Forest classifier are exemplified. Running recognition is used as a case study. The exploitation of the symmetry of the Fourier series contributes to the computational efficiency of the Fast Fourier Transform (FFT). Classification models achieve a higher recognition accuracy of 92.38% with frequency domain features compared to 83.48% when using pure time-domain features. This work provides a comprehensive mathematical perspective on HAR systems in sports watches and demonstrates how the FFT enables precise activity recognition despite limited computational capacity. The findings contribute to the understanding of the mathematical foundations of modern activity recognition systems.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Human Activity Recognition	2
1.1.1	Forschungsfrage und Zielsetzung	3
1.2	Gesundheitsüberwachung und Fitnesstracking durch Smartwatches	4
1.2.1	Fourieranalyse als mathematisches Werkzeug	5
1.3	Aufbau der Arbeit	5
2	Erfassung und Vorverarbeitung der Rohdaten	7
2.1	Sensoren	7
2.2	Datenerfassung	8
2.2.1	Analoge Signale	8
2.2.2	Digitale Signale	10
2.3	Datenvorverarbeitung	12
2.3.1	Filterung	13
2.3.2	Normalisierung der Daten	14
2.4	Segmentierung	15
2.4.1	Die gleitenden Fenster	15
2.4.2	Optimale Fensterlänge und Abtastrate für das Accelerometer in Sportuhren	16
2.5	Zusammenfassung	17
3	Theoretische Grundlagen der Fouriertransformation in der Signalverarbeitung	19
3.1	Grundprinzipien der Signalanalyse: Zeit- und Frequenzbereich	20
3.1.1	Zeitbereich	20
3.1.2	Frequenzbereich	20
3.2	Die Fourier-Reihe	21
3.2.1	Eindeutigkeit der Fourier-Reihe	23
3.2.2	Die Symmetrieeigenschaften der Basisfunktionen	25
3.2.3	Die komplexe Fourierreihe	27
3.2.4	Die reellen und komplexen Fourier-Koeffizienten	29
3.3	Die Fourier-Transformationen	31
3.3.1	Die kontinuierliche Fourier-Transformation	32
3.3.2	Die inverse Fourier-Transformation	33
3.4	Die Diskrete Fourier-Transformation	33
3.4.1	Fouriertransformation eines diskreten Signals	34
3.4.2	Beschränkte Signale	35

3.4.3	Definition der Diskrete Fourier-Transformation	37
3.5	Fast Fourier Transform	38
3.5.1	Die Recheneffizienz der FFT	39
3.6	Short Term Fourier Transformation	40
3.7	Zusammenfassung	42
4	Die Feature Extraktion	43
4.1	Merkmale der Zeit- vs. Frequenzdomäne	43
4.2	Feature-Extraktion in der Zeitdomäne	44
4.2.1	Grundlegende Statistische Merkmale	45
4.3	Feature-Extraktion in der Frequenzdomäne	47
4.3.1	Merkmale aus der Frequenzdomäne	47
4.3.2	Die wichtigsten Frequenzmerkmale	48
4.4	Feature Wahl - Dimensionsreduzierung	50
4.5	Zusammenfassung	51
5	Klassifizierung von Bewegungen	53
5.1	Grundlagen zur maschinellen Erkennung menschlicher Aktivitäten	53
5.1.1	Lernansätze	54
5.2	Klassifikation - Algorithmen	55
5.2.1	Support Vector Maschine	56
5.2.2	K-Nearest Neighbors	57
5.2.3	Random Forest	57
5.2.4	Markov Chains and Hidden Markov Models	58
5.2.5	Neurale Netzwerke und Deep Learning Achitektur	58
5.3	Zusammenfassung	61
6	Der Weg der Rohdaten bis zur Klassifizierung - Am Beispiel Laufen	63
6.1	Methodik	63
6.1.1	Training eines Klassifikators	64
6.1.2	Limitationen	68
6.2	Klassifikation von Aktivität <i>Laufen</i> - Ein Beispiel	68
6.2.1	Rohdatenimport	68
6.2.2	Vorverarbeitung	69
6.2.3	Feature-Extraktion	70
6.2.4	Klassifizierung	73
6.2.5	Kadenz und weitere Laufanalysen	75
6.3	Zusammenfassung	79
7	Fazit	81
7.1	Erkenntnisse zur Signalverarbeitung	81
7.2	Spezifische Forschungsergebnisse zur Fourier-Transformation	82
7.2.1	Recheneffizienz	82
7.2.2	Energieeffizienz	82

7.2.3	Erkennungsgenauigkeit	82
7.3	Limitationen und Ausblick	83

Abbildungsverzeichnis

1.1	Nutzung von Wearables in Europa	1
1.2	Human Activity Recognition	3
2.1	Datenvorverarbeitung bei Sportuhren	7
2.2	Funktionsgraph der Beschleunigung beim Gehen	9
2.3	Superposition von 3 Signalen	10
2.4	Sampling von 3 Signalen mit Abtastrate von 1 Hz	11
3.1	Transformation von Zeit- in Frequenzbereich	32
3.2	Fouriertransformation	32
3.3	Leistungsspektrogramm	41
3.4	Anwendung von Fensterfunktionen auf ein Signal und die entsprechenden Fouriertransformationen	42
4.1	Datenverarbeitungsprozess	44
5.1	Klassifizierung	55
5.2	HMM Architektur	58
5.3	Deep Learning Architektur	60
6.1	Garmin Sportmodusauswahl	63
6.2	Platzierung des Sensors	64
6.3	Links: Gehen, rechts: Laufen	69
6.4	Klassifizierungsausgabe in MatLab	74
6.5	Klassifizierung durch trainierten RF	75
6.6	Schrittfrequenz: Garmin(oben) und MatLab(unten)	76

1 Einleitung

Der rasante technologische Fortschritt des letzten Jahrzehnts erhöht nicht nur die Effizienz in der Arbeitswelt, sondern ermöglicht auch eine präzise Quantifizierung im Sportbereich. Diese Weiterentwicklung der modernen Technologien ermöglicht das Einsetzen von Aktivitätserkennung mit Hilfe von Wearables, um menschliche Alltagsaktivitäten zu überwachen, zu analysieren und ungewöhnliche Veränderungen in ihrer täglichen Routine zu identifizieren [6]. Wearables, am Körper tragbare Computersysteme, führen mittlerweile eine Vielzahl von Funktionen aus, darunter die Erfassung von Vitaldaten wie Herzfrequenz und Blutdruck, Aktivitätsverfolgung durch Schrittzählung und Bewegungsanalyse, GPS-basierte Standortverfolgung, Schlafmonitoring, Stresslevelmessung sowie die Bereitstellung von Echtzeit-Feedback und personalisierten Trainingsempfehlungen. Smart-Accessoires wie Smartwatches und Fitnesstracker erfahren daher eine stetig wachsende Verbreitung [4].



Abbildung 1.1: Nutzung von Wearables in Europa
Quelle: [4]

Aktuelle Statistiken belegen diesen Trend eindrucksvoll: Etwa 30 % der Europäer verwenden bereits Wearables. In Österreich stieg der Anteil der 18 - bis 64 - Jährigen, die diese Technologie aktiv nutzen, von 20 % im Jahr 2021 auf 31 % im Jahr 2023. Laut der deutschen Studie von Bitkom *Die Zukunft der Consumer Technology 2024* [15] nutzen bereits 36 % der Deutschen ab 16 Jahren eine Smartwatch. Die COVID-19-Pandemie beschleunigte diese Entwicklung, da während der Lockdowns das Eigenmonitoring der Gesundheit einen höheren Stellenwert erlangte. Diese dynamische Verbreitung spiegelt sich auch im steigenden Markturnsatz von Wearables wider [12].

1.1 Human Activity Recognition

Die automatische Erkennung menschlicher Aktivitäten (Human Activity Recognition, HAR) stellt ein zentrales Forschungsgebiet dar, in dem Wearables eine bedeutende Rolle einnehmen. Wie die Statistiken zeigen, findet die Aktivitätserkennung vielfältige Anwendungsbereiche in unserer modernen Gesellschaft – weit über den Sportkontext hinaus [23, 6, 26]. Im medizinischen Bereich unterstützen HAR – Geräte das Gesundheitspersonal bei der Patientenüberwachung durch Früherkennung abnormaler Verhaltensmuster und Sturzerkennung bei älteren Menschen. Dadurch wird die Qualität der medizinischen Betreuung verbessert und die Selbstständigkeit der Patienten gefördert. In Smart-Home-Umgebungen optimiert die Aktivitätserkennung Komfort und Energieeffizienz durch nutzeradaptive Steuerung von Heizung, Beleuchtung und anderen Systemen. Auch Sicherheitssysteme profitieren von HAR zur Identifikation ungewöhnlicher Verhaltensmuster und potenzieller Gefahrensituationen.

Der aktuelle Forschungsfokus im Bereich HAR richtet sich auf drei zentrale Aspekte [26]: die Entwicklung präziserer Erkennungsalgorithmen, effizientere Datenverarbeitung und die Optimierung von Sensortechnologien. Besondere Herausforderungen stellen dabei die Extraktion relevanter Merkmale aus Sensordaten, die Anwendung geeigneter Klassifikationstechniken und die mathematisch fundierte Modellierung menschlicher Bewegung dar. Durch die zunehmende Verbreitung von Wearables und Smartphones mit integrierten Sensoren wächst der Bedarf an ressourceneffizienten HAR-Methoden mit hoher Präzision [26].

Obwohl Bewegungserkennung bereits ein gut erforschtes Thema darstellt und qualitativ hochwertige Systeme auf dem Markt verfügbar sind [23], besteht weiterhin erhebliches Verbesserungspotenzial. Insbesondere mobile Geräte mit Beschleunigungssensoren erweisen sich als vielversprechend, da ihr geringer Energieverbrauch eine ganztägige Datenerfassung und -verarbeitung ermöglicht [2]. Aktuelle Forschungen konzentrieren sich verstärkt auf die Anwendung verschiedener komplexerer Maschinenlernansätze wie neuronale Netze, um die Erkennungsfähigkeit zu optimieren und komplexere Aktivitätsmuster zu identifizieren [6].

In der HAR-Forschung wird grundsätzlich zwischen zwei methodischen Ansätzen unterschieden: der sensorbasierten und der sichtbasierten Aktivitätserkennung [23]. Die sensorbasierte Variante nutzt Daten von am Körper getragenen Sensoren wie Beschleunigungsmessern, Gyroskopen oder Magnetometern, die in Smartphones oder tragbaren Geräten integriert sind. Diese Sensordaten durchlaufen einen mehrstufigen Verarbeitungsprozess [23]: zunächst die Schritte der Signalvorverarbeitung und der Merkmalsextraktion in Zeit- und Frequenzdomänen. Anschließend werden mit Hilfe von Dimensionsreduktionsverfahren wie der *Principal Component Analysis* (PCA) die relevantesten Merkmale identifiziert, bevor trainierte Maschinenlernalgorithmen (ML-Algorithmen) wie Entscheidungsbäume, Support-Vektor-Maschinen oder neuronale Netze die Aktivitäten klassifizieren.

Sichtbasierte Systeme hingegen arbeiten mit Video- und Bildmaterial aus Kameras. Sie

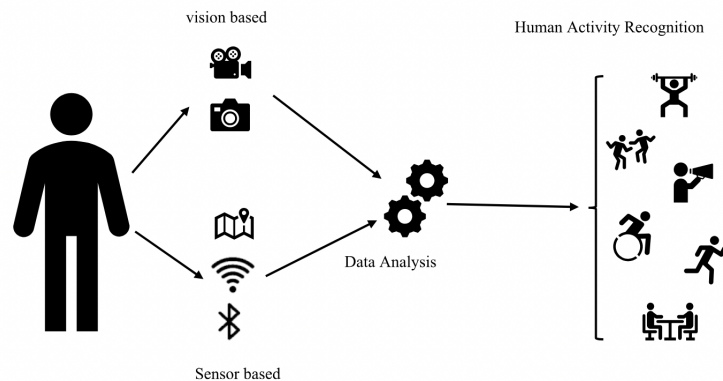


Abbildung 1.2: Human Activity Recognition
Quelle: [23]

analysieren visuelle Informationen durch Silhouettenerkennung, Bewegungsverfolgung oder modellbasierte Ansätze, die anatomische Proportionen und Gelenkbewegungen des menschlichen Körpers modellieren [23]. Diese Systeme können sowohl über einfache Bewegungserkennung als auch über komplexe 3D-Skelett-Rekonstruktionen arbeiten.

Die vorliegende Arbeit fokussiert sich ausschließlich auf die sensorbasierte Aktivitätserkennung, um die mathematischen Grundlagen der Funktionsweise von Sportuhren systematisch zu untersuchen.

1.1.1 Forschungsfrage und Zielsetzung

Obwohl Sportuhren und ihre Aktivitätserkennung sich hoher Beliebtheit erfreuen, gibt es keine klare Aufarbeitung der einzelnen Schritte bei der Analyse von Bewegungsdaten in Sportuhren und deren mathematischer Fundierung. Diese Arbeit adressiert diese Lücke und untersucht systematisch, wie die Fourier-Transformation als mathematisches Werkzeug zur Analyse periodischer Bewegungsmuster in Sportuhren eingesetzt werden kann.

Die zentralen Forschungsfragen lauten:

1. Welche spezifischen mathematischen Transformationen und Algorithmen sind erforderlich, um die Rohdaten der Beschleunigungssensoren in Sportuhren effizient zu verarbeiten und in aussagekräftige Trainingsanalysen zu überführen?
2. Wie wirken sich die spezifischen Eigenschaften der Fourier-Transformation auf Rechenaufwand, Energieverbrauch und Erkennungsgenauigkeit bei der Aktivitätserkennung in Sportuhren aus?

Ziel dieser Arbeit ist es, eine mathematisch fundierte Analyse des gesamten Signalverarbeitungsprozesses von der Rohsignalerfassung bis zur Aktivitätsklassifikation zu liefern,

1 Einleitung

wobei der Schwerpunkt auf der Rolle der Fourier-Transformation für die Analyse von Bewegungsdaten liegt. Im Gegensatz zu bestehenden Forschungsarbeiten, die häufig entweder die verfahrenstechnischen Aspekte oder die Klassifikationsalgorithmen im Bereich des ML betonen, bietet diese Arbeit eine umfassende mathematische Perspektive auf den gesamten Verarbeitungsprozess der Beschleunigungsdaten.

1.2 Gesundheitsüberwachung und Fitnesstracking durch Smartwatches

Insbesondere unter Sporttreibenden haben HAR-Systeme in Form von Smartwatches als Sportuhren besondere Bedeutung erlangt. Spezialisierte Modelle sind darauf ausgelegt, Alltagsaktivitäten zu identifizieren und bis zu 20 verschiedene Sportarten aufzuzeichnen und zu analysieren. Diese Geräte werden primär zur präzisen Erfassung und Steuerung sportlicher Aktivitäten genutzt und dienen so der Gesundheitsförderung. Die meisten Sportuhren verfügen über die Grundfunktionalität, Bewegungen wie Laufen, Radfahren und Gehen zu registrieren und zu analysieren. Für diese Funktionalität werden in modernen Sportuhren verschiedene Sensoren integriert: Beschleunigungssensoren, GPS-Module, Gyroskope sowie optische PPG-Sensoren und weitere [23]. Diese liefern Rohdaten zur Bestimmung von Parametern wie Herzfrequenz, zurückgelegter Strecke, Geschwindigkeit, Schrittzahl und Kalorienverbrauch. Sportuhren fungieren dadurch als Instrument der Trainingsoptimierung, der Fortschrittsmessung sowie der Analyse von Leistungs- und Gesundheitsstatus. Sie unterstützen zudem bei der Belastungssteuerung und Regenerationskontrolle, was Übertraining verhindert und die Trainingseffizienz steigert [31]. Entscheidend dabei ist, dass die Rohdaten der Sensoren allein noch keine hilfreichen Bewegungsanalysen ermöglichen. Erst die komplexe mathematische Verarbeitung durch effiziente Algorithmen erlaubt das präzise Klassifizieren und Analysieren der Bewegungen [29].

Die Erkennung und Analyse von Bewegungen mit diesen Sensoren stellt jedoch eine anspruchsvolle Aufgabe dar, die durch diverse Faktoren erschwert wird [23, 6]. Die natürliche Variabilität menschlicher Bewegungsmuster – jeder Mensch führt Aktivitäten auf individuell unterschiedliche Weise aus – erschwert die Entwicklung universeller Erkennungsalgorithmen. Zudem sind die von den Sensoren gelieferten Daten häufig mit Rauschen behaftet und können durch externe Faktoren verfälscht werden. Eine besondere Herausforderung liegt in der Abgrenzung ähnlicher Aktivitäten wie Gehen und langsames Joggen, die vergleichbare Beschleunigungsmuster erzeugen. Hinzu kommt die technische Anforderung, dass Sportuhren umfangreiche Berechnungen in Echtzeit durchführen müssen, während sie gleichzeitig energieeffizient arbeiten sollen – ein Dilemma, das die Komplexität der einsetzbaren Algorithmen und damit die Präzision der Bewegungserkennung begrenzt.

1.2.1 Fourieranalyse als mathematisches Werkzeug

Bei der Bewältigung dieser technischen Herausforderungen nimmt die Fourier-Analyse eine Schlüsselrolle ein. Sie ermöglicht die Transformation von Zeitreihendaten in den Frequenzbereich und erlaubt damit eine effiziente Verarbeitung und Interpretation der Sensordaten in Sportuhren [26]. Die resultierenden Frequenzspektren bieten distinktive Muster, die eine präzise Erkennung verschiedener Aktivitäten ermöglichen. So weisen periodische Bewegungen wie Gehen, Laufen und Rennen charakteristische Frequenzen von 0-2 Hz, 2-2,8 Hz sowie über 2,8 Hz auf, wodurch sie eindeutig klassifiziert werden können [18].

In anderen Worten, liegt ein entscheidender Vorteil der Fourier-Analyse in ihrer Fähigkeit, komplexe Zeitreihendaten auf wenige aussagekräftige Frequenzkomponenten zu reduzieren. Dies vereinfacht die Bewegungserkennung erheblich und verringert den Speicherbedarf – ein kritischer Faktor bei Sportuhren mit begrenzten Ressourcen [26]. Zudem ermöglicht die Fourier-Transformation das Filtern hoher Frequenzen und damit die Eliminierung von Rausch- und Störsignalen im Datensatz.

Die effiziente Implementierung der Fourier-Transformation durch den Fast Fourier Transform (FFT) Algorithmus reduziert den Rechenaufwand drastisch und ermöglicht eine Echtzeitverarbeitung bei gleichzeitig geringem Energieverbrauch [7]. Diese Kombination aus mathematischer Eleganz und praktischer Effizienz macht die Fourier-Analyse zu einem relevanten Werkzeug für die präzise und ressourcenschonende Aktivitätserkennung in Sportuhren.

1.3 Aufbau der Arbeit

In diesem Sinne beschäftigt sich die vorliegende Masterarbeit mit der mathematischen Grundlage und Anwendung von Methoden zur Human Activity Recognition (HAR) im Kontext von Sportuhren. Der Fokus liegt dabei auf der systematischen Untersuchung des gesamten Prozesses von der Rohsignalerfassung bis zur Aktivitätsklassifikation, wobei die Fourier-Transformation als besonders effizientes Analyseinstrument für periodische Bewegungssignale herausgearbeitet wird. Jedes Kapitel trägt gezielt zur Beantwortung der Forschungsfrage bei. Die Arbeit gliedert sich in folgende Kapitel:

In Kapitel 2 wird der Prozess der Datenerfassung sowie deren Vorverarbeitung durch Segmentierung, Normalisierung und Filterung behandelt. Kapitel 3 widmet sich ausführlich den mathematischen Grundlagen der Fourier-Transformation und ihren verschiedenen Formen. Hier wird näher auf die besondere Eignung der Fourieranalyse durch ihren geringen Rechenaufwand für Sportuhren und für die Analyse einfacher Bewegungsmuster eingegangen. Dieses theoretische Fundament bildet die Basis für das nachfolgende Kapitel.

1 Einleitung

In Kapitel 4 wird die Extraktion von Merkmalen aus den Daten sowohl in der Frequenz- als auch in der Zeitdomäne untersucht. Die hier gewonnenen Erkenntnisse stellen die Grundlage für die anschließende Klassifizierung dar, indem sie die charakteristischen Eigenschaften verschiedener Bewegungsmuster mathematisch erfassbar machen.

Kapitel 5 bietet einen kurzen Überblick über den abschließenden Schritt der Aktivitätserkennung: die Klassifizierung durch ML-Algorithmen. Dieser kompakte Abschnitt erläutert kurz die Grundlagen von maschinellem Lernen (ML) und stellt jene Algorithmen vor, die bei Aktivitätserkennungen eingesetzt werden.

In Kapitel 6 wird eine praktische Implementierung in MATLAB vorgestellt, die zur Darstellung und zum tieferen Verständnis der mathematischen Vorgänge dient. Die theoretischen Konzepte sollen anschaulich illustriert und die einzelnen Schritte der Signalverarbeitung bei der Aktivitätserkennung transparent gemacht werden.

Den Abschluss der Arbeit bildet eine Zusammenfassung der Ergebnisse, eine Reflexion über die Limitationen des vorgestellten Ansatzes sowie einen kurzen Ausblick auf zukünftige Forschungsperspektiven im Bereich der sensorbasierten Aktivitätserkennung in Sportuhren.

2 Erfassung und Vorverarbeitung der Rohdaten

Die Aufnahme und Vorverarbeitung von Beschleunigungsdaten bilden den Anfang der Bewegungserkennung einer Sportuhr. In diesem Kapitel wird beschrieben, wie die aufgenommenen Rohdaten verarbeitet werden, sodass anschließend Merkmale zur Aktivitätserkennung extrahiert werden können.

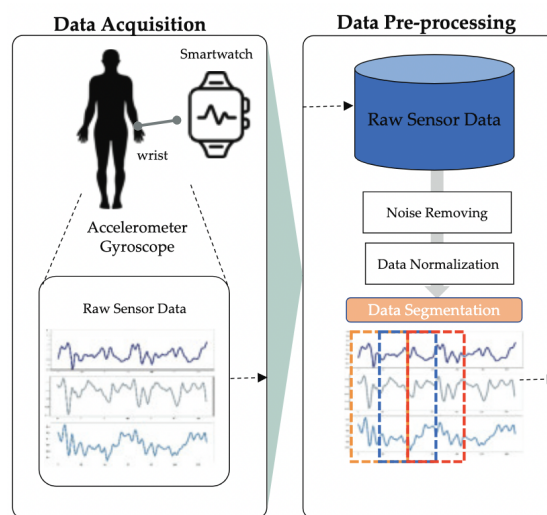


Abbildung 2.1: Datenvorverarbeitung bei Sportuhren

Quelle: [19]

2.1 Sensoren

In jeder modernen Sportuhr befinden sich ein Accelerometer, ein GPS-Empfänger und ein Gyroskop. Diese Systeme nehmen durchgehend Daten über die Handbewegung des Tragenden auf [5]. Der triaxiale Beschleunigungssensor zeichnet die Beschleunigung in drei Achsen (x,y,z) mit einer Abtastrate von 10 - 200 Hz auf [26, 24]. Sowohl die Erdbeschleunigung von ca. 9.81 m/s^2 als auch die bewegungsindizierte Beschleunigung werden gemessen. Das Gyroskop hingegen erfasst die Winkelgeschwindigkeit (in $^\circ/\text{s}$ oder rad/s) in drei Achsen und somit die Drehbewegungen des Arms [23]. Diese Daten ergänzen

die Beschleunigungsdaten und dienen einer präziseren Aktivitätserkennung, speziell bei komplexeren (Dreh-)Bewegungen. Modernere, teurere Geräte haben auch ein Magnetometer eingebaut. Das Magnetometer liefert Orientierungsdaten durch Erdmagnetfeldmessung und unterstützt in Kombination mit anderen Sensoren, durch die Bestimmung der Himmelsrichtungen, die Genauigkeit der Bewegungsdaten. Viele Geräte haben auch PPG-Sensoren, optische Sensoren, die mit Hilfe einer emittierenden Lichtquelle und einem Rezeptor die Herzaktivität messen [13]. Wenn das Herz Blut in das Gewebe pumpt, verringert sich die Menge des detektierten Lichts. Die Veränderungen in der Menge des erfassten Lichts werden zur Schätzung der Herzfrequenz herangezogen. Hier handelt es sich um eine Funktion, die aktiv von Benutzenden ein- und ausgeschaltet werden kann. Die Herzfrequenzmessung wird in dieser Arbeit nicht näher beleuchtet, da ihre Genauigkeit derzeit noch stark von dem Uhrenmodell und der Intensität der Aktivität abhängt [13]. Außerdem dient die Herzfrequenz nicht als primäre Datenquelle für die Bewegungserkennung, sondern als ergänzender Parameter für die Bewertung der Qualität und Intensität der sportlichen Betätigung. Bei der sensorbasierten Aktivitätserkennung spielen die Beschleunigungsdaten die wichtigste Rolle [26]. Die Werte von den anderen Sensoren dienen zur Verfeinerung und Verbesserung der Bewegungsanalyse. Aus diesem Grund fokussiert sich die Arbeit auf die Verarbeitung der Daten des Accelerometers.

2.2 Datenerfassung

In der Signalverarbeitung unterscheidet man zwischen analogen und digitalen Signalen.

2.2.1 Analoge Signale

Kontinuierliche Datenreihen wie menschliche Bewegungen kommen in der Natur vor und werden analog genannt. Diese Form der Signale kann als Funktion $f : \mathbb{R} \rightarrow \mathbb{R}$ dargestellt werden, wobei $f(t) \in \mathbb{R}$ und $t \in \mathbb{R}$ gilt [20]. Modelliert man menschliche Bewegung mit Hilfe von Beschleunigung, so wird jedem Zeitwert ein Beschleunigungswert zugeordnet, wodurch die kleinsten Veränderungen festgestellt werden können. Funktionen mit einer kontinuierlichen Zeitachse ($f(t) \in \mathbb{R}$) nennt man zeitkontinuierliche Signale. Weiters handelt es sich um ein periodisches Signal $f : \mathbb{R} \rightarrow \mathbb{R}$ mit $\lambda \in \mathbb{R}_{>0}$, falls für alle $t \in \mathbb{R}$, $f(t) = f(t + \lambda)$ gilt [20].

Zyklische Bewegungen wie Gehen, Laufen und Radfahren erzeugen periodische Beschleunigungsmuster, welche sich durch Sinusfunktionen darstellen lassen [18]. Mathematisch werden diese Bewegungsformen durch

$$f(t) := A \sin(2\pi(\omega t - \varphi)), t \in \mathbb{R}$$

dargestellt [20]. Die Parameter A , ω und φ beschreiben Amplitude, Frequenz mit $\lambda = 1/\omega$ und Phase der Funktion [20]. Der Abstand zwischen 2 Höhepunkten wird durch $\lambda = 1/\omega$

beschrieben [18]. Umgelegt auf die physikalische Bedeutung für das Laufen, repräsentiert die Amplitude die Intensität der Bewegung und die Frequenz die Schrittzahl. Die physikalische Bedeutung der Phase einer Sinusfunktion beim Laufen besteht darin, dass sie den exakten zeitlichen Verlauf der Beschleunigungsereignisse wie Bodenkontakt, Abstoßung und Flugphase im Bewegungszyklus festlegt.

In der Signal- und Fourieranalyse wird vorwiegend mit der normalisierten Gleichung

$$\cos_{\omega,\varphi}(t) := \sqrt{2} \cos(2\pi(\omega t - \varphi)) \quad (2.1)$$

gearbeitet. Die Leistung (durchschnittliche Energie) der Schwingung wird durch den Normalisierungsfaktor $A = \sqrt{2}$ standardisiert.

Die Abbildung 2.2 präsentiert sowohl den Geschwindigkeits- als auch den Beschleunigungsgraphen, der eine Gehbewegung beschreibt. Sinuskurven ermöglichen die Beschrei-

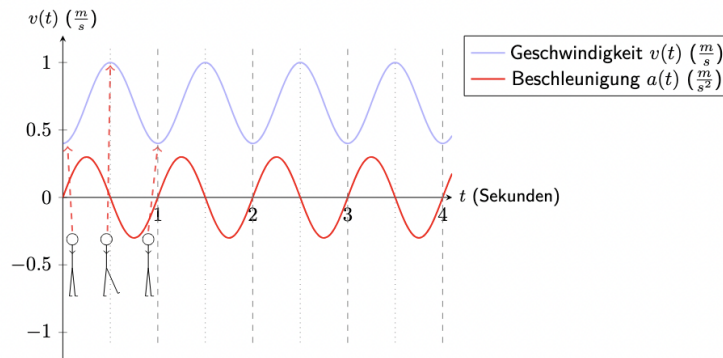


Abbildung 2.2: Funktionsgraph der Beschleunigung beim Gehen

Quelle: [18]

bung der Beschleunigung beim Gehen als auch beim Laufen, da diese Aktivitäten rhythmisch verlaufen. Wie auf der Grafik zu erkennen ist, erhöht sich beim Start eines Schrittes, der durch das Heben und das nach vorne Schwingen eines Beines eingeleitet wird, die Beschleunigung [27]. Während der zweiten Hälfte des Schrittes verringert sich die Beschleunigung, bis der Fuß aufsetzt [27]. Zweischrittzyklen bilden einen Gangzyklus.

Der Bewegungsablauf beim Laufen unterscheidet sich deutlich von dem des Gehens. Der Bodenkontakt, die Standphase, ist beim Laufen deutlich verkürzt, wodurch die Schwungphase sich verlängert. Außerdem kommt die Flugphase zwischen Stand- und Schwungphase hinzu, in der beide Füße sich in der Luft befinden [27]. Diese Änderungen spiegeln sich in dem Grafen der Sinusfunktion, die die Laufbewegung beschreibt, wider. Im Vergleich zum Gehen weist die Beschleunigungskurve signifikant größere Amplituden auf. Durch das stärkere Anheben des Körpers und den intensiveren Aufprall erreichen die Werte der vertikalen Beschleunigung das Zwei- oder Dreifache beim Laufen. Weiters steigt die Frequenz, da die Schrittfrequenz beim Laufen deutlich höher ausfällt. Zusammengefasst zeichnen sich Sinusfunktionen, die Laufen repräsentieren, durch größere Unterschiede

2 Erfassung und Vorverarbeitung der Rohdaten

zwischen Maxima und Minima auf als auch durch die stärkeren Steigungen und Abfälle der Beschleunigungskurve.

Diese Abbildung 2.2 stellt einen perfekt modellierten Grafen dar. Die vom Beschleunigungssensor aufgenommenen Daten zeigen jedoch keine makellose Sinus- oder Kosinusfunktionen. Das Accelerometer der Sportuhr zeichnet nämlich nicht nur die Haupt-, sondern auch alle parallel laufenden Bewegungen, wie zum Beispiel die Fuß und Armbewegungen, auf [18]. Das bedeutet, dass es sich bei dem aufgezeichneten Signal um eine Superposition, eine Überlagerung mehrerer Signale, handelt.

Mathematisch wird die Superposition von drei Signalen f , g und h mit

$$(f + g + h)(t) = f(t) + g(t) + h(t)$$

beschrieben. Die Grafik 2.3 illustriert, wie die Überlagerung von 3 trigonometrischen Funktionen aussehen kann.

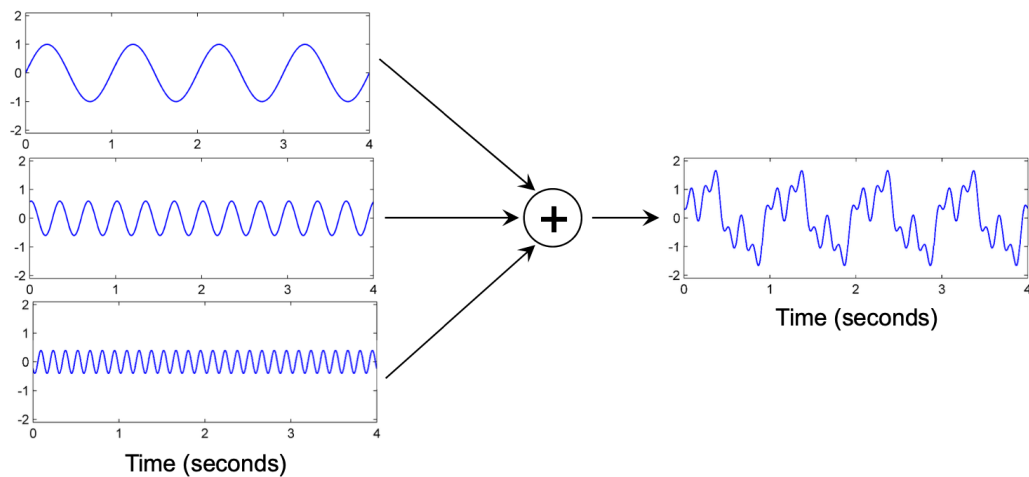


Abbildung 2.3: Superposition von 3 Signalen
Quelle: [20]

2.2.2 Digitale Signale

Computer und Mini-Computer wie Sportuhren arbeiten mit endlichem Speicher und diskreten Zuständen, wodurch sie nur eine begrenzte, wenn auch sehr große Anzahl unterschiedlicher Zahlenwerte verarbeiten können. Analoge Signale hingegen werden mathematisch durch kontinuierliche Funktionen repräsentiert, deren Definitions- und Wertebereiche überabzählbar unendlich viele Punkte enthalten, wodurch zwischen je zwei beliebig nah beieinanderliegenden Werten stets weitere Werte existieren [20]. Um die

Daten mit Hilfe von Computern zu verarbeiten, muss die kontinuierliche Wellenform in eine diskrete Repräsentation des Signals, in ein **digitales Signal** umgewandelt werden. Durch Abtastung (Sampling) und Quantisierung wird die unendliche Anzahl der Zahlen zu einer endlichen Anzahl von Werten reduziert.

Sampling

Bei der Signalverarbeitung beschreibt der Begriff *Sampling* die Umwandlung eines zeit-kontinuierlichen Signals (CT) in ein zeit-diskretes Signal (DT), laut Müller [20]. Bei diesem Vorgang wird das kontinuierliche Signal der menschlichen Bewegung nur an festgelegten Zeitpunkten durch den Beschleunigungssensor erfasst. Meist erfolgt eine äquidistante Abtastung, bei der die Zeitwerte in regelmäßigen Intervallen entnommen werden. Bei diesem Transformationsprozess wird $f : \mathbb{R} \rightarrow \mathbb{R}$ zu einer diskreten Abbildung $x : \mathbb{Z} \rightarrow \mathbb{R}$ mit $\mathbb{Z} \subset \mathbb{R}$ konvertiert. Die Konvertierung erfolgt mittels der Formel

$$x(n) := f(n \cdot T)$$

für $n \in \mathbb{Z}$, wobei T die Sampling-Periode darstellt und n ganzzahlige Werte aufweist [20].

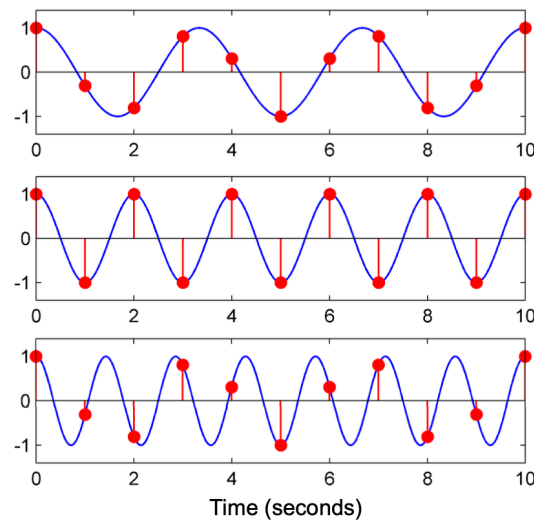


Abbildung 2.4: Sampling von 3 Signalen mit Abtastrate von 1 Hz
Quelle: [20]

Der Kehrwert $f_s = 1/T$ wird als Abtastfrequenz bezeichnet und gibt die Anzahl der Messungen je Zeiteinheit an (in Hz). Nach Müller [20] handelt es sich grundsätzlich beim Sampling um einen informationsreduzierenden Prozess, nachdem Daten zwischen den Messpunkten verloren gehen. Doch laut Nyquist-Shannon-Theorem ist unter spezifischen

Voraussetzungen bezüglich des Frequenzspektrums eine vollständige Wiederherstellung des Ursprungssignals realisierbar [20]. Bei zu niedriger Abtastrate werden komplexere Funktionen unzureichend beschrieben und Frequenzbestandteile des Signals sind nicht mehr differenzierbar, wodurch das zugehörige Signal nicht mehr exakt rekonstruierbar ist. Dieses Phänomen wird Aliasing genannt [20].

Im Abschnitt 2.4 werden die Wichtigkeit und die Schwierigkeit der Wahl von Abtastrate und Fensterlänge für die Bewegungsanalyse bearbeitet.

Quantisierung

Sampling ist der Prozess, der die kontinuierliche Zeitachse in eine Menge von diskreten Werten umgewandelt hat. Quantisierung beschreibt hingegen den zweiten Schritt von der Wandlung des analogen zum digitalen Signal, die Transformation des unendlichen Wertebereichs (\mathbb{R}) in einen endlichen $\Gamma \subset \mathbb{R}$. In anderen Worten werden die kontinuierlichen Werte des Amplitudenbereichs auf eine finite Anzahl diskreter Werte abgebildet. Dies geschieht, indem der gesamte Amplitudenbereich in gleichmäßige Intervalle eingeteilt wird, wobei jedes Intervall durch einen spezifischen digitalen Wert repräsentiert wird.

Formal kann die Quantisierung als eine Funktion Q verstanden werden, die jeden kontinuierlichen Wert $x(n)$ des abgetasteten Signals auf einen diskreten Wert $Q(x(n))$ abbildet [20]:

$$Q : \mathbb{R} \rightarrow \{q_1, q_2, \dots, q_m\}$$

Die Menge der möglichen Quantisierungsstufen ist $\{q_1, q_2, \dots, q_m\}$, wobei m endlich ist.

Durch die Rundung der kontinuierlichen Werte auf diskrete Stufen entsteht der sogenannte Quantifizierungsfehler. In Sportuhren ist die Quantisierung besonders wichtig für die Umsetzung präziser Bewegungsanalysen. Die Bittiefe des Analog-Digital-Wandlers bestimmt, wie fein die gemessenen Beschleunigungswerte aufgelöst werden können. Typischerweise verwenden moderne Sportuhren Analog-Digital-Umwandler (ADCs) mit 12 bis 16 Bit, was 4.096 bis 65.536 verschiedene Quantisierungsstufen ermöglicht [24]. Je besser die Quantisierungsauflösung, desto subtiler ist die Erkennung der Bewegungsunterschiede. [20]

2.3 Datenvorverarbeitung

Bei der Bewegungsanalyse werden die Rohdaten zunächst von Rauschen befreit (gefiltert) und normalisiert. Anschließend erfolgt die Segmentierung des kontinuierlichen Datenstroms mithilfe von Sliding Windows. Für die Echtzeitverarbeitung ist diese Segmentierung der erste Schritt, um den Datenstrom abschnittsweise analysieren zu können. Dabei wird jedes Fenster einzeln vorverarbeitet, indem Filterung und Normalisierung auf die segmentierten Daten jedes Fensters angewandt werden.

2.3.1 Filterung

Die korrekte Interpretation der Beschleunigungsdaten für die Aktivitätserkennung wird erst durch Filter ermöglicht [2]. Einerseits dient die Filterung zur Trennung der Gravitationskomponente und der tatsächlichen Bewegungsbeschleunigung [24]. Andererseits setzt man Filter zur Rauschunterdrückung oder zum Herausheben von Frequenzkomponenten ein [2]. Die verschiedenen Filter werden in diesem Abschnitt vorgestellt.

Rauschunterdrückung

Zur Reduzierung unerwünschter Störsignale in den Sensordaten von Sportuhren werden primär zwei komplementäre Filtertypen implementiert, der Medianfilter und der Butterworth-Tiefpassfilter [2, 22]. Der **Medianfilter** dient zur Entfernung von Ausreißern, ohne dabei die Kantencharakteristik der Signale zu beeinträchtigen, die durch plötzliche Beschleunigungsänderungen entstehen. Er fungiert als einer der ersten Vorverarbeitungsschritte nach der Rohdatenerfassung vom Beschleunigungssensor. Außerdem wird zum Beispiel ein **Butterworth-Tiefpassfilter** (3. Ordnung mit 20 Hz Grenzfrequenz) zur Reduzierung von hochfrequentem Rauschen angewandt. Nachdem 99% der Beschleunigung menschlicher Bewegungen unter 15 Hz liegt, wird die Grenze von 20 Hz als sinnvoll erachtet.

Trennung von Gravitationskomponente und Bewegungskomponente

Das Accelerometer nimmt die Summe aus Gravitationsbeschleunigung und Bewegungsbeschleunigung auf. Für die Extraktion der Gravitationskomponente wird ein weiterer Butterworth-Tiefpassfilter eingesetzt [29]. Als Grenzfrequenz fällt die Wahl auf 0.3 Hz von Xue [29], da angenommen wird, dass die Gravitationskraft als niederfrequent (unter 0.3 Hz) angenommen wird. Dieser Schritt dient zur Bestimmung der Orientierung im Raum, beispielsweise um die Ausrichtung des Handgelenks zu erkennen.

Außerdem wird ein Butterworth-Hochpassfilter mit Grenzfrequenz 0.1 – 0.3 Hz für die Isolierung der Bewegungskomponente eingesetzt. Hier ist das Ziel, die Gravitationskomponente zu eliminieren und ausschließlich die höherfrequenten Bewegungskomponenten für die Frequenzanalyse mithilfe der FFT zu behalten [24]. Die Anwendung des Hochpassfilters ist essentiell für die Korrektheit der Bewegungsanalyse. Bei diesem Filterschritt wird zusätzlich großteils die *Direct Current Komponente (DC)* herausgefiltert.

Elimination der Direct-Current Komponente

Die DC-Komponente ist definiert als der Mittelwert eines Signals und stellt die Null-Frequenz-Komponente bei der Fourier-Analyse dar [1]. Ein symmetrisches Signal, das gleiche Flächen oberhalb und unterhalb der Zeitachse aufweist (wie reine Sinuswellen),

2 Erfassung und Vorverarbeitung der Rohdaten

besitzt einen Mittelwert von null und damit keine DC-Komponente. Asymmetrische Signale hingegen weisen einen von null verschiedenen Mittelwert auf [1].

In Beschleunigungsdaten von Sportuhren entsteht die DC-Komponente hauptsächlich durch die Gravitationsbeschleunigung ($9.81m/s^2$), die das Signal asymmetrisch macht, und durch Sensor-Offset-Fehler [10]. Nachdem der Hochpassfilter den DC-Anteil aus den kontinuierlichen Sensordaten nicht vollständig entfernt, benötigt es zusätzlich eine segmentweise Mittelwertsubtraktion in jedem Analysefenster, um verbleibende lokale DC-Offsets vor der FFT-Analyse zu eliminieren [10]. Diese Kombination aus globaler Hochpassfilterung und lokaler DC-Korrektur gewährleistet, dass die spektralen Features für die Aktivitätsklassifizierung nicht durch dominante Null-Frequenz-Anteile verfälscht werden.

2.3.2 Normalisierung der Daten

Die Normalisierung der Bewegungsdaten bringt mehrere Vorteile mit sich. Einerseits ermöglicht sie die Vergleichbarkeit der erfassten Daten von verschiedenen verbauten Sensoren in einer Sportuhr, da unterschiedliche Sensorkalibrierungen und Größenordnungen auf einen gemeinsamen standardisierten Wertebereich gebracht werden. Rohdaten mit verschiedenen Skalierungen stellen dadurch beim weiteren Rechenprozess keine Hürde dar. Andererseits kann das Normalisieren der Daten die Genauigkeit der Klassifizierung von Aktivitäten verbessern. Viele Klassifizierungsalgorithmen (neuronale Netze, SVM, k-NN) arbeiten besser mit normalisierten Daten und erzielen eine höhere Genauigkeit [26]. Was die Beschleunigungsdaten betrifft, kann eine Normalisierung sowohl vor als auch nach der Feature-Extraktion erfolgen, je nach verwendeten Features und Algorithmenanforderungen, um wichtige Informationen zu erhalten.

Min-Max-Normalisierung

Die gängigste Methode bei Sportuhren ist die Normalisierung der Daten auf den Bereich $[0, 1]$ [26, 29]:

$$x_{norm} = \frac{x - \min(x)}{\max(x) - \min(x)} \quad (2.2)$$

Diese Variante wird häufig angewandt, wenn die nachfolgenden Algorithmen (besonders neuronale Netze) Eingabewerte in einem begrenzten Bereich erwarten, was auf die modernen, komplexeren Klassifikationsmodelle von Sportuhren zutrifft. Laut San-Segundo u. a. [22] verbessern einfache Normalisierungsschritte wie die Min-Max-Normalisierung aufgrund der großen Variabilität der Beschleunigungsaufnahmen (Arm- und Ganzkörperbewegungen) nur in Kombination mit fortgeschrittenen ML-Ansätze oder neuronalen Netzwerken die Genauigkeit der Klassifikation.

2.4 Segmentierung

Die Segmentierung spielt eine fundamentale Rolle in der Funktionsweise von Smartwatches. Bei diesem wichtigen Vorverarbeitungsschritt werden die von dem Beschleunigungssensor erfassten kontinuierlichen Daten der Zeitachse in kürzere Abschnitte, die auch *Fenster* oder *Frames* genannt werden, unterteilt [26]. Der Hauptzweck der Segmentierung liegt in der Aufbereitung der Rohdaten für maschinelles Lernen und Mustererkennungsalgorithmen, da kontinuierliche Sensordaten für die direkte Verarbeitung durch ML-Algorithmen meist zu komplex und unstrukturiert sind [3].

Die Segmentierung erfüllt folgende Zwecke [3, 14, 19]:

1. Stationarität: Innerhalb eines hinreichend kurzen Zeitfensters kann das nicht-stationäre Beschleunigungssignal als näherungsweise stationär betrachtet werden. Dadurch wird die Anwendung von Frequenzanalysetechniken wie der FFT erst sinnvoll (näheres im nächsten Kapitel).
2. Echtzeitverarbeitung: Durch die Segmentierung können Teildaten bereits, bevor der gesamte Datenstrom vorliegt, analysiert werden.
3. Adaptive Analyse: Die segmentweise Verarbeitung erlaubt die dynamische Anpassung der Algorithmen an sich verändernde Signaleigenschaften.
4. Ressourceneffizienz: Die Verarbeitung kleinerer Datenabschnitte reduziert den Speicher- und Rechenaufwand, was wichtig für Sportuhren ist.

Die Segmentierung ist somit ein zentraler Schritt für die Erkennung von menschlichen Aktivitäten wie Gehen, Laufen, Sprinten oder Stehen durch Sportuhren [3].

2.4.1 Die gleitenden Fenster

Bei den *Sliding Windows*, den *gleitenden Fenstern*, handelt es sich um die am weitesten verbreitete Segmentierungsform in der Aktivitätserkennung, da sie sich für sowohl periodische Bewegungen wie Laufen als auch statische wie Stehen als besonders akkurat erwiesen hat [3, 14, 19]. Der Grund dafür ist die Eignung für die Echtzeitanwendung durch die einfache Implementierung und den minimalen Vorverarbeitungsaufwand. Dabei werden Sensordaten in Segmente in kürzere Abschnitte unterteilt, sodass keine Lücken zwischen den einzelnen Fenstern entstehen [14].

Bei Sportuhren und Fitnesstrackern werden hauptsächlich zwei Arten *gleitenden Fenster* verwendet [19]:

1. Überlappende Fenster (Overlapping Temporal Window, OW): Bei dieser Methode überlappen sich aufeinanderfolgende Fenster um einen bestimmten Prozentsatz (häufig 50%). Bei einem 10-Sekunden-Fenster wäre das eine Überlappung von 5 Sekunden mit dem vorhergehenden Fenster.

2 Erfassung und Vorverarbeitung der Rohdaten

2. Nicht-überlappende Zeitfenster (Non-Overlapping Temporal Window, NOW): Bei dieser Methode gibt es keine Überlappung zwischen aufeinanderfolgenden Fenstern.

Die Untersuchungsergebnisse zeigen, dass überlappende Segmentierung allgemein genauere Ergebnisse lieferte [19]. Die überlappende Methode bietet eine konsistente Datenmenge für die Analyse, keine Informationen an den Übergängen gehen verloren und ist durch das kontinuierliche Arbeiten geeignet für die Echtzeit-Anwendung [14]. Weiters stehen den Klassifizierungsalgorithmen eine höhere Menge an Datenpunkten für die Überlappung zur Verfügung, was zu den besseren Ergebnissen beisteuern könnte [19]. Es sei erwähnt, dass der höhere Rechenaufwand durch die Überlappung der Fenster einen höheren Akkuverbrauch bei Sportuhren bedeutet. Allgemein lässt sich eine steigende Tendenz zur Verwendung von überlappenden Fenstern feststellen, da Baños u. a. [3] noch von ihrem seltenen Einsatz spricht und die Anzahl der Studien, die OW verwenden, seither gestiegen ist [26, 14, 19]. Die Hardware von Sportuhren hat sich stark weiterentwickelt und somit sind Rechenkapazität sowie Akkuleistung gestiegen, was auch rechenintensivere Vorgänge ermöglicht.

2.4.2 Optimale Fensterlänge und Abtastrate für das Accelerometer in Sportuhren

Die Wahl der Fensterlänge ist entscheidend für die Verarbeitung von Accelerometer-Daten von Human Activity Recognition (HAR)-Systemen in Sportuhren und hat direkte Auswirkungen auf die Erkennungsgenauigkeit, Reaktionszeit und Energieeffizienz. Laut Jaén-Vargas u. a. [14] und Mekruksavanich, Jantawong und Jitpattanakul [19] handelt es sich dabei um einen Kompromiss zwischen verschiedenen Faktoren: Kürzere Fenster erkennen schnelle Aktivitätswechsel besser. Sie sind für die Echtzeit-Analyse bevorzugt und weisen eine höhere Genauigkeit auf, verursachen jedoch höheren Rechenaufwand. Für komplexere Aktivitätsmuster scheinen längere Fenster, die den Rechenaufwand reduzieren, aber die Bearbeitungszeit verlängern, geeignet.

Die optimale Fensterlänge in Sportuhren, in die die Beschleunigungsdaten geteilt werden, ist abhängig vom Anwendungsfall [24]. In gängigen Sportuhren wird die Wahl der Fensterlängen an die zu erkennende Bewegung angepasst und beträgt meist zwischen 1 und 5 Sekunden. Bei Sportuhren hat man die Möglichkeit, Aktivitäten wie Laufen, Schwimmen und Radfahren auszuwählen, aufzuzeichnen und ein Echtzeit-Feedback zu erhalten. Bei der Auswahl dieser Option handelt es sich um Voreinstellungen.

Für die Erkennung von Geh- und Laufbewegungen hat sich eine Fensterlänge von etwa 2.5 Sekunden als effektiv erwiesen[24]. In dieser Zeit wurde zumindest ein Schrittzyklus durchgeführt. Grundsätzlich sollte bei periodischen Bewegungen mindestens ein Zyklus in einem Fenster liegen. Bei kurzen, schnellen Bewegungen wie Sprüngen oder Richtungswechseln werden kürzere Fenster von 1-2 Sekunden herangezogen, um die charakteristischen Beschleunigungsmuster genau zu erfassen[24]. Baños u. a. [3] kam zu dem Ergebnis,

dass Abschnitte von 1 – 2 Sekunden einen guten Kompromiss zwischen Erkennungsgeschwindigkeit und Genauigkeit bieten. Außerdem können kleinere Fenster von 0.25 – 0.5 Sekunden für die Erkennung bestimmter kurzer Aktivitäten höhere Genauigkeit aufweisen. Dies deckt sich mit den Ergebnissen Jaén-Vargas u. a. [14], die zeigten, dass bei einer Abtastrate von 100 Hz ein Fenster von 20 Frames (0.2 Sekunden) für eine genaue Erkennung von Walking, Sit-to-Stand und Squatting-Aktivitäten ausreichend ist.

Außerdem stehen die Fensterlänge und verwendete Abtastrate des Accelerometers im direkten Zusammenhang [24]. Zur Sammlung von ausreichend Datenpunkten für eine präzise Analyse bei einer niedrigeren Abtastrate von 20-50 Hz werden längere Fenster von 2-3 Sekunden verwendet. Um Energie einzusparen und somit die Akkulaufzeit zu verlängern, arbeitet man mit niederfrequenten Abtastraten bei Sportuhren. Die Kombination reicht für alltägliche Aktivitätserkennung wie Gehen, Stehen oder Sitzen aus. Komplexere Bewegungsanalysen wie die Erkennung von Laufstilen oder die Sturzerkennung profitieren von höheren Abtastraten (50 – 100 Hz) mit kürzeren Fenstern von 1-2 Sekunden, die eine feinere zeitliche Auflösung bieten. [24]

Moderne Sportuhren variieren tatsächlich ihre Abtastraten abhängig von der erkannten Bewegung oder dem ausgewählten Sportmodus [29]. Für Alltagsaktivitäten wie Gehen wird oft eine niedrigere Abtastrate (z.B. 10 – 30 Hz)[19] verwendet, während bei intensiveren Aktivitäten wie Laufen oder Sport höhere Raten (50 – 100 Hz) eingesetzt werden [26]. Für die Analyse von komplexeren Sportarten wie Schwimmen oder Golfschwung-Analysen bieten die Top-Geräte die Aktivierung von noch höheren Abtastraten (> 100 Hz), um detailliertere Bewegungsmuster zu erfassen [24].

Der Zusammenhang zwischen Abtastrate und Frequenzinhalt der zu messenden Aktivität ist entscheidend [18]. Nach dem Nyquist-Shannon-Abtasttheorem muss die Nyquist-Frequenz (die Hälfte der Abtastrate) höher sein als die höchste relevante Frequenzkomponente der gemessenen Aktivität [18]. Nachdem die relevanten Frequenzen beim Gehen nur bis 3-4 Hz reichen, sind niedrige Abtastraten ausreichend. Beim Laufen entstehen höhere Frequenzkomponenten.

Zusammengefasst werden Fenstergröße und Abtastrate basierend auf den Eigenschaften der zu erkennenden Aktivitäten, der Komplexität der Sensordaten und den Anforderungen der Anwendung gewählt, um die Erkennungsgenauigkeit als auch die Batterielebensdauer zu maximieren. Eine pauschale Antwort für die Verwendung von optimaler Fenstergröße und Abtastrate gibt es nicht, da die Sportuhren am Markt verschiedenste Zusammenspiele von Hard- und Software-Konfigurationen aufweisen.

2.5 Zusammenfassung

In diesem Kapitel wurde der Weg des analogen Beschleunigungssignals durch das Abtasten mit Hilfe eines Sensors zu einem digitalen Signal erklärt. Bevor die Daten weiterverarbeitet werden können, setzt man Tief- und Hochpassfilter zur Rauschunterdrückung und

2 Erfassung und Vorverarbeitung der Rohdaten

Trennung der Gravitationsbeschleunigung aus den Daten ein. In vielen Fällen normalisiert man die Daten, um die weiteren Rechenschritte zu vereinfachen. Vor der Feature-Extraktion wird das kontinuierliche Signal in sich überlappende Fenster segmentiert. Eine gute Kombination aus Abtastrate und Fensterlänge beeinflusst die Genauigkeit des Endergebnisses der Klassifikation. Bevor der nächste Schritt der Merkmal-Extraktion betrachtet wird, werden im nächsten Kapitel die Grundlagen der Fourieranalyse als Vorbereitung bearbeitet.

3 Theoretische Grundlagen der Fouriertransformation in der Signalverarbeitung

Die Analyse von Sensordaten spielt eine zentrale Rolle in der Aktivitätserkennung von Sportuhren. Beschleunigungssensoren und Gyroskope zeichnen die menschlichen Bewegungen in Form von komplexen Signalmustern auf, die in ihrer Rohform schwer zu interpretieren sind. Erst mit Hilfe von geeigneten mathematischen Werkzeugen lassen sich nützliche Daten wie Aktivitätsart, Schrittzahl und Bewegungsqualität berechnen. Die Fourieranalyse stellt in diesem Kontext eine der fundamentalsten Methoden zur Merkmals-Extraktion dar. Der Franzose, Jean-Baptiste Joseph Fourier entwickelte am Anfang des 19. Jahrhunderts die Fourieranalyse, ein mathematisches Konzept, basierend auf der Erkenntnis, dass periodische Funktionen als Summe von Sinus- und Kosinusfunktionen gesehen werden können.

Heute bildet diese Theorie, die ihren Anfang in der Untersuchung von Wärmeausbreitung hatte, die Grundlage vieler technischer Entwicklungen, speziell im Bereich der digitalen Welt, und so auch von der digitalen Signalverarbeitung. Das Identifizieren von harmonischen Komponenten durch die Zerlegung eines komplexeren Signals ermöglicht, Periodizität und Frequenzmuster im Frequenzbereich zu erkennen, was im Zeitbereich oft verborgen bleibt.

Dieses Kapitel dient als Einführung zu den theoretischen Grundlagen der Fourieranalyse in der Signalverarbeitung. Dieses mathematische Konzept wird systematisch von den Fourier-Reihen für periodische Signale bis hin zur diskreten Fouriertransformation (DFT) und ihrer effizienten Implementierung als Fast Fourier Transform (FFT) erarbeitet. Nachdem das Wissen dieses Kapitels grundlegend für das Verständnis der nachfolgenden Merkmalsextraktion bei Beschleunigungsdaten ist, liegt der Fokus auf den mathematischen Eigenschaften der Transformation, der Interpretation des Frequenzspektrums sowie den Grenzen und Herausforderungen bei der Anwendung auf nicht-stationäre Signale, wie sie bei menschlichen Bewegungen entstehen.

3.1 Grundprinzipien der Signalanalyse: Zeit- und Frequenzbereich

Die Sensorsignale lassen sich in zwei komplementären Domänen, dem Zeit- und dem Frequenzbereich, analysieren. Beide Darstellungen repräsentieren dieselben Informationen auf unterschiedliche Weise und ermöglichen dadurch zwei verschiedene Blickwinkel auf die Signaleigenschaften.

3.1.1 Zeitbereich

Bei Funktionen der Zeit handelt es sich um Signale des Zeitbereichs [20]. Wie in 2.2 erklärt, gehören die Rohdaten des Beschleunigungsmessers einer Sportuhr, die in regelmäßigen Zeitabschnitten erfasst werden, zu der Funktionsgruppe:

$$x(t) \text{ oder } x[n] \text{ für diskrete Signale} \quad (3.1)$$

Aus der Form der Darstellung kann die Veränderung der Beschleunigung pro Zeiteinheit abgelesen werden. Folgende für die Bewegungsanalyse relevante Merkmale können in diesem Bereich extrahiert werden:

- Amplituden (Maxima und Minima der Beschleunigung)
- Zeitliche Muster und Abfolgen von Ereignissen
- Übergänge zwischen verschiedenen Aktivitäten
- Statistische Eigenschaften wie Mittelwert, Varianz und Standardabweichung

In der Zeitdomäne scheitert jedoch die Identifikation von periodischen Mustern.

3.1.2 Frequenzbereich

Der Frequenzbereich beschreibt dasselbe Signal als Funktion der Frequenz. Die Frequenzkomponenten der Superposition, dem Gesamtsignal, werden dadurch sichtbar gemacht. Die Fouriertransformation ermöglicht die Umwandlung von Zeit- zu Frequenzdomäne [20]:

$$X(f) = \int_{-\infty}^{\infty} x(t)e^{-j2\pi ft} dt \quad (3.2)$$

Verborgene Periodizitäten können durch diese Transformation identifiziert werden. Nachdem Aktivitäten verschiedene charakteristische Frequenzmuster erzeugen, ist dies besonders interessant für die Bewegungserkennung. Berechnungen in dieser Domäne ermöglichen die Bestimmung von dominanten Frequenzen und harmonischen Strukturen als

auch von der Energieverteilung über Frequenzbänder und der Unterscheidung zwischen Bewegungen und Rauschen im Signal [20].

Moderne Sportuhren kombinieren beide Domänen für die Durchführung der Bewegungsanalyse. Während der Zeitbereich gut für die Erkennung von abrupten Änderungen und nicht-periodischen Ereignissen geeignet ist, ermöglicht der Frequenzbereich die präzise Identifikation rhythmischer Aktivitäten. Für die Umwandlung vom Zeit- in den Frequenzbereich benötigen wir ein mathematisches Werkzeug, mit dem sich der nächste Abschnitt befassen wird.

3.2 Die Fourier-Reihe

In technischen Bereichen wie der Signalverarbeitung und der Informatik wird ein Signal als Funktion, die Informationen über den Zustand oder das Verhalten eines physikalischen Systems übermittelt, angenommen. Im vorherigen Kapitel 2.2 wurde festgehalten, dass natürlich vorkommende Signale mithilfe trigonometrischer Funktionen wie Sinus- und Cosinus dargestellt werden können. Bei Beschleunigungsdaten handelt es sich meist um eine Superposition von mehreren Funktionen, da der Sensor die Signale von den verschiedenen Teilbewegungen aufnimmt.

Im Kapitel 2.2 wurden die Sinusfunktion und Kosinusfunktion zur Veranschaulichung von Schwingungen und menschlichen Bewegungen genannt. Eine nähere Betrachtung der zwei trigonometrischen Funktionen zeigt ihre Beziehung zueinander. Sie sind phasenverschoben:

$$\cos(x) = \sin\left(x + \frac{\pi}{2}\right) \quad (3.3)$$

Bei den Funktionen $\cos(nx)$, $\sin(nx)$ mit $n \in \mathbb{N}$ handelt es sich um die Basisfunktionen des Vektorraums $L^2[-\pi, \pi]$ der 2π -periodischen, quadratisch-integrierbaren Funktionen [20]:

$$L^2_{\text{per}}[-\pi, \pi] = \left\{ f : \mathbb{R} \rightarrow \mathbb{C} \mid f(x + 2\pi) = f(x) \text{ für alle } x \in \mathbb{R}, \int_{-\pi}^{\pi} |f(x)|^2 dx < \infty \right\} \quad (3.4)$$

Zusätzlich handelt es sich um ein Orthogonalsystem mit dem Skalarprodukt

$$(f, g) = \int_{-\pi}^{\pi} f(x)g(x)dx,$$

da die Cosinus- und Sinusfunktion orthogonal zueinander stehen. Das bedeutet, dass ihr Skalarprodukt über ein bestimmtes Intervall null ist.

Folgende Eigenschaften folgen aus der Orthogonalität [20] für $n, m \in \mathbb{N}, n \neq m$:

$$\int_{-\pi}^{\pi} \sin(mx) dx = \int_{-\pi}^{\pi} \cos(mx) dx = 0 \quad (3.5)$$

3 Theoretische Grundlagen der Fouriertransformation in der Signalverarbeitung

$$\int_{-\pi}^{\pi} \sin(mx) \sin(nx) dx = \int_{-\pi}^{\pi} \cos(mx) \cos(nx) dx = 0 \quad (3.6)$$

$$\int_{-\pi}^{\pi} \sin(mx) \cos(nx) dx = 0 \quad (3.7)$$

$$\int_{-\pi}^{\pi} \sin^2(mx) dx = \int_{-\pi}^{\pi} \cos^2(mx) dx = \pi \quad (3.8)$$

Durch Nachrechnen lassen sich die Eigenschaften beweisen, worauf in dieser Arbeit verzichtet wird.

Die Fourier-Reihe bildet das mathematische Fundament der Fourieranalyse und das mathematische Werkzeug, um mit komplexen periodischen Signalen wie den Beschleunigungsdaten zu arbeiten. Es handelt sich um einen eleganten Ansatz, periodische Funktionen als unendliche Summe von Sinus- und Kosinusfunktionen zu repräsentieren, was zu einer Zerlegung eines Signals in seine harmonischen Komponenten führt. Die Orthogonalität ermöglicht erst die Fouriertransformation, da einerseits die Zerlegung einer Funktion in ihre Frequenzkomponenten dadurch eindeutig ist und die Fourier-Koeffizienten einfach berechnet werden können.

Definition 3.1 Sei $f : (-T/2, T/2) \rightarrow \mathbb{R}$ eine stückweise stetige Funktion mit der Periode T , so kann f als Fourier-Reihe wie folgt dargestellt werden:

$$f(t) = \frac{a_0}{2} + \sum_{k=1}^{\infty} \left[a_k \cos\left(\frac{2\pi kt}{T}\right) + b_k \sin\left(\frac{2\pi kt}{T}\right) \right] \quad (3.9)$$

Wobei die Fourierkoeffizienten a_0 , a_k und b_k durch folgende Integrale definiert sind:

$$a_0 = \frac{2}{T} \int_{-T/2}^{T/2} f(t) dt \quad (3.10)$$

$$a_k = \frac{2}{T} \int_{-T/2}^{T/2} f(t) \cos\left(\frac{2\pi kt}{T}\right) dt \quad (3.11)$$

$$b_k = \frac{2}{T} \int_{-T/2}^{T/2} f(t) \sin\left(\frac{2\pi kt}{T}\right) dt \quad (3.12)$$

Die Zerlegung der Funktion in die durch die trigonometrischen Basisfunktionen dargestellten Schwingungen spiegelt das Anschreiben einer Funktion als ihre Fourier-Reihe wider. Diesen Prozess nennt man Fourieranalyse.

Basierend auf der normalisierten Schwingungsbeschreibung durch Kosinus (2.1), lässt sich eine alternative Darstellung der Fourierreihe mit $\cos_k(t) := \sqrt{2} \cos(2\pi kt)$ und $\sin_k(t) := \sqrt{2} \sin(2\pi kt)$ herleiten:

$$f(t) = a_0 + \sum_{k \in \mathbb{N}} a_k \cos_k(t) + \sum_{k \in \mathbb{N}} b_k \sin_k(t) \quad (3.13)$$

3.2.1 Eindeutigkeit der Fourier-Reihe

Wie bereits angedeutet, sind die Fourier-Koeffizienten und die Fourierreihe einer Funktion eindeutig.

Beweis 1 Die Eindeutigkeit der Zerlegung einer periodischen Funktion in ihre Frequenzkomponenten ist eine direkte Konsequenz der Orthogonalität der trigonometrischen Basisfunktionen. In diesem Abschnitt wird die Eindeutigkeit bewiesen.

Formulierung des Problems: Wir betrachten eine 2π -periodische Funktion $f \in L^2[-\pi, \pi]$ und nehmen an, dass f zwei verschiedene Fourier-Reihenentwicklungen besitzt:

$$f(x) = \frac{a_0}{2} + \sum_{k=1}^{\infty} [a_k \cos(nx) + b_k \sin(nx)]$$

und

$$f(x) = \frac{c_0}{2} + \sum_{k=1}^{\infty} [c_k \cos(nx) + d_k \sin(nx)]$$

Wir wollen zeigen, dass $a_0 = c_0$, $a_k = c_k$ und $b_k = d_k$ für alle $k \in \mathbb{N}$.

Beweis Angenommen, es handelt sich um zwei verschiedene Darstellungen. In dem Fall muss für einen der Koeffizienten $a_k \neq c_k$ oder $b_k \neq d_k$ gelten. Wir bilden die Differenzfunktion aus den zwei Fourier-Reihen:

$$g(x) = f(x) - f(x) = \frac{a_0 - c_0}{2} + \sum_{k=1}^{\infty} [(a_k - c_k) \cos(kx) + (b_k - d_k) \sin(kx)]$$

Aus $f(x) - f(x) = 0, \forall x$, folgt

$$g(x) = 0 \quad (3.14)$$

für alle x .

Mithilfe der Orthogonalitätseigenschaften zeigen wir das Wegfallen der Differenzkoeffizienten.

3 Theoretische Grundlagen der Fouriertransformation in der Signalverarbeitung

Schritt 1: z.z. $a_0 = c_0$

Wir integrieren $g(x)$ über $[-\pi, \pi]$:

$$\int_{-\pi}^{\pi} g(x) dx = \int_{-\pi}^{\pi} \left[\frac{a_0 - c_0}{2} + \sum_{k=1}^{\infty} [(a_k - c_k) \cos(kx) + (b_k - d_k) \sin(kx)] \right] dx$$

Aufgrund der Orthogonalitätseigenschaften (3.5) und $\int_{-\pi}^{\pi} 1 dx = 2\pi$ erhalten wir:

$$\int_{-\pi}^{\pi} g(x) dx = \frac{a_0 - c_0}{2} \cdot 2\pi = \pi(a_0 - c_0)$$

Aus (3.14), folgt:

$$\pi(a_0 - c_0) = 0 \Rightarrow a_0 = c_0$$

Schritt 2: z.z. $a_m = c_m$ für ein beliebiges $m \in \mathbb{N}$

Dafür integrieren wir das Produkt aus $g(x)$ mit $\cos(mx)$:

$$\int_{-\pi}^{\pi} g(x) \cos(mx) dx = \int_{-\pi}^{\pi} \left[\frac{a_0 - c_0}{2} + \sum_{k=1}^{\infty} [(a_k - c_k) \cos(kx) + (b_k - d_k) \sin(kx)] \right] \cos(mx) dx$$

Aufgrund der Orthogonalitätseigenschaften 3.5 und 3.6 für die Kosinusfunktion sowie 3.7 als auch

$$\int_{-\pi}^{\pi} \cos(mx) \cos(mx) dx = \pi \text{ für } m \neq 0$$

kann die Gleichung vereinfacht werden:

$$\int_{-\pi}^{\pi} g(x) \cos(mx) dx = (a_m - c_m) \cdot \pi$$

Da 3.14 gilt, folgt:

$$(a_m - c_m) \cdot \pi = 0 \Rightarrow a_m = c_m$$

Schritt 3: z.z. $b_m = d_m$ für ein beliebiges $m \in \mathbb{N}$

Analog multiplizieren wir $g(x)$ mit $\sin(mx)$ und integrieren:

$$\int_{-\pi}^{\pi} g(x) \sin(mx) dx = \int_{-\pi}^{\pi} \left[\frac{a_0 - c_0}{2} + \sum_{n=1}^{\infty} [(a_n - c_n) \cos(nx) + (b_n - d_n) \sin(nx)] \right] \sin(mx) dx$$

Mit den Orthogonalitätseigenschaften 3.5 und 3.6 für die Sinusfunktion sowie 3.7 als auch $\int_{-\pi}^{\pi} \sin(mx) \sin(mx) dx = \pi$ für $m \neq 0$ erhalten wir:

$$\int_{-\pi}^{\pi} g(x) \sin(mx) dx = (b_m - d_m) \cdot \pi$$

Daraus folgt

$$(b_m - d_m) \cdot \pi = 0 \Rightarrow b_m = d_m$$

aufgrund von (3.14) .

Da nun gezeigt wurde, dass $a_0 = c_0$, $a_k = c_k$ und $b_k = d_k$ für alle $k \in \mathbb{N}$, sind die beiden Fourier-Reihenentwicklungen identisch. In anderen Worten sind die Zerlegung einer Funktion in ihre Frequenzkomponenten eindeutig.

Die Eindeutigkeit dieses mathematischen Werkzeugs bildet die Grundlage für zuverlässige und präzise Bewegungsanalysen in Sportuhren, da sie garantiert, dass jede periodische Funktion genau eine Darstellung als Summe von Sinus- und Kosinustermen besitzt. Funktionen wie Aktivitätserkennung, Leistungsanalyse und biomechanisches Feedback basieren darauf, dass jedem Bewegungsmuster ein spezifisches Frequenzspektrum zugeordnet werden kann. Die Eindeutigkeit hat somit eine hohe Relevanz für die Genauigkeit und Zuverlässigkeit moderner Sportuhren.

3.2.2 Die Symmetrieeigenschaften der Basisfunktionen

Die Symmetrieeigenschaften der trigonometrischen Basisfunktionen spielen eine wichtige Rolle in der Signalverarbeitung. Besonders von Bedeutung ist die Symmetrie in rechenleistungsbeschränkten Anwendungsbereichen wie der Bewegungsanalyse von Sportuhren. Bei der Kosinusfunktion handelt es sich um eine symmetrische (gerade) Funktion, die durch

$$f(-x) = f(x)$$

charakterisiert ist. Für die Kosinusfunktion gilt also [17]:

$$\cos(-x) = \cos(x) \quad \forall x \in \mathbb{R}$$

Die Sinusfunktion gehört zu den asymmetrischen (ungeraden) Funktionen, für die $f(-x) = -f(x)$ gilt. Daher besitzt die Sinusfunktion die folgende Eigenschaft [17]:

$$\sin(-x) = -\sin(x) \quad \forall x \in \mathbb{R}.$$

Aufgrund der genannten Symmetrieeigenschaften der Basisfunktion kann jedes Signal mit Hilfe der Fourierreihe in gerade (symmetrische) und ungerade (antisymmetrische) Komponenten zerlegt werden: Die Kosinusterme repräsentieren die geraden Anteile und Sinusterme die ungeraden Anteile.

Die Produkte von geraden und ungeraden Funktionen verhalten sich folgendermaßen [17]:

3 Theoretische Grundlagen der Fouriertransformation in der Signalverarbeitung

1. Produkt zweier gerader Funktionen (f und g) ergibt eine gerade Funktion:

$$(f \cdot g)(-x) = f(-x) \cdot g(-x) \quad (3.15)$$

$$= f(x) \cdot g(x) \quad (3.16)$$

$$= (f \cdot g)(x) \quad (3.17)$$

2. Produkt zweier ungerader Funktionen (f und g) ist eine gerade Funktion:

$$(f \cdot g)(-x) = f(-x) \cdot g(-x) \quad (3.18)$$

$$= (-f(x)) \cdot (-g(x)) \quad (3.19)$$

$$= f(x) \cdot g(x) \quad (3.20)$$

$$= (f \cdot g)(x) \quad (3.21)$$

3. Produkt einer geraden (f) und einer ungeraden (g) Funktion wird jedoch zu einer ungeraden Funktion:

$$(f \cdot g)(-x) = f(-x) \cdot g(-x) \quad (3.22)$$

$$= f(x) \cdot (-g(x)) \quad (3.23)$$

$$= -f(x) \cdot g(x) \quad (3.24)$$

$$= -(f \cdot g)(x) \quad (3.25)$$

Das Integral einer ungeraden Funktion über ein symmetrisches Intervall wie $[-\pi, \pi]$ entspricht null, da sich die Flächeninhalte oberhalb und unterhalb der x -Achse aufheben. Die oben genannten Eigenschaften und diese Tatsache vereinfachen die Berechnung der Fourierkoeffizienten stark, worauf im nächsten Abschnitt näher eingegangen wird.

Auswirkungen auf die Fourierreihe

Die allgemeine Form der Fourierreihe einer 2π -periodischen Funktion $f(x)$ ist:

$$f(x) = \frac{a_0}{2} + \sum_{k=1}^{\infty} [a_k \cos(kx) + b_k \sin(kx)]$$

Für gerade Funktionen wirkt sich die Symmetrieeigenschaft wie folgt aus [17]: Ist eine Funktion symmetrisch, so gilt $b_k = 0$ für alle $k \in \mathbb{N}$. Nachdem das Produkt $f(x) \sin(nx)$ ungerade und dessen Integral über $[-\pi, \pi]$ null ist, reduziert sich die Fourierreihe von geraden Funktionen zu $f(x) = \frac{a_0}{2} + \sum_{k=1}^{\infty} a_k \cos(kx)$. Bei ungeraden Funktionen fallen alle Kosinuskoeffizienten $a_0 = 0$ und $a_k = 0$ weg und die Fourierreihe entspricht $f(x) = \frac{a_0}{2} + \sum_{k=1}^{\infty} a_k \sin(kx)$ [17]. Zusammengefasst enthält die Fourierreihe von geraden Funktionen ausschließlich Kosinusterme und von ungeraden Funktionen ausschließlich Sinusterme.

Relevanz für die Bewegungserkennung

Das Ausnutzen der Symmetrieeigenschaften der Basisfunktionen durch die Zerlegung eines Signals in seine symmetrischen und asymmetrischen Komponenten verringert den Rechenaufwand um die Hälfte. Sportuhren müssen daher nur die Hälfte der Fourierkoeffizienten für eine Funktion mit bekannter Symmetrie berechnen, wodurch Rechenzeit und Energie eingespart werden können. Außerdem verringert es den verbrauchten Speicher der Sportuhr, der stark begrenzt ist, da nur halb so viele Werte gespeichert werden müssen. Zusätzlich erzeugen viele Aktivitäten wie Gehen oder Laufen Beschleunigungssignale, die näherungsweise gerade und ungerade Komponenten enthalten, wodurch die Symmetrie auch direkt bei der Klassifizierung von Bewegungen unterstützt.

3.2.3 Die komplexe Fourierreihe

Das Konvertieren von der Fourier-Reihe vom Reellen ins Komplexe vereinfacht Berechnungen für die Fourier-Transformation und das Beweisen von beispielsweise den Eigenschaften der Sinus- und Kosinusfunktion. Die Beziehung zwischen der komplexen Exponentialfunktion und den trigonometrischen Funktionen wird durch die Eulersche Formel beschrieben [17].

Definition 3.2 *Eulersche Formel: Für $x \in \mathbb{C}$ gilt*

$$e^{ix} = \cos(x) + i \sin(x) \quad (3.26)$$

Daraus folgt, dass Kosinus und Sinus mit

$$\sin(x) = \frac{1}{2i} (e^{ix} - e^{-ix}) \quad \text{und} \quad \cos(x) = \frac{1}{2} (e^{ix} + e^{-ix}) \quad (3.27)$$

durch die komplexe Exponentialfunktion ausgedrückt werden können.

Die folgenden Eigenschaften machen die komplexe Exponentialfunktion zu einer sinnvollen Basis für die Fourieranalyse [20]:

$$\exp(i\gamma) = \exp(i(\gamma + 2\pi)), \quad (3.28)$$

$$|\exp(i\gamma)| = 1, \quad (3.29)$$

$$\overline{\exp(i\gamma)} = \exp(-i\gamma), \quad (3.30)$$

$$\exp(i(\gamma_1 + \gamma_2)) = \exp(i\gamma_1) \exp(i\gamma_2) \quad (3.31)$$

Die erste Eigenschaft (3.28) zeigt die Periodizität der komplexen Exponentialfunktion mit einer Periode von 2π . Der Ablauf einer Periode, einer vollständigen Rotation um 2π führt zum Ausgangspunkt zurück. Durch (3.28) kann man auf die Periodizität der Fourier-Basis schließen. Da der Betrag stets 1 unabhängig des Arguments γ (3.29) ist, liegen alle Werte von $\exp(i\gamma)$ auf dem Einheitskreis von \mathbb{C} (3.29). Dadurch eignet sich die

komplexe Exponentialfunktion besonders für die Analyse von Schwingungen ohne Dämpfung. Relevant für die Symmetrieeigenschaften der Fourier-Transformation ist (3.30). Die komplexe Konjugierte der Exponentialfunktion entspricht der Exponentialfunktion mit negativem Argument. Die Multiplikativität (3.31) der Exponentialfunktion wandelt die Addition im Exponenten zu Multiplikationen um. Die Zerlegung von komplexen Schwingungen und weitere Berechnungen in der Fourier-Analyse werden durch diese Eigenschaft ermöglicht.

Nun kann die komplexe Fourierreihe eingeführt werden. Dafür wird die reellwertigen Sinusfunktionen \cos_k und \sin_k , die für $k \in \mathbb{N}$ definiert sind, durch die komplexwertigen Exponentialfunktionen $\exp_k : [0, 1] \rightarrow \mathbb{C}$ mit Hilfe von (3.27) ersetzt, definiert durch [20]:

$$\exp_k(t) := \exp(2\pi i k t) \quad (3.32)$$

Das Signal \exp_k hat eine Periode von $(1/k)$ für $k \neq 0$, wobei \exp_0 das konstante Signal 1 beschreibt. Die Menge $\{\exp_k \mid k \in \mathbb{Z}\}$ bildet eine Orthonormalbasis (ON-Basis) des komplexen Hilbert-Raums $L^2[0, 1]$, mit den Eigenschaften $\|\exp_k\| = 1$ für $k \in \mathbb{Z}$ und $\langle \exp_k \mid \exp_\ell \rangle = 0$ für $k \neq \ell$.

Definition 3.3 Die komplexe Fourier-Reihe eines beliebigen Signals $f \in L^2([0, 1])$ kann wie folgt angeschrieben werden:

$$f(t) = \sum_{k \in \mathbb{Z}} c_k \exp_k(t) = \sum_{k \in \mathbb{Z}} c_k \exp(2\pi i k t) \quad (3.33)$$

Die entsprechenden (komplexen) Fourier-Koeffizienten $c_k \in \mathbb{C}$ werden durch folgende Gleichung berechnet:

$$c_k = \int_{t \in [0, 1]} f(t) \exp(-2\pi i k t) dt \quad (3.34)$$

Die **Fourier-Transformation** von $f \in L^2([0, 1])$ $\hat{f} : \mathbb{Z} \rightarrow \mathbb{C}$ ist wie folgt definiert:

$$\hat{f}(k) := c_k \quad (3.35)$$

Hier wird ein kontinuierliches Zeitsignal auf ein diskretes projiziert.

Energieerhaltung

Zwischen f und \hat{f} lässt sich eine weitere wichtige Eigenschaft herauslesen. Die Fouriertransformation erhält die Energie, sodass f und \hat{f} die gleiche haben.

Für ein besseres Verständnis wird zuerst die Frage beantwortet, wie die Energie eines Signals mathematisch ausgedrückt wird.

Definition 3.4 *Energie eines kontinuierlichen Signals*

$$E = \int_{-\infty}^{\infty} |f(t)|^2$$

Definition 3.5 *Energie eines diskreten Signals*

$$E = \sum_{n=-\infty}^{\infty} |f[n]|^2$$

Die Energieerhaltung der Fourier-Transformation wird durch das Parseval-Theorem ausgedrückt [20]:

$$\|f\|_{L^2([0,1])} = \|\hat{f}\|_{\ell^2(\mathbb{Z})} \quad (3.36)$$

Die Energie von \hat{f} wird im Vektorraum $\ell^2(\mathbb{Z})$ gemessen und die von f im Vektorraum $L^2([0,1])$. Diese energieerhaltende Beziehung zwischen zwei Hilbert-Räumen wird Isometrie genannt. Daraus folgt, dass es sich bei der Abbildung $\hat{f} \rightarrow f$, der Fourierreihe ebenfalls um eine energieerhaltende Isometrie handelt [20].

In der Praxis garantiert die Energieerhaltung, dass die Transformation eines Signals vom Zeit- in den Frequenzbereich keine Information verliert. In anderen Worten sind beide Darstellungen vollständig äquivalent. Als Konsequenz können in der Bewegungsanalyse durch Beschleunigungsdaten Aktivitäten sowohl im Zeitbereich (als Beschleunigungsmuster) als auch im Frequenzbereich ohne Informationsverlust analysiert werden. Weiters hilft die Energieerhaltung bei der Entwicklung von Filtern für Sensordaten. Das Ziel bei der Filterung ist die Unterdrückung von unerwünschten Frequenzkomponenten bei gleichzeitigem Erhalt der Gesamtenergie des Signals.

3.2.4 Die reellen und komplexen Fourier-Koeffizienten

Ein grundlegendes Verständnis der Beziehung zwischen komplexen und reellen Fourierkoeffizienten ist essenziell für die Signalverarbeitung.

Ein reelles Signal $f(t)$, wie es von den Beschleunigungssensoren einer Sportuhr geliefert wird, kann auf zwei Arten dargestellt werden, der komplexen und reellen Fourierreihe [20]:

$$f(t) = a_0 + \sum_{k=1}^{\infty} [a_k \cos(2\pi kt) + b_k \sin(2\pi kt)] \quad (3.37)$$

und

$$f(t) = \sum_{k \in \mathbb{Z}} c_k \exp_k(t) = \sum_{k \in \mathbb{Z}} c_k \exp(2\pi ikt) \quad (3.38)$$

3 Theoretische Grundlagen der Fouriertransformation in der Signalverarbeitung

Aus der Symmetrie der Basisfunktionen (siehe 3.2.1) folgt die Redundanz der komplexen Koeffizienten [20]:

$$c_{-k} = \overline{c_k} \quad (3.39)$$

Das bedeutet, dass der Koeffizient für die negative Frequenz $-k$ das komplexe Konjugierte des Koeffizienten für die positive Frequenz k ist. Diese Eigenschaft wird von den komplexen Koeffizienten erfüllt, da ein reelles Signal keine komplexen Anteile enthalten kann. Daher müssen sich alle imaginären Komponenten in der Fourierreihe gegenseitig aufheben.

Folgende Formeln berechnen die reellen (a_k, b_k) aus den komplexen (c_k) Koeffizienten aus:

$$a_0 = c_0 \quad (3.40)$$

$$a_k = \sqrt{2} \operatorname{Re}(c_k) = c_k + c_{-k} \quad (3.41)$$

$$b_k = -\sqrt{2} \operatorname{Im}(c_k) = i(c_k - c_{-k}) \quad (3.42)$$

Umgekehrt kann man die komplexen Koeffizienten aus den reellen berechnen:

$$c_0 = a_0 \quad (3.43)$$

$$c_k = \frac{a_k - ib_k}{2} \text{ für } k > 0 \quad (3.44)$$

$$c_{-k} = \frac{a_k + ib_k}{2} \text{ für } k > 0 \quad (3.45)$$

Diese Umwandlung findet häufig Anwendung in der Bewegungsanalyse mit Sportuhren, da je nach Berechnung die sinnvollere Form gewählt wird. Bei der FFT, der Phasenberechnung und für detailliertere Bewegungsanalysen enthalten komplexe Koeffizienten mehr Informationen. Bei den Eingangsdaten als auch dem Leistungsspektrum und der Diskreten Fourier Transformation (DFT) entstehen stets reelle Werte. Zusammengefasst erlaubt uns die Redundanz, effizienter zu rechnen und dennoch alle Informationen des Signals zu erhalten.

Der Beweis für (3.41), (3.42) und (3.40) zeigt die Schritte der Umrechnung anschaulich:

$$\begin{aligned}
f(t) &= \sum_{k \in \mathbb{Z}} c_k \mathbf{exp}_k(t) \\
&= c_0 + \sum_{k=1}^{\infty} \left(c_k \mathbf{exp}_k(t) + \overline{c_k \mathbf{exp}_k(t)} \right) \\
&= c_0 + \sum_{k=1}^{\infty} 2 \operatorname{Re} (c_k \mathbf{exp}_k(t)) \\
&= c_0 + \sum_{k=1}^{\infty} (2 \operatorname{Re}(c_k) \cos(2\pi k t) - 2 \operatorname{Im}(c_k) \sin(2\pi k t)) \\
&= c_0 + \sum_{k=1}^{\infty} \sqrt{2} \operatorname{Re}(c_k) \mathbf{cos}_k(t) + \sum_{k=1}^{\infty} \left(-\sqrt{2} \operatorname{Im}(c_k) \right) \mathbf{sin}_k(t)
\end{aligned} \tag{3.46}$$

Der komplexe Fourier-Koeffizient $\hat{f}(\omega)$ kann auch in dieser Form als Summe von Real- und Imaginärteil angeschrieben werden:

$$\hat{f}(\omega) = \int_{t \in \mathbb{R}} f(t) \cos(-2\pi \omega t) dt + i \int_{t \in \mathbb{R}} f(t) \sin(-2\pi \omega t) dt \tag{3.47}$$

Alternativ lässt sich der Fourierkoeffizient in Polarform darstellen:

$$\hat{f}(\omega) = |\hat{f}(\omega)| e^{i\phi_\omega} \tag{3.48}$$

Dabei entspricht $|\hat{f}(\omega)|$ dem Betrag (die Magnitude) und ϕ_ω der Phase. Der Betrag gibt die Stärke der entsprechenden im Signal vertretenen Frequenz an, während die Phase die zeitliche Verschiebung dieser Frequenzkomponente darstellt. Der Betragswert charakterisiert meist bereits die relevanten Frequenzmuster, weswegen die Phaseninformation bei der Bewegungserkennung bei Sportuhren oft vernachlässigt wird.

3.3 Die Fourier-Transformationen

Bis jetzt wurden periodische Funktionen behandelt, da die Fourier-Reihe ausschließlich auf periodische Signale anwendbar ist. Ihre Idee der Repräsentation eines Signals durch die Summe seiner Teilschwingungen kann jedoch mithilfe der Fourier-Transformation auf nicht-periodische kontinuierliche Signale angewandt werden.

In modernen Sportuhren ist die Fourier-Transformation ein zentrales Werkzeug für die Bewegungserkennung. Accelerometer-Sensoren liefern kontinuierlich Beschleunigungsdaten im Zeitbereich, die allein betrachtet wenig Aussagekraft über spezifische Bewegungsmuster haben. Erst die Transformation dieser Daten in den Frequenzbereich ermöglicht die zuverlässige Unterscheidung zwischen Aktivitäten wie Gehen, Laufen oder Schwimmen durch ihre Frequenzmuster.

3 Theoretische Grundlagen der Fouriertransformation in der Signalverarbeitung

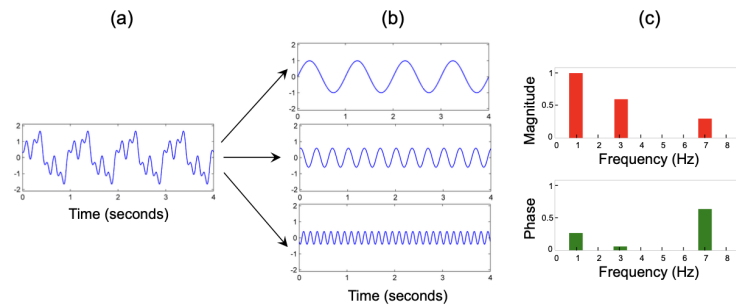


Abbildung 3.1: Transformation von Zeit- in Frequenzbereich

3.3.1 Die kontinuierliche Fourier-Transformation

Definition 3.6 Die komplexe Formulierung der Fourier-Reihe bildet die Grundlage für die Fourier-Transformation eines reellwertigen Signals $f : \mathbb{R} \rightarrow \mathbb{R}$ mit $f \in L^1(\mathbb{R})$, die als eine komplexwertige Funktion $\hat{f} : \mathbb{R} \rightarrow \mathbb{C}$ gemäß definiert wird [20]:

$$\hat{f}(\omega) = \int_{t \in \mathbb{R}} f(t) \exp(-2\pi i \omega t) dt = c_\omega \quad (3.49)$$

Diese Transformation bildet ein Signal aus dem Zeitbereich in den Frequenzbereich ab. Für jede Frequenz ω erhält man einen komplexen Koeffizienten, der als Repräsentation sowohl der Amplitude als auch der Phase der entsprechenden Frequenzkomponente im Originalsignal dient.

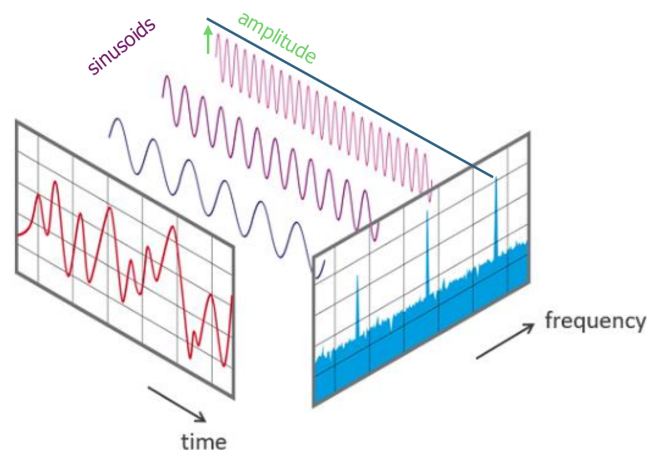


Abbildung 3.2: Fouriertransformation
Quelle: [9]

3.3.2 Die inverse Fourier-Transformation

Die inverse Fouriertransformation (IFT) transformiert die Funktion aus dem Frequenzbereich zurück in den Zeitbereich [20]:

$$f(t) = \int_{\omega \in \mathbb{R}} \hat{f}(\omega) \exp(2\pi i \omega t) d\omega = \int_{\omega \in \mathbb{R}} c_{\omega} \exp(2\pi i \omega t) d\omega \quad (3.50)$$

Die inverse Fouriertransformation ist nicht nur ein mathematisches Gegenstück zur direkten Transformation, sondern ein praktisches Werkzeug im Signalverarbeitungsprozess von Sportuhren. Sie bildet die Brücke zwischen der Frequenzanalyse und konkreten, im Zeitbereich messbaren Bewegungsmustern. Die IFT findet ihre Anwendung bei der Filterung von unerwünschten, störenden Frequenzen, die im Frequenzbereich durchgeführt werden. Dafür wird das Signal vom Zeit- in den Frequenzbereich transformiert mit Hilfe der Fast-Fourier-Transformation (die später in diesem Kapitel eingeführt wird). Anschließend dämpft und entfernt man unerwünschte Frequenzen. Danach wird das Signal wieder umgewandelt, um es in der Zeitdomäne weiterzuverarbeiten. Beispielsweise können bei der Lauferkennung Frequenzen über 10 Hz, um Erschütterungen oder Vibrationen herauszufiltern, gedämpft werden, während die relevanten Schrittfrequenzen (1-4 Hz) erhalten bleiben.

Zur Reduzierung von Sensorrausch glättet man Beschleunigungsdaten. Nachdem bei Sportuhren oft Glättungstechniken im Frequenzbereich durchgeführt werden und das Ergebnis der Glättung Relevanz im Zeitbereich hat, kommt auch hier die inverse Fourier-Transformation zum Einsatz. Weiters findet sie ihren Nutzen bei der Entwicklung von Aktivitätserkennungsalgorithmen. Außerdem werden zur Speicherung manche Sensordaten im Frequenzbereich komprimiert. Diese komprimierten Darstellungen werden erst zur Weiterverarbeitung wieder in den Zeitbereich umgewandelt.

3.4 Die Diskrete Fourier-Transformation

Das Rechnen mit der kontinuierlichen Beschreibung von Signalen durch die Fourierreihe enthält Integrale oder unendliche Summen, die nicht durchführbar sind. Daher wird in der Praxis mit Annäherungen der Fouriertransformation mit endlichen Summen gearbeitet. Im Kapitel zur Datenerfassung (2.2) wurde bereits die Transformation von analogen zu digitalen Daten veranschaulicht. Die kontinuierlichen Bewegungssignale werden durch das Sampling vom Beschleunigungssensor in diskrete Datenpunkte transformiert [20]. Dadurch wird aus dem Ursprungssignal $f : \mathbb{R} \rightarrow \mathbb{R}$ ein diskretes Signal $x : \mathbb{Z} \rightarrow \mathbb{R}$ mit $\mathbb{Z} \subset \mathbb{R}$ der Form

$$x(n) := f(n \cdot T).$$

Während die kontinuierliche Fourier-Transformation mit unendlich vielen Datenpunkten arbeitet, erfordert die praktische Bewegungsanalyse in Sportuhren eine diskrete Herange-

hensweise. Daher wird ein Methodenwechsel von einer kontinuierlichen zu einer diskreten Variante der Fourier-Transformation benötigt. Bei der Diskreten Fourier-Transformation (DFT), die in diesem Abschnitt eingeführt wird, handelt es sich um die gesuchte Methode. Sie ermöglicht das Rechnen mit Bewegungssignalen, die durch den Beschleunigungssensor in diskreten Zeitintervallen erfasst wurden.

3.4.1 Fouriertransformation eines diskreten Signals

Zunächst wird die Fouriertransformation für diskrete Zeitsignale eingeführt, die die spektrale Analyse endlicher Energiesignale im Frequenzbereich ermöglicht.

Definition 3.7 *Fouriertransformation eines diskreten Zeitsignals*

Für ein diskretes Signal $x[n] \in \ell^2(\mathbb{Z})$ gilt [20]:

$$x[n] = \int_{\omega \in [0,1]} c_{\omega} \exp_{\omega}(n) d\omega = \int_{\omega \in [0,1]} c_{\omega} \exp(2\pi i \omega n) d\omega \quad (3.51)$$

Wobei c_{ω} die frequenzabhängigen Fourier-Koeffizienten und $n \in \mathbb{Z}$ die ganzzahligen Zeitindizes beschreiben.

Weiters werden die dazugehörigen Fourier-Koeffizienten wie folgt durch die frequenzabhängige Funktion $\hat{x}(\omega) : [0, 1) \rightarrow \mathbb{C}$ bestimmt [20]:

$$c_{\omega} = \hat{x}(\omega) = \sum_{n \in \mathbb{Z}} x(n) \overline{\exp_{\omega}(n)} = \sum_{n \in \mathbb{Z}} x(n) \exp(-2\pi i \omega n) \quad (3.52)$$

Die diskrete Fourier-Transformation kann als Approximation der kontinuierlichen Fourier-Transformation und Werkzeug der effektiven Analyse von periodischen Signalen betrachtet werden.

Periodizität und Aliasing

Die diskrete Fourier-Darstellung (3.51) besagt, dass ein diskretes Signal x als Superposition von Elementarfunktionen, den komplexen Exponentialfunktionen $\exp(\omega)$, dargestellt werden kann. Diese Funktionen werden mit einer Abtastperiode $T = 1$ abgetastet. Aus der Periodizität erlangt man die wichtige Erkenntnis, dass für die Analyse dieser Darstellung nur Frequenzen ω im Intervall $[0, 1)$ benötigt werden - nicht der gesamte Frequenzbereich von $-\infty$ bis $+\infty$. Folgende Gleichung erklärt die Erkenntnis mathematisch [20]:

$$\exp_{\omega+k}(n) = \exp(2\pi i(\omega + k)n) = \exp(2\pi i \omega n) \exp(2\pi i k n) = \exp(2\pi i \omega n) = \exp_{\omega}(n) \quad (3.53)$$

für ganzzahlige k und n und $\exp(2\pi i k n) = 1$, da die Exponentialfunktion 2π -periodisch ist.

Diese Gleichung besagt, dass wir nach der Abtastung eines kontinuierlichen Signals nicht mehr zwischen Frequenzen unterscheiden können, die sich um ganzzahlige Vielfache unterscheiden. Eine Frequenz von 0.3Hz erscheint nach der Abtastung genau wie Frequenzen von 1.3Hz , 2.3Hz usw. Dieses Phänomen nennt sich **Aliasing**: Höhere Frequenzen ($\omega + k$ mit $k \geq 1$) *tarnen* sich als niedrigere Frequenzen im Basisintervall $[0, 1)$ und sind nach der Abtastung nicht mehr von diesen unterscheidbar [20].

Die praktische Konsequenz ist, dass bei der Analyse von abgetasteten Signalen nur der Frequenzbereich $[0, 1)$ betrachtet werden muss. Jede Frequenz außerhalb dieses Bereichs erscheint durch den Aliasing-Effekt als Frequenz innerhalb dieses Intervalls.

Das **Nyquist-Shannon-Abtasttheorem** stellt Bedingungen auf, sodass ein kontinuierliches, bandbegrenzte Signal verlustfrei abgetastet und rekonstruiert werden kann, ohne dass der *Aliasing* Effekt auftritt.

Theorem 1 *Nyquist-Shannon-Abtasttheorem* Eine Funktion $f(t)$ mit endlicher Energie und Bandbreite W (in Hz) kann vollständig durch ihre Abtastwerte an den Punkten $k/(2W)$ rekonstruiert werden[16]:

$$f(t) = \sum_{k=-\infty}^{\infty} a_k \frac{\sin(2\pi W(t - k/(2W)))}{2\pi W(t - k/(2W))} \quad (3.54)$$

wobei $\sum_{k=-\infty}^{\infty} |a_k|^2 < \infty$ und $a_k = f(k/(2W))$.

Das Theorem legt fest, dass ein Signal mit Bandbreite W Hz mit mindestens $2W$ Abtastungen pro Sekunde (der Nyquist-Rate) für eine einwandfreie Rekonstruktion abgetastet werden muss. Die Bandbreite entspricht dem Frequenzbereich zwischen der niedrigsten und höchsten signifikanten Frequenzkomponente des Signals. Landau [16] zeigt in seinem Paper, dass die Nyquist-Rate eine fundamentale Grenze darstellt. Für eine stabile Rekonstruktion darf sie nicht unterschritten werden, unabhängig von der Wahl der Abtastzeitpunkte oder der spezifischen Struktur der Frequenzbänder.

Beim Arbeiten mit einer Abtastrate von beispielsweise 100Hz in Sportuhren, wird die Analyse von dem Frequenzbereich von 0 bis 50Hz (die Nyquist-Frequenz) durchgeführt. Alle höheren Frequenzen werden durch Aliasing auf diesen Bereich abgebildet und können nicht mehr eindeutig identifiziert werden.

3.4.2 Beschränkte Signale

Um nun von der Fouriertransformation von diskreten Signalen $x \in \ell^2(\mathbb{Z})$ zur DFT zu gelangen, werden beschränkte Signale betrachtet. Bei der Betrachtung der Fouriertrans-

3 Theoretische Grundlagen der Fouriertransformation in der Signalverarbeitung

formation für Signale endlicher Länge lässt sich eine matrixtheoretische Darstellung herleiten, die rechnerisch effizient ist [20]. Dieser Abschnitt widmet sich der in Müller [20] präsentierten Herleitung.

Angenommen die Energie des Signals x mit der Länge N nimmt außerhalb von $[0 : N - 1]$ null ($x(n) \approx 0$) an, dann ergibt sich aus (3.51) folgende Gleichung:

$$\hat{x}(\omega) = \sum_{n \in \mathbb{Z}} x(n) \exp_{\omega}(n) \approx \sum_{n=0}^{N-1} x(n) \exp_{\omega}(n). \quad (3.55)$$

Nachdem \hat{x} 1-periodisch ist, muss weiterhin nur $\omega \in [0, 1)$ berücksichtigt werden.

In der Praxis wird die Fouriertransformation für eine endliche Anzahl von Frequenzen berechnet. Außerdem erfolgt eine $1/K$ -Abtastung des Frequenzraums $[0, 1)$, zur Betrachtung der Frequenzen $\omega = k/K$ für $k \in [0 : K - 1]$. Für die weitere Herleitung nehmen wir $N = K$ an, was eine kompakte matrixtheoretische Formulierung ermöglicht.

Einführung von der Exponentialfunktion als Vektor

Für jedes $k \in [0 : N - 1]$ definieren wir den Vektor $\mathbf{u}_k \in \mathbb{C}^N$ durch

$$u_k(n) := \exp_{k/N}(n) = \exp(2\pi i k n / N) \quad (3.56)$$

für $n \in [0 : N - 1]$. Die ersten N Abtastwerte der kontinuierlichen Exponentialfunktion $\exp_{k/N}$ werden durch diesen Vektor beschrieben.

Die Fouriertransformation für die diskreten Frequenzen $\omega = k/N$ kann nun unter Verwendung von (3.56) als Skalarprodukt

$$\hat{x}(k/N) = \sum_{n=0}^{N-1} x(n) \exp_{k/N}(n) = \mathbf{x}^T \mathbf{u}_k = \langle \mathbf{x} | \mathbf{u}_k \rangle \quad (3.57)$$

ausdrücken. $\mathbf{x} = (x(0), x(1), \dots, x(N - 1))^T \in \mathbb{C}^N$ stellt den Vektor der Signalwerte dar.

Orthonormalität der Exponentialfunktionen

Die skalierten Vektoren \mathbf{u}_k / \sqrt{N} bilden eine Orthonormalbasis des komplexen Hilbertraums \mathbb{C}^N , was aus den Eigenschaften der komplexen Exponentialfunktion (3.30), (3.31) und (3.2) und der geometrischen Summenformel folgt:

$$\langle \mathbf{u}_k | \mathbf{u}_\ell \rangle = \begin{cases} N & \text{für } k = \ell \\ 0 & \text{für } k \neq \ell \end{cases} \quad (3.58)$$

Daher sind die Vektoren $\{\mathbf{u}_k/\sqrt{N} | k \in [0 : N-1]\}$ eine Orthonormalbasis des \mathbb{C}^N .

Dank der Eigenschaft der Orthonormalbasis kann jedes Signal \mathbf{x} eindeutig als Linearkombination dieser Basisvektoren dargestellt werden:

$$\mathbf{x} = \frac{1}{N} \sum_{k=0}^{N-1} \langle \mathbf{x} | \mathbf{u}_k \rangle \mathbf{u}_k \quad (3.59)$$

Die Quintessenz dieses Abschnittes: Ein Signal endlicher Länge kann als gewichtete Überlagerung von abgetasteten und beschränkten Exponentialfunktionen \mathbf{u}_k dargestellt werden. Die Fourierkoeffizienten $\langle \mathbf{x} | \mathbf{u}_k \rangle$ dienen als die Gewichte.

3.4.3 Definition der Diskrete Fourier-Transformation

Die Diskrete Fourier-Transformation (DFT) kann nun mit Hilfe der Darstellung (3.57) der Fourierkoeffizienten definiert werden [20] und bringt uns einen Schritt näher zu der in Sportuhren verwendete Rechenmethode.

Definition 3.8 (Diskrete Fourier-Transformation) Für ein diskretes Signal $x = (x(0), x(1), \dots, x(N-1))^T \in \mathbb{C}^N$ ist die Diskrete Fourier-Transformation definiert als die Abbildung $\mathbb{C}^N \rightarrow \mathbb{C}^N$, die x auf den Vektor $\mathbf{X} = (X(0), X(1), \dots, X(N-1))^T \in \mathbb{C}^N$ abbildet, wobei:

$$X(k) := \langle \mathbf{x} | \mathbf{u}_k \rangle = \sum_{n=0}^{N-1} x(n) \exp\left(-\frac{2\pi i k n}{N}\right), \quad k \in [0 : N-1] \quad (3.60)$$

Der Vektor u_k repräsentiert die k -te Komponente der Fourier-Basis.

Die DFT stellt eine lineare Abbildung dar. Somit kann sie durch eine Matrix-Vektor-Multiplikation ausgedrückt werden. Die entsprechende $N \times N$ Matrix lautet für $n, k \in [0 : N-1]$:

$$\text{DFT}_N(n, k) = \exp(-2\pi i k n / N) \quad (3.61)$$

Eine kompaktere Darstellung ist formulierbar mit der primitive N -te Einheitswurzel $\rho_N = \exp(2\pi i / N)$, der Beziehung $\rho_N^{kn} = \exp(2\pi i k n / N)$ und der Formulierung von $\sigma_N := \overline{\rho_N} = \exp(-2\pi i / N)$. Diese Beziehungen, zusammengefasst in $\text{DFT}_N(n, k) = \sigma_N^{kn}$, ermöglichen die folgende strukturierte Matrix-Darstellung der DFT:

$$\text{DFT}_N = \begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \sigma_N & \sigma_N^2 & \dots & \sigma_N^{N-1} \\ 1 & \sigma_N^2 & \sigma_N^4 & \dots & \sigma_N^{2(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \sigma_N^{N-1} & \sigma_N^{2(N-1)} & \dots & \sigma_N^{(N-1)(N-1)} \end{pmatrix} \quad (3.62)$$

3 Theoretische Grundlagen der Fouriertransformation in der Signalverarbeitung

Es ist anzumerken, dass die Matrix symmetrisch ist, da ihre Spalten durch die Vektoren u_k und ihre Zeilen durch u_k^T definiert sind.

Dank der Matrix $DFT_N(n, k)$ reicht eine einzige Matrix-Vektor Multiplikation $X = DFT_N \cdot x$ zur Berechnung der Fourier-Transformation \hat{x} eines diskreten beschränkten Zeitsignals x mit Länge N für die Frequenzen $\omega = k/N$, $k \in [0 : N - 1]$ aus.

Außerdem ist die Inverse der DFT-Matrix, ebenfalls eine lineare Abbildung, gegeben durch $DFT_N^{-1} = \frac{1}{N} \overline{DFT_N}^T$.

$$DFT_N^{-1} = \frac{1}{N} \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \rho_N & \rho_N^2 & \cdots & \rho_N^{N-1} \\ 1 & \rho_N^2 & \rho_N^4 & \cdots & \rho_N^{2(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \rho_N^{N-1} & \rho_N^{2(N-1)} & \cdots & \rho_N^{(N-1)(N-1)} \end{pmatrix} \quad (3.63)$$

Der nächste Abschnitt befasst sich mit der technischen Umsetzung der Berechnung der DFT.

3.5 Fast Fourier Transform

Nach der Einführung der diskrete Fouriertransformation (DFT), wenden wir uns dem Algorithmus zu, der die praktische Anwendung der DFT überhaupt erst ermöglicht: der Fast Fourier Transform (FFT).

Von der DFT zur FFT: Eine historische Notwendigkeit

Die Berechnung der DFT erfordert für ein Signal mit N Samples ungefähr N^2 komplexe Multiplikationen und Additionen. Heutzutage, im digitalen Zeitalter, umfassen Signale oft Tausende, wenn nicht Millionen Samples. Bei dieser enormen Anzahl an Rechnungen wird die Verwendung der DFT undurchführbar [20]. Ein Signal mit 1.000 Samples würde bereits eine Million Rechenoperationen erfordern. Mit 100.000 Samples steigt dieser Wert auf 10 Milliarden.

Schon vor der Vorstellung des modernen FFT-Algorithmus von Cooley und Tukey 1965 findet er Erwähnung in der Literatur [7]. Runge und König (1924) sowie Stumpff (1937) veröffentlichten bereits Arbeiten mit Fokus auf die Ausnutzung von Symmetrien der Sinus- und Kosinusfunktionen zur Minimierung des Berechnungsaufwands. Die rekursive Struktur des Algorithmus, der später den Durchbruch brachte, erwähnten sie nur beiläufig. Ältere Berechnungsmethoden reichten in dem Prä-Computer-Zeitalter aus, da damals mit weitaus kleineren Datenmengen gearbeitet worden ist. Durch das Aufkommen elektronischer Computer und dem steigenden Bedarf, große Datenmengen zu verarbeiten, entdeckten Cooley, Lewis und Welch [7] den FFT-Algorithmus wieder. Aufgrund

der hohen Recheneffizienz beeinflusste die FFT die digitale Welt stark, insbesondere die Signalverarbeitung.

Der Schlüssel zum Verständnis der FFT liegt in der Beobachtung, dass die DFT einer Sequenz $x(n)$ der Länge N in zwei separate Berechnungen aufgeteilt werden kann [7]:

1. Eine DFT der geradzahlig indizierten Elemente $x(0), x(2), x(4), \dots$
2. Eine DFT der ungeradzahlig indizierten Elemente $x(1), x(3), x(5), \dots$

Diese beiden kleineren DFTs können durch einfache Operationen mit einem *Twiddle-Faktoren* (*Drehfaktoren*) $\sigma_n^k = e^{-2\pi i/N}$ kombiniert werden, um das endgültige Ergebnis zu erhalten [7].

Die folgende Gleichung veranschaulicht die mathematische Darstellung des ersten Schritts der FFT-Zerlegung der ersten M -Einträge $\mathcal{X}(M+k), k \in [M-1]$ [20]. Erst wird die DFT in zwei Teile separiert nach (3.5). Anschließend werden beide Teile mit dem *Twiddle-Faktoren* multipliziert.

$$\begin{aligned}
 \mathcal{X}(k) &= \sum_{n=0}^{N-1} x(n) \sigma_N^{kn} \\
 &= \sum_{n=0}^{M-1} x(2n) \sigma_N^{k2n} + \sum_{n=0}^{M-1} x(2n+1) \sigma_N^{k(2n+1)} \\
 &= \sum_{n=0}^{M-1} x(2n) \sigma_M^{kn} + \sigma_N^k \sum_{n=0}^{M-1} x(2n+1) \sigma_M^{kn}
 \end{aligned} \tag{3.64}$$

Die zweite Gleichung zeigt, dass man die letzten M -Einträge $\mathcal{X}(M+k), k \in [M-1]$ mit dem gleichen Rechenschema nur dem negativen *Twiddle-Faktor* erlangt, was auf die Symmetrieeigenschaften der Exponentialfunktionen zurückzuführen ist (3.2.2).

$$\begin{aligned}
 \mathcal{X}(M+k) &= \sum_{n=0}^{N-1} x(n) \sigma_N^{(M+k)n} \\
 &= \sum_{n=0}^{M-1} x(2n) \sigma_N^{(M+k)2n} + \sum_{n=0}^{M-1} x(2n+1) \sigma_N^{(M+k)(2n+1)} \\
 &= \sum_{n=0}^{M-1} x(2n) \sigma_M^{kn} - \sigma_N^k \sum_{n=0}^{M-1} x(2n+1) \sigma_M^{kn}
 \end{aligned} \tag{3.65}$$

3.5.1 Die Recheneffizienz der FFT

Die außergewöhnliche Recheneffizienz des Fast Fourier Transform (FFT) Algorithmus basiert auf der Reduktion der Komplexität von $O(N^2)$ auf $O(N \log N)$ [7]. Die rekursiven

3 Theoretische Grundlagen der Fouriertransformation in der Signalverarbeitung

Zerlegung der N -Punkt DFT in zwei $N/2$ -Punkt DFTs und somit in kleinere Teilprobleme ermöglicht diese Optimierung. Diese Methode, als *Successive Doubling Method* [7] bekannt, kann rekursiv angewendet werden, solange N eine Potenz von 2 ist. Die Anzahl der Operationen $\mu(N)$ wird durch die Rekursionsgleichung

$$\mu(N) \leq 2 \cdot \mu(N/2) + 1.5N$$

beschrieben [20]. Durch wiederholte Anwendung dieser Rekursion erhält man schließlich $\mu(N) \leq 1.5N \cdot \log_2(N)$ für $N = 2^L$. Dadurch wird gezeigt, dass die Komplexität des FFT-Algorithmus auf $O(N \cdot \log N)$ reduziert ist. Cooley, Lewis und Welch (1969) zeigen, dass sich die Berechnungszeit für $N = 2048$ von 1567.8 Sekunden mit konventionellen Methoden auf nur 2.4 Sekunden mit der FFT reduziert - eine Verbesserung um den Faktor 653.

Wie Cooley, Lewis und Welch [7] anmerken, hat die FFT zu einer Vielzahl von Anwendungen geführt, die aufgrund des Rechenaufwands früher nicht möglich waren. Zu diesen Anwendungen gehören Spektralanalyse, Signalverarbeitung, Fourier-Spektroskopie, Bildverarbeitung und die Lösung von Differentialgleichungen.

Die Recheneffizienz der FFT hat komplexe Frequenzanalysen auf tragbaren Geräten mit begrenzter Rechenleistung erst praktikabel gemacht. Während frühe Sportuhren bereits grundlegende Funktionen wie Schrittzählung und Zeitmessung mit einfachen algorithmischen Verfahren realisierten, ermöglichte die FFT eine neue Generation intelligenter Wearables für präzise Bewegungsanalyse und Klassifikationen.

3.6 Short Term Fourier Transformation

Die klassische Fourier-Transformation wandelt ein Zeitsignal in seine Frequenzbestandteile um, liefert aber keine zeitliche Information über das Auftreten der Frequenzen. Bei zeitvarianten Signalen, wie menschliche Bewegungen es sind, spielt diese Information jedoch eine entscheidende Rolle. Die Short-Term-Fourier-Transformation (STFT) ermöglicht eine Zeit-Frequenz-Analyse, indem sie das Signal in kurze, überlappende Abschnitte unterteilt und für jeden dieser Abschnitte eine separate Fourier-Transformation durchführt.

Erst wird das Signal, wie in (2.4) beschrieben, durch *Overlapping Windows* segmentiert. Anschließend wird jedes Segment mit einer Fensterfunktion zur Reduktion von Verfälschungen, die durch die abrupten Übergänge an den Rändern des Segments entstehen können, multipliziert. Dann wird auf jedes Fenster die FFT angewandt. Am Schluss erhält man eine Zeit-Frequenz-Darstellung (Spektrogramm) als Zusammenfassung der Ergebnisse.

Mathematisch wird die diskrete STFT eines Signals $x(n)$ mit einer Fensterfunktion $w(n)$ der Länge N wie folgt definiert:

$$\mathcal{X}(m, k) := \sum_{n=0}^{N-1} x(n + mH) \bar{w}(n) e^{-i2\pi\omega n} \quad (3.66)$$

$m \in \mathbb{Z}$ repräsentiert den Zeitindex (Frame-Nummer), $k \in [0 : K]$ den Frequenzindex mit der Nyquist-Frequenz $K = N/2$ und $H \in \mathbb{N}$ die Schrittgröße (hop size).

Die STFT erzeugt für jeden Zeitabschnitt (Frame) m und jede Frequenzkomponente k einen komplexen Koeffizienten $\mathcal{Y}(m, k)$, der sowohl Amplituden- als auch Phaseninformationen enthält. Um das Ergebnis zu veranschaulichen, berechnet man den Betrag dieser komplexen Koeffizienten und quadriert diesen, was das Leistungsspektrogramm ergibt:

$$\mathcal{Y}(m, k) := |\mathcal{X}(m, k)|^2 \quad (3.67)$$

Dieses Leistungsspektrogramm zeigt die Energieverteilung des Signals über Zeit (x -Achse) und Frequenz(y -Achse).

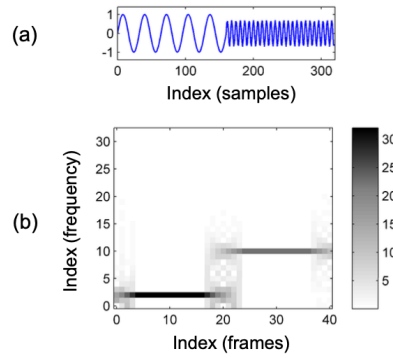


Abbildung 3.3: Leistungsspektrogramm
Quelle: [20]

Die Aufgaben der Fensterfunktion

Die Wahl der Fensterfunktion $w[n]$ ist entscheidend für die Qualität der STFT. Durch die Segmentierung der Daten entstehen ohne Fensterfunktion (oder mit einer rechteckigen Fensterfunktion) die abrupten Übergänge an den Segmentgrenzen unerwünschte Frequenzkomponenten im Spektrum. Geeignete Fensterfunktionen (z.B. Hanning, Hamming, Blackman) minimieren diese Effekte (Spektral Leakage) durch sanftes Ein- und Ausblenden. Weiters dient die Fensterfunktion zur Isolation der Signalabschnitte, indem sie einen begrenzten Bereich des Signals ausschneidet, um lokale spektrale Eigenschaften zu analysieren. Zusätzlich beeinflusst die Wahl der Fenstergröße direkt die Zeit- und Frequenzauflösung.

1. Schmale Fenster (weniger Samples): Bessere zeitliche Auflösung, schlechtere Frequenzauflösung

3 Theoretische Grundlagen der Fouriertransformation in der Signalverarbeitung

2. Breite Fenster (viele Samples): Bessere Frequenzauflösung, schlechtere zeitliche Auflösung

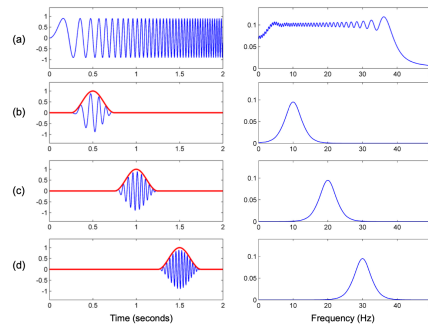


Abbildung 3.4: Anwendung von Fensterfunktionen auf ein Signal und die entsprechenden Fouriertransformationen

Quelle: [20]

3.7 Zusammenfassung

In diesem Kapitel wurden die fundamentalen Konzepte der Signalverarbeitung im Zeit- und Frequenzbereich behandelt. Ausgehend von den Grundprinzipien der Signalanalyse haben wir die mathematischen Grundlagen der Fourier-Reihe mit ihren Symmetrieeigenschaften und komplexen Darstellungsformen sowie ihrer Relevanz für Sportuhren erläutert. Mit der kontinuierlichen Fourier-Transformation und ihrer Inversen wurden die theoretischen Konzepte zur Analyse kontinuierlicher Signale vorgestellt, um die Verarbeitung von diskreten Signalen, wie es die Rohdaten der Accelerometer sind, besser zu verstehen. Die diskrete Fouriertransformation (DFT) als wichtiges Werkzeug für die digitale Signalverarbeitung wurde detailliert, speziell in Bezug auf beschränkte Signale, eingeführt. Mit der Fast Fourier Transformation (FFT) lernten wir ein recheneffizientes Verfahren kennen, das die praktische Anwendung der DFT erst ermöglicht. Zuletzt konnte mithilfe der bereits eingeführten Konzepte die Short Term Fourier Transformation, als Tool der Wahl für Sportuhren, präsentiert werden.

Die vorgestellten Transformationsmethoden bilden das mathematische Fundament für die im folgenden Kapitel behandelte Merkmalsextraktion. Während die Fourier Transformationen Signale in ihre Frequenzbestandteile zerlegen, geht es bei der Merkmalsextraktion um das Ableiten charakteristischer Eigenschaften aus den Zeit- und Frequenzdarstellungen. Diese Features sind entscheidend für die automatische Klassifikation und Erkennung von Aktivitäten in modernen Wearables.

4 Die Feature Extraktion

Sowohl in der Gesundheitsüberwachung als auch in der Sportwelt gewinnt die Analyse menschlicher Bewegung durch Beschleunigungssensoren in Sportuhren an großer Bedeutung. In der effektiven Extraktion und Verarbeitung relevanter Merkmale aus den Sensorrohdaten liegt der Schlüssel zur präzisen Aktivitätserkennung. Während im vorherigen Kapitel die mathematischen Grundlagen der Fourier-Transformation behandelt wurden, zeigt dieses Kapitel die konkrete Anwendung der Fouriertransformation sowie anderer Methoden zur Feature-Extraktion und deren Optimierung für den Einsatz in Sportuhren. Hohe Ansprüche werden an Sportuhren gestellt. Einerseits verfügen sie über begrenzte Rechenleistung und Energieressourcen, andererseits sollen sie dennoch komplexe Bewegungsmuster in Echtzeit analysieren und klassifizieren. Die Herausforderung besteht darin, aus dem kontinuierlichen Strom von Beschleunigungsdaten jene Merkmale zu extrahieren, die verschiedene Aktivitäten zuverlässig unterscheidbar machen. Dazu werden sorgfältig ausgewählte und effizient implementierte Feature-Extraktionsverfahren eingesetzt. Zuerst widmet sich das Kapitel der Merkmal-Extraktion im Zeitbereich. Anschließend werden Feature-Extraktionsverfahren mit Verwendung der Fouriertransformation in der Frequenzdomäne behandelt. Abschließend werden die Schritte der Dimensionsreduktion und der Feature-Selektion vorgestellt.

4.1 Merkmale der Zeit- vs. Frequenzdomäne

Bei der Bewegungserkennung mit Sportuhren liegt ein klassischer Kompromiss zwischen Zeit- und Frequenzdomäne vor. Wie bereits in Kapitel 3.1 erklärt, können Sensordaten im Frequenz- und im Zeitbereich analysiert werden [20]. Zeitmerkmale wie Mittelwerte, Standardabweichungen und Signal Magnitude Area bieten entscheidende Vorteile durch ihre geringe Rechenintensität, die gute zeitliche Auflösung und das unmittelbare Erkennen von Spitzen bzw. Änderungen im Signal. Dadurch ermöglichen sie Echtzeitanalysen mit minimaler Latenz [30]. Jedoch sind diese Features anfälliger für Rauschen und Störsignale, da sie Schwierigkeiten bei der Erkennung von periodischen Bewegungen und der Unterscheidung von ähnlichen Signalen haben [21]. Frequenzmerkmale wie Fourier-Koeffizienten, Spektralenergie und dominante Frequenzen erfassen hingegen periodische Muster präzise [21]. Dadurch charakterisieren sie zyklische Bewegungen wie Laufschritte besser [30]. Zusätzlich zeigt die Klassifizierung mit Frequenz-Features eine deutlich höhere Erkennungsgenauigkeit (92.38% im Frequenzbereich vs. 83.48% im Zeitbereich)

4 Die Feature Extraktion

[26]. Die FFT benötigt mehr Rechenkapazität und einen größeren Datensatz als die Zeit-Merkmale, wodurch der Energieaufwand des Endgerätes höher ist. Ein weiterer Nachteil ist, dass die Features in der Frequenzdomäne blind gegenüber Veränderungen im Datensatz sind, wodurch die Erkennung von Veränderungen der Daten erst im nächsten Fenster erkannt wird.

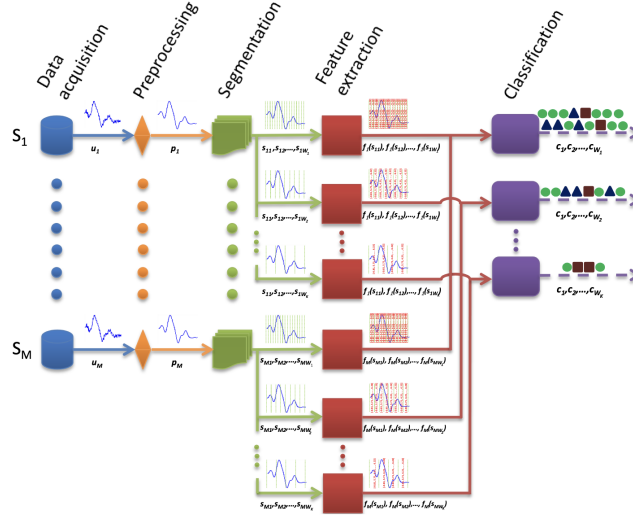


Abbildung 4.1: Datenverarbeitungsprozess

Quelle: [3]

In modernen Sportuhren-Algorithmen werden meist hybride Ansätze verfolgt [30]. Zeitdomänenmerkmale dienen als erste schnelle Aktivitätserkennung, während Frequenzdomänenmerkmale für die detaillierte Klassifikation spezifischer Bewegungsmuster eingesetzt werden. Diese Kombination nutzt die Stärken beider Domänen: Effizienz und Reaktionsschnelligkeit der Zeitdomäne mit der periodischen Mustererkennungsfähigkeit der Frequenzdomäne [21]. So kann eine präzise und energieeffiziente Bewegungserkennung erreicht werden, die sowohl grobe Aktivitätsübergänge als auch feinere Bewegungsdetails erfassen kann.

4.2 Feature-Extraktion in der Zeitdomäne

In der Zeitdomäne werden verschiedene statistische Merkmale direkt aus den Sensordaten berechnet. Diese Merkmale charakterisieren verschiedene Aspekte der Bewegungsmuster.

Nach der Segmentierung (2.4) der Rohdaten werden direkt in den einzelnen Zeitfenstern statistische Merkmale extrahiert [30]. Für jede Achse (x , y , z) des Beschleunigungssensors werden separate Merkmale bestimmt. Zusätzlich wird ein kombiniertes Maß, die

Vektormagnitude, die die Gesamtintensität der Beschleunigung unabhängig von der Richtung angibt, berechnet:

$$||v(i)|| = \sqrt{d_x(i)^2 + d_y(i)^2 + d_z(i)^2} \quad (4.1)$$

Hier repräsentiert $d_x(i)$, $d_y(i)$ und $d_z(i)$ jeweils die (i)-ten Beschleunigungswerte der x -, y - und z -Achse in jedem analysierten Fenster. Im Anschluss werden die weiteren Zeitdomänen-Merkmale für ein gegebenes Fenster $W = \{d_1, d_2, ..d_n\}$ vorgestellt.

4.2.1 Grundlegende Statistische Merkmale

Folgende statistische Merkmale können in der Zeitdomäne aus den Rohdaten bestimmt werden [29, 2, 30, 26]: Extremwerte, Arithmetisches Mittel, Standardabweichung, Signal Magnitude Area, Korrelationen und Perzentilen. Anhand der Bewegungsanalyse von Läufen wird die Bedeutung der einzelnen Features erklärt.

Extremwerte: Minimum, Maximum und Range

Die Extremwerte eines Beschleunigungssignals markieren die Bewegungsausschläge einer Aktivität [30, 10]:

$$Maximum_w = \max\{d_1, d_2, ..d_n\}$$

$$Minimum_w = \min\{d_1, d_2, ..d_n\}$$

$$Range_w = max_w - min_w$$

Maximal- und Minimalwerte geben Aufschluss über Beschleunigungs- und Bremsphasen. Laufen weist zum Beispiel höhere Betragswerte als Gehen auf, da intensivere Stoßimpulse und abruptere Bremsvorgänge auftreten. Auch der Range-Wert steigt mit (Bewegungs)-Laufintensität und hilft dadurch, zur Differenzierung zwischen Gehen (geringere Range), Joggen (mittlere Range) und Sprinten (hohe Range).

Arithmetisches Mittel

Der Mittelwert m_W der Beschleunigungswerte innerhalb eines Zeitfensters wird für jede Achse x , y und z sowie für $||v(i)||$ nach folgender Formel berechnet [30]:

$$m_W = \frac{1}{n} \sum_{i=1}^n W[i] \quad (4.2)$$

Die Mittelwerte zeigen eine hohe Korrelation mit der Intensität der Bewegung auf. Beim Laufen weist der Mittelwert charakteristische Muster auf, charakterisiert durch höhere positive Mittelwerte in der vertikalen (typischerweise y -Achse) und horizontalen (x -Achse) Komponente als beim Gehen. In der Waagrechten werden die Beschleunigungswerte in Laufrichtung und in der Senkrechten die Auf- und Ab-Bewegungen repräsentiert.

Standardabweichung

Die Standardabweichung quantifiziert die Variabilität der Beschleunigungssignale [30]:

$$\sigma_W = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (W[i] - m_W)^2} \quad (4.3)$$

Auch die Standardabweichung zeigt sich bei Laufaktivitäten signifikant höher als bei ruhigeren Bewegungen wie Gehen oder Stehen. Eine präzise Unterscheidung von Aktivitätsintensität wird durch die besonders charakteristisch erhöhten vertikalen Komponenten, die mit zunehmender Laufgeschwindigkeit ansteigen, ermöglicht.

Signal Magnitude Area

Für die Lauferkennung hat sich die Signal Magnitude Area (SMA) als besonders aussagekräftig erwiesen. SMA ist eine Kombination der absoluten Signalwerte aller drei Achsen [30, 10]:

$$SMA = \frac{1}{n} \sum_{i=1}^n (|x_i| + |y_i| + |z_i|)$$

Die SMA korreliert stark mit dem Energieverbrauch bei körperlicher Aktivität. Sie ist in einem nahezu linearen Zusammenhang mit der Laufgeschwindigkeit, wodurch sie ein ausgezeichneter Prädiktor für die Laufintensität ist.

Perzentile

Perzentile sind statistische Kennwerte, die eine Datenverteilung in hundert Teile unterteilen. Sie geben den Prozentsatz der Gesamtdaten an, die einen bestimmten Wert unterschreiten. Bei der Aktivitätsanalyse mithilfe von Sensoren ermöglichen Perzentile das Aufzeigen der Verteilung der Messwerte. Niedrige Perzentile (10., 25.) erfassen meist die Abbremsungen und ruhigeren Phasen einer Bewegung. Hohe Perzentile (75., 90.) hingegen repräsentieren die intensiveren Beschleunigungsphasen. Ein Vorteil gegenüber dem Mittelwert zeigt sich in der Robustheit der Perzentile gegenüber Ausreißern.

Die Formel für den 50. Perzentil (Median) lautet beispielsweise [30]:

$$50^{te} \text{ Perzentile} = \left(\frac{n+1}{2} \right)^{te} \text{ Term} \quad (4.4)$$

Korrelation

Die Unterscheidung von Aktivitäten in eine Richtung kann durch die Korrelation zwischen Achsen unterstützt werden [8, 11].

$$c_{x,y} = \frac{\sum_{i=1}^N (W_{xi} - \text{mean}(W_x)) * (W_{yi} - \text{mean}(W_y))}{\sqrt{\sum_{i=1}^N (W_{xi} - \text{mean}(W_x))^2 * (W_{yi} - \text{mean}(W_y))^2}} \quad (4.5)$$

4.3 Feature-Extraktion in der Frequenzdomäne

Merkmale der Zeitdomäne können direkt aus den Rohdaten gewonnen werden, im Gegensatz zu den Frequenzmerkmalen, wo der Datensatz erst bearbeitet werden muss. Nach der Segmentierung der Zeitreihendaten in überlappende Fenster (2.4) wird anschließend zur Reduzierung von Spektralleckage eine Fensterfunktion (z.B. Hamming-Fenster, siehe 3.5) auf jedes Frame angewandt. Anschließend transformieren die Sportuhren die Daten in den Frequenzbereich. Dafür wird die Short-Time Fourier Transform (STFT) durchgeführt, welche einer gefensterten DFT entspricht, die durch die FFT effizient berechnet wird:

$$\mathcal{X}(m, k) := \sum_{n=0}^{N-1} x(n + mH) \bar{w}(n) e^{-i2\pi\omega n} \quad (4.6)$$

Hierbei bezeichnet m den Index des Zeitfensters, H die Schrittweite zwischen benachbarten Fenstern (Hop-Size), $w(n)$ die Fensterfunktion, N die Länge der *Sliding Windows* und k entspricht der Frequenzkomponente. Weiters gilt $\omega = k/N$.

Zusätzlich folgt die Berechnung des Leistungsspektrums:

$$P(f) = \frac{|\mathcal{X}(f)|^2}{N} \quad (4.7)$$

wobei N der Anzahl der Datenpunkte entspricht. Nachdem die Daten in die Frequenzdomäne umgewandelt sind, werden Merkmale dieses Bereichs berechnet.

4.3.1 Merkmale aus der Frequenzdomäne

Bei der Verarbeitung von Beschleunigungssensordaten können unter anderem folgende Merkmale aus der Frequenzdomäne extrahiert werden. Jedoch werden nicht alle Merkmale stets berechnet, sondern ihre Wahl hängt von der vorgesehenen Bewegungserkennung ab [29].

1. **Spektrale Energiedichte (Power Spectral Density, PSD):** Die Verteilung der Signalleistung über verschiedene Frequenzen.

4 Die Feature Extraktion

2. **Spektrale Energie:** Die Energie in bestimmten Frequenzbereichen oder im gesamten Spektrum.
3. **Dominante Frequenz:** Die Frequenz mit der höchsten Amplitude im Spektrum.
4. **Frequenzbereichsenergie:** Energie in spezifischen Frequenzbändern (z.B. 0 – 2.5 Hz, 2.5 – 5 Hz).
5. **Spektrale Entropie:** Ein Maß für die Unordnung oder Zufälligkeit im Frequenzspektrum.
6. **Spektrale Zentroide:** Das Massenzentrum des Spektrums.
7. **Mittlere Frequenz:** Der gewichtete Durchschnitt der Frequenzen im Spektrum.
8. **Median-Frequenz:** Die Frequenz, die das Leistungsspektrum in zwei gleiche Hälften teilt.

4.3.2 Die wichtigsten Frequenzmerkmale

Die Auswahl und Implementierung der Merkmale sind stark von Faktoren wie verfügbaren Hardware-Ressourcen, Batterielaufzeit und spezifischen Anwendungsfällen abhängig. Bei der Aktivitätserkennung mithilfe von kommerziellen Sportuhren extrahiert man in den meisten Fällen drei Features der Frequenzdomäne [26, 2], die in diesem Abschnitt näher betrachtet werden. Ihre Rolle in der Bewegungsanalyse und die Informationen, die sie preisgeben, werden anhand der Aktivität *Laufen* erläutert.

Dominante Frequenz

Bei der dominanten Frequenz handelt es sich um eines der wichtigsten Frequenzmerkmale und ist der Wert der Frequenz mit der höchsten Amplitude. Sie ist besonders wertvoll, da durch relativ geringen Rechenaufwand aussagekräftige Informationen extrahiert werden können. Dadurch ist sie perfekt geeignet für ressourcenbeschränkte Geräte wie Sportuhren und wird wie folgt berechnet:

$$f_{\text{dominant}} = f[\text{argmax}(P(f))]$$

Die hohe Relevanz der dominanten Frequenz zeigt sich bei der Lauf- und Ganganalyse, da sie direkt mit der Schrittfrequenz korreliert [27]. Dieses Frequenzmerkmal bestimmt direkt über die charakteristischen Frequenzbereiche den Aktivitätstyp:

- Gehen: 1,5 – 2,0 Hz
- Joggen: 2,0 – 2,8 Hz
- Laufen: 2,8 – 3,5 Hz

- Sprinten: $> 3,5 \text{ Hz}$

Diese Frequenzen ermöglichen eine Grundunterscheidung zwischen verschiedenen Fortbewegungsarten und Intensitäten.

Bei der Analyse von Sportuhr-Daten entspricht die dominante Frequenz entweder der Arm- oder der Beinbewegungsfrequenz. Die Armbewegung entspricht meist der Hälfte der Beinfrequenz [18]:

$$f_{\text{Schritt}} \approx 2 \cdot f_{\text{Arm}} \quad (4.8)$$

Die Schrittfrequenz wird in Schritte pro Minute angegeben und entspricht in der Laufanalyse der Kadenz. Sie gibt Auskunft über den Laufstil und die Lauffeffizienz.

Außerdem ist das Berechnen der dominanten Frequenz über die Fourieranalyse das Werkzeug zur Gesamt-Schrittzahlmessung. Da mit dem Zusammenhang 4.8 und folgender Formel Sportuhren die Anzahl der Schritte bestimmen [18].

$$\text{steps}(f, t) = f[\text{Hz}] \cdot t[\text{s}] \cdot 60[\text{s/min}]$$

Spektrale Energie in spezifischen Frequenzbändern

Die spektrale Energie repräsentiert die Verteilung der Signalstärke in bestimmten Frequenzbereichen. In der Bewegungsanalyse gibt sie Aufschluss über die Intensität spezifischer Bewegungskomponenten und kann zur Beurteilung der Bewegungsqualität, Effizienz und zur Identifikation charakteristischer Muster verschiedener Aktivitäten genutzt werden.

Zur Berechnung wird von jeder Frequenzkomponente das Betragsquadrat $|\mathcal{X}(f)|^2$ berechnet, was der Leistung bei dieser Frequenz entspricht [2]:

$$|\mathcal{X}(f)|^2 = a_f^2 + b_f^2 \quad (4.9)$$

Um die spektrale Energie in einem spezifischen Frequenzband $[f_1, f_2]$ zu erhalten, wird die Summe aller Betragsquadrate in diesem Bereich berechnet [2]:

$$E[f_1, f_2] = \sum_{f=f_1}^{f_2} |\mathcal{X}(f)|^2 \text{ mit der Einheit } m^2/s^4 \quad (4.10)$$

Aus den berechneten Energiewerten in diesen Bändern können Informationen zur Klassifizierung, zur Bestimmung von Laufparametern oder anderen Bewegungsmerkmalen extrahiert werden. Die spektrale Energie zeichnet ein detailliertes Bild der Qualität und Charakteristik der Bewegung auf. Die spektrale Energie korreliert mit der Intensität der Aktivität, was durch stärkere Beschleunigungen angezeigt wird. Hohe spektrale Energie weist auf ausgeprägtere rhythmische Muster und größeren mechanischen Energieaufwand des Sporttreibenden hin. Zusätzlich lassen sich Erschütterungen und Stoßbelastungen in

4 Die Feature Extraktion

höheren Frequenzbereichen ($8 - 20 \text{ Hz}$) analysieren. Jede Aktivität erhält so eine charakteristische Energieverteilung. Jedoch sind oft Verhältnisse zwischen verschiedenen Frequenzbändern relevanter als absolute Werte. Die Bestimmung von Bewegungseffizienz wird zum Beispiel durch das Verhältnis von niedrigen und hohen Frequenzenergien ermöglicht.

Spektrale Entropie

Die spektrale Entropie quantifiziert die Unregelmäßigkeit oder Komplexität der spektralen Verteilung [2]. Gemäß der angegebenen Formel wird sie berechnet als [2]:

$$H(f) = \frac{1}{N} \sum_{i=0}^{N-1} c_i \log(c_i), c_i = \frac{\sqrt{a_i^2 + b_i^2}}{\sum_{k=0}^{N-1} \sqrt{a_k^2 + b_k^2}} \quad (4.11)$$

wobei c_i die normalisierten Koeffizienten des Leistungsspektrums darstellen, berechnet aus den Fourier-Koeffizienten a_i und b_i .

Auch die spektrale Entropie dient wichtigen Zwecken in der Bewegungsanalyse. Einerseits hilft sie bei der Unterscheidung von Aktivitäten, die eine ähnliche Leistungsspektraldichte aufweisen, aber unterschiedliche Bewegungsmuster haben. Dies ist besonders nützlich für die Klassifizierung von Aktivitäten, die energetisch ähnlich sind, sich aber in ihrer Regelmäßigkeit oder Struktur unterscheiden.

Unterscheidung ähnlicher Aktivitäten: Sie hilft, zwischen Aktivitäten zu unterscheiden, die eine ähnliche Leistungsspektraldichte aufweisen, aber unterschiedliche Bewegungsmuster haben. Dies ist besonders nützlich für die Klassifizierung von Aktivitäten, die energetisch ähnlich sind, sich aber in ihrer Regelmäßigkeit oder Struktur unterscheiden, wie Laufen und Treppensteigen. Aktivitäten mit sehr regelmäßigen, rhythmischen Bewegungen (wie Gehen) weisen eine niedrigere Entropie auf, im Gegensatz zu komplexeren, unregelmäßigeren Bewegungen (wie bestimmte Sportarten oder Alltagsaktivitäten). Daher gibt die Entropie Aufschluss über die Komplexität der Bewegungen. Zusätzlich dient die Entropie der Ergänzung der Zeitmerkmale wie Mittelwert, Standardabweichung oder SMA und verbessert so die Gesamtgenauigkeit der Aktivitätserkennung.

Zusammenfassend gibt die dominante Frequenz an, *wie oft* etwas passiert, während die spektrale Energie Aussage darüber trifft, *wie stark* Bewegungen in diesem Frequenzband ausgeführt wurden. Die spektrale Entropie hingegen beschreibt, *wie gleichmäßig* die Energie über verschiedene Frequenzen verteilt ist, und ermöglicht dadurch die Unterscheidung zwischen regelmäßigen und unregelmäßigen Bewegungsmustern.

4.4 Feature Wahl - Dimensionsreduzierung

Nach dem ersten Schritt der Feature-Extraktion erhalten moderne Sportuhren einen hochdimensionalen Merkmal-Vektor. Dieser Vektor besteht aus der Vielzahl an Zeit-

und Frequenzdomänen-Features, die aus den Rohdaten der Beschleunigungssensoren extrahiert wurden [29]. Für jedes Datenfenster berechnet die Uhr einen hochdimensionalen Feature-Vektor. Durch Anwendung eines mathematischen Werkzeugs zur Reduzierung der Dimensionen wird der hochdimensionale Feature-Raum auf die wesentlichen Informationen reduziert [2]. Dieser Schritt dient zur Wahl der nützlichen Merkmale unter der Verwendung eines *Feature Selection Algorithmus* [8]. So wird eine effiziente und genaue Erkennung von Aktivitäten ermöglicht, ohne die begrenzten Ressourcen der Geräte zu überlasten. Aufgrund der Reduktion der Dimensionen arbeiten Algorithmen zur Aktivitätserkennung effizienter unter der Verwendung von weniger Rechenkapazität und weniger Energie. Dieser Schritt ermöglicht die Echtzeitverarbeitung der Daten, da die Berechnungen in der Uhr durch den geringeren Datensatz effizienter und schneller sind. Zusätzlich dient das Reduzieren dazu, die wesentlichen Unterscheidungsmerkmale auszusuchen und verbessert dadurch die Klassifizierungsgenauigkeit [26].

Die Principal Component Analysis (PCA) scheint die häufigst verwendete Methode, die ursprünglichen, oft korrelierten Features in neue, unkorrelierte Hauptkomponenten zu transformieren [26, 23]. Diese werden nach ihrer Varianz angeordnet, wobei Komponenten mit sehr geringer Varianz weggelassen werden [2].

Die PCA reduziert die Dimension des Datensatzes in folgenden Schritten:

1. Normalisierung der Daten durch Subtraktion des Mittelwerts
2. Berechnung der Kovarianzmatrix
3. Berechnung der Eigenvektoren und Eigenwerte der Kovarianzmatrix
4. Auswahl der Komponenten und Bildung eines neuen Feature-Vektors
5. Ableitung eines neuen Datensets

Das praktische Endergebnis ist ein kompakter, informationsreicher Datensatz, der die wesentlichen Aspekte der Bewegungsmuster erfasst. Rauschhafte Informationen wurden während der PCA entfernt. Bei einer typischen Anwendung zur Aktivitätserkennung können so die ursprünglichen 60+ Merkmale [8] auf wenige Hauptkomponenten (z.B. $k = 5$) reduziert werden [25], die oft 95 % der Varianz der Daten erklären. Das neue Datenset wird an die Klassifikationsalgorithmen weitergeleitet, die bestimmen, welche Aktivität der Benutzer gerade ausführt.

Die Implementierung der Dimensionsreduktion macht den Unterschied zwischen mehrstündiger und ganztägiger Akkulaufzeit von Sportuhren bei kontinuierlicher Aktivitätserkennung [25].

4.5 Zusammenfassung

Durch die Feature-Extraktion wird ein Vektor gebildet, der Informationen über die durchgeführte Bewegung enthält. Die Gewinnung der relevanten Merkmale aus Sensordaten

4 Die Feature Extraktion

wird in zwei verschiedenen Domänen, dem Zeit- und Frequenzbereich, durchgeführt. In der Zeitdomäne werden statistische Kenngrößen wie Extremwerte, Mittelwerte, Standardabweichungen und die Signal Magnitude Area direkt aus den Rohdaten berechnet. Diese Merkmale zeichnen sich durch geringe Rechenintensität und hohe zeitliche Auflösung aus, sind jedoch anfälliger für Störsignale. Die Frequenzdomäne hingegen nutzt die Fouriertransformation, um periodische Bewegungsmuster präziser zu erfassen. Dominante Frequenzen, spektrale Energie und spektrale Entropie ermöglichen eine genauere Klassifizierung zyklischer Bewegungen trotz höherem Rechenaufwand. Die Merkmale der zwei Bereiche ergänzen sich auf optimale Weise: Zeitdomänen-Features bieten Effizienz und Schnelligkeit, während Frequenzdomänen-Merkmale periodische Muster genauer erfassen. Der hybride Ansatz moderner Sportuhren-Algorithmen nutzt diese komplementären Stärken und erreicht durch gezielte Dimensionsreduktion eine beeindruckende Balance zwischen Erkennungsgenauigkeit und Ressourceneffizienz. Diese sorgfältige Optimierung ermöglicht es, aus dem kontinuierlichen Strom von Beschleunigungsdaten die charakteristischen Merkmale zur Unterscheidung von Aktivitäten zu extrahieren.

5 Klassifizierung von Bewegungen

Die Klassifizierung von Bewegungen stellt den finalen und entscheidenden Schritt in der Bewegungserkennung durch Beschleunigungsdaten dar. Nach der Erfassung, Vorverarbeitung der Rohdaten und der Extraktion der relevanten Merkmale werden im letzten Schritt die aufbereiteten Daten konkreten Bewegungsmustern und Aktivitäten zugeordnet. Die numerischen Feature-Vektoren werden so in semantisch bedeutsame Informationen wie *Gehen*, *Laufen* oder *Schwimmen* umgewandelt.

Die hohe Variabilität menschlicher Bewegungen stellt eine Herausforderung in der Bewegungsklassifizierung dar. Selbst dieselbe Aktivität wird von verschiedenen Personen unterschiedlich ausgeführt und erzeugt unterschiedliche Beschleunigungsmuster. Zusätzlich limitieren die begrenzten Rechenressourcen und Energiebeschränkungen von Sportuhren das Arbeiten der Klassifizierungsalgorithmen. Um diese Herausforderungen zu meistern und das präzise, schnelle Arbeiten der Algorithmen zu ermöglichen, ist eine sorgfältige Auswahl des Werkzeugs notwendig.

In diesem Kapitel werden die wichtigsten Klassifizierungsalgorithmen, die in modernen Sportuhren zum Einsatz kommen oder kommen könnten, betrachtet. Zuerst werden die verschiedenen Formen des maschinellen Lernens, die in HAR-Systemen eingesetzt werden, vorgestellt. Dann werden überwachte Verfahren des maschinellen Lernens wie Support Vector Machines, Random Forests und k-Nearest Neighbors, die sich durch ihre Effizienz und Robustheit auszeichnen, betrachtet. Anschließend werden fortgeschrittenere Methoden wie Hidden Markov Models untersucht, die zunehmend in High-End-Geräten implementiert werden.

5.1 Grundlagen zur maschinellen Erkennung menschlicher Aktivitäten

Um aus Sensordaten interpretierbare Aktivitäten zu gewinnen, setzen HAR-Systeme wie Sportuhren verschiedene Techniken des maschinellen Lernens ein. Die Klassifikation beginnt mit Rohdaten der Sensoren (z.B. Beschleunigungsmesser, Gyroskope), die in definierte Zeitfenster segmentiert werden (siehe Abschnitt 2.2). Aus diesen Zeitfenstern werden charakteristische Merkmale extrahiert und zu Feature-Vektoren zusammengefasst (siehe Abschnitt 4). Diese Merkmalsvektoren bilden die Objekte des Trainingsdatensatzes [23]. Die Qualität der Aktivitätserkennung hängt maßgeblich von der Repräsentativität der Trainingsdaten ab. Jeder Vektor im Datensatz kann mit einer Aktivitätsklasse

(z.B. *Gehen, Laufen, Sitzen*) versehen werden (beschriftete Daten) oder unbeschriftet vorliegen.

5.1.1 Lernansätze

Drei Lernansätze werden bei der Entwicklung von HAR-Systemen unterschieden: *Überwachtes Lernen* (Supervised Learning), Unüberwachtes Lernen (Unsupervised Learning) und Hybridansätze (Semi-supervised Learning) [2]. Beim *Überwachten Lernen* trainiert man ein Modell mit beschrifteten Datensätzen. *Unüberwachtes Lernen* arbeitet hingegen mit unbeschrifteten Datensätzen, und diese Modelle erlernen selbstständig Muster und Strukturen zu erkennen. Beim Hybridansatz wird ein Modell teilweise durch *Überwachtes Lernen*, teilweise durch *Unüberwachtes Lernen* trainiert.

Supervised Learning

Bei der sensorbasierten Bewegungserkennung wird primär mit *Supervised Learning* gearbeitet, da die gewünschte Ausgabe stärker kontrolliert werden kann und *Unsupervised Learning* komplexere Modelle benötigt [8, 24]. Supervised Learning bildet die Grundlage für die automatische Aktivitätserkennung bei modernen Sportuhren. Bevor ein maschinelles Lernmodell auf einer Sportuhr implementiert wird, muss es mit Trainingsdaten gefüttert und trainiert werden [29]. Bei den Trainingsdaten handelt es sich um Sensordaten von Probanden, die verschiedene Aktivitäten ausführen, die im Anschluss beschriftet werden. Die erfassten kontinuierlichen Beschleunigungswerte werden durch Sliding-Window-Techniken segmentiert. Durch Vorverarbeitung und Feature-Extraktion, bei der sowohl zeitbasierte als auch frequenzbasierte Merkmale berücksichtigt werden, wird ein Merkmalsvektor pro Fenster gewonnen, der mit einer Aktivitätsklasse beschriftet wird.

Die Beschriftung (Labeling) dieser Daten ist ein zentraler Schritt im Training des Modells: Systeme speichern Sensordaten, während der Sammelprozess (z.B. das Laufen, Radfahren oder Schwimmen einer Person) überwacht wird und manuell Aktivitätslabels und Zeitstempel erfasst werden [29]. So wird jede Datenprobe einer Aktivitätsklasse zugeordnet und auf dem Server gespeichert. Der Algorithmus – sei es ein Random Forest, eine Support Vector Machine oder ein neuronales Netz – lernt durch wiederholtes Training die komplexen Zusammenhänge zwischen den Merkmalen und den zugehörigen Beschriftungen [19]. Die Qualität des Modells wird dabei durch Kreuzvalidierung oder Hold-out-Validierung überprüft, um Überanpassung zu vermeiden. Dieser Trainingsprozess findet auf leistungsstarken Computern statt. Nach Abschluss des Trainings wird das statische Modell auf die Sportuhr übertragen. Das trainierte Modell ermöglicht Funktionen wie automatische Aktivitätserkennung, Schrittzählung und sportartspezifische Leistungsanalysen. Anders als bei kontinuierlichem Lernen findet auf der Uhr selbst in der Regel kein weiteres Training statt – die Modelle bleiben aufgrund von Ressourcenbeschränkungen

und Qualitätssicherungsaspekten unverändert. Die Sportuhr kann neue, ähnliche Sensordaten selbstständig analysieren und in Echtzeit entscheiden, welche Aktivität der Träger gerade ausführt.

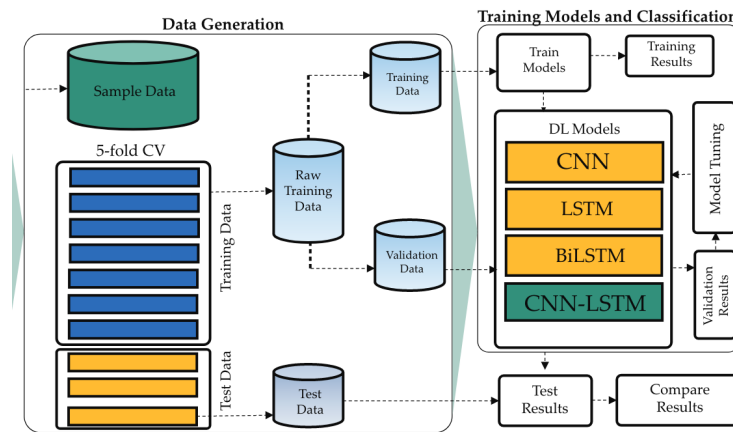


Abbildung 5.1: Klassifizierung
Quelle: [19]

Semi-supervised Learning

Nachdem das Beschriften von Datensätzen jedoch sehr zeitintensiv und kostspielig sein kann und Experten erfordert, werden für modernere Geräte bereits Wege zur Reduzierung und Automatisierung des Beschriftungsprozesses gesucht [23]. Bei dem Hybridansatz werden vergleichsweise wenige gelabelte Daten mit einer großen Menge ungelabelter Daten kombiniert [29]. Die beschrifteten Daten liefern die Grundstruktur für die Aktivitätserkennung, während die ungelabelten Daten helfen, die Variabilität und Vielfalt der Bewegungsmuster besser zu erfassen. Die Vorteile von solchen Modellen sind ein reduzierter Bedarf an manuell gelabelten Daten und eine individuelle Anpassung an den Nutzenden.

5.2 Klassifikation - Algorithmen

Der Trend bei Machine Learning Algorithmen in Sportuhren zeigt eine deutliche Verschiebung von traditionellen Methoden hin zu Deep Learning Ansätzen. Anfänglich wurden klassische Algorithmen wie Support Vector Machines, k-nearest Neighbors und Decision Trees für die Aktivitätserkennung eingesetzt. Diese relativ einfachen Verfahren erforderten manuelle Merkmalsextraktion und hatten Schwierigkeiten mit komplexen Bewegungsmustern [28]. Mit zunehmender Rechenleistung selbst in kompakten Wearables hat

sich der Fokus in der Forschung auf Deep Learning Architekturen verlagert. Convolutional Neural Networks (CNNs) haben sich als besonders effektiv für die Erkennung von Mustern in Sensordaten erwiesen, während Recurrent Neural Networks (RNNs) und insbesondere Long Short-Term Memory (LSTM) Netzwerke hervorragend für die Analyse von Zeitreihen geeignet sind, wie sie bei Bewegungsdaten auftreten [6].

Moderne Sportuhren nutzen zunehmend hybride Ansätze, die verschiedene maschinelle Lernverfahren und traditionelle Methoden kombinieren, um sowohl räumliche als auch zeitliche Aspekte der Bewegungsdaten zu erfassen. In der Forschung werden CNN-LSTM Kombinationen untersucht, wobei CNNs Merkmale aus den Sensordaten extrahieren und LSTMs die zeitlichen Zusammenhänge modellieren [14]. Während solche Deep Learning-Ansätze hauptsächlich in Laborstudien und High-End-Geräten zum Einsatz kommen, verwenden die meisten Consumer-Sportuhren weiterhin optimierte traditionelle Algorithmen wie Random Forest in Kombination mit spezialisierten Verfahren für bestimmte Aktivitäten. Diese Entwicklung ermöglicht eine präzisere Aktivitätserkennung, verbesserte Bewegungsanalysen und eine bessere Personalisierung der Trainingsempfehlungen, was zur kontinuierlichen Verbesserung der Funktionalität und des Nutzens von Sportuhren beiträgt. Im folgenden Abschnitt werden erst traditionelle Klassifikatoren und anschließend die moderneren Ansätze, die in Sportuhren eingesetzt sind oder werden könnten, vorgestellt.

5.2.1 Support Vector Maschine

Support Vector Machines (SVMs) sind binäre Klassifikatoren aus dem Bereich des maschinellen Lernens, die von Vapnik entwickelt wurden [2]. Ihre mathematische Grundlage stammt aus der statistischen Lerntheorie und wird durch Attal u. a. [2] und Rosati, Balestra und Knaflitz [21] wie folgt erklärt.

SVMs suchen in den Trainingsdaten nach der optimalen Trennlinie (oder Hyperebene in mehrdimensionalen Räumen), die verschiedene Aktivitätsklassen wie *Laufen* und *Nicht-Laufen* voneinander separiert. Dieser Klassifikator findet jene Trennlinie, die den maximalen Abstand (Margin) zu den nächstgelegenen Datenpunkten von beiden Klassen aufweist. Die gefundenen Punkte nennt man *Support Vektoren*. Bei der Aktivitätserkennung setzt die SVM den *Kernel-Trick* ein: Die Daten werden implizit in einen höherdimensionalen Raum transformiert, in dem eine lineare Trennung möglich wird, ohne das Durchführen der Berechnung in der höheren Dimension. Eine nichtlineare Kernel-Funktion ermöglicht diesen Schritt.

Da SVMs ursprünglich nur zwei Klassen unterscheiden können, werden für die Erkennung mehrerer Aktivitäten wie Laufen, Schwimmen oder Radfahren entweder mehrere SVMs kombiniert oder hierarchische Entscheidungsstrukturen aufgebaut. Folgende zwei Strategien kommen in diesem Fall zum Einsatz:

- Eine Klasse gegen alle anderen Klassen (One-vs-All)

- Oder jede Klasse gegen jede andere Klasse (One-vs-One)

In der Studie von Rosati, Balestra und Knaflitz [21] wurden beispielsweise 21 SVMs paarweise kombiniert, um 7 verschiedene Aktivitäten zu differenzieren.

5.2.2 K-Nearest Neighbors

Der k-nearest Neighbors (KNN) Algorithmus ist ein überwachtes Klassifikationsverfahren, das neue Elemente durch ihren Abstand zu den verschiedenen Elementen des Trainingssets erkennt [2]. KNN kommt ohne komplexen Lernprozess aus [2], was für die Implementierung in Sportuhren mit begrenzter Rechenleistung ein Vorteil ist. Wenn eine neue Bewegung erkannt werden soll, vergleicht die Uhr die aktuellen Sensordaten (z.B. Beschleunigungswerte) mit den gespeicherten Referenzdaten und berechnet den euklidischen Abstand zu jedem dieser Bewegungsmuster, erklärt Attal u. a. [2]. Die k ähnlichsten Referenzmuster, die den geringsten Abstand haben, werden in Betracht gezogen. Die am häufigsten vorkommende Aktivitätsklasse unter diesen K Nachbarn wird als Ergebnis ausgegeben. Diese Mehrheitsentscheidung ermöglicht die Unterscheidung verschiedener Aktivitäten wie Gehen, Laufen, Radfahren oder Schwimmen, wobei die Sensordaten kontinuierlich mit gespeicherten Referenzmustern verglichen werden. Ein wichtiger Parameter ist die Wahl von K , wobei häufig $k = \sqrt{N}$ (mit N als Anzahl der Trainingsdaten) als Ausgangswert verwendet wird [21]. In Sportuhren hat sich dieser Ansatz besonders bewährt, da Forschungsstudien sowohl mit Zeit- als auch mit Frequenzmerkmalen hohe Genauigkeiten bei der Erkennung verschiedener körperlicher Aktivitäten nachgewiesen haben [2]. Der Hauptnachteil ist der steigende Rechenaufwand bei wachsender Datenmenge, was bei batteriebetriebenen Geräten wie Sportuhren berücksichtigt werden muss.

5.2.3 Random Forest

Der Random Forest (RF) Classifier ist ein fortschrittlicher überwachter Klassifikationsalgorithmus, der auf dem Prinzip der Ensemblemethode basiert [2]. Er besteht aus vielen einzelnen Entscheidungsbäumen, die jeweils unabhängig voneinander trainiert werden. Für jeden Baum wird eine zufällige Teilmenge der Trainingsdaten mit Zurücklegen gezogen (Bootstrap-Stichprobe) [2, 30]. Das bedeutet, dass einige Datenpunkte mehrfach vorkommen können, während andere gar nicht berücksichtigt werden. Bei jedem Verzweigungspunkt eines Baumes wird nur eine zufällige Teilmenge der verfügbaren Merkmale betrachtet. Dies fördert die Diversität zwischen den Bäumen und verhindert, dass dominante Merkmale in allen Bäumen verwendet werden [2]. In Sportuhren verarbeitet der RF kontinuierlich die erfassten Bewegungsdaten und ordnet sie durch Mehrheitsentscheidung aller Bäume einer spezifischen Aktivität zu – sei es Laufen, Schwimmen, Radfahren oder sogar komplexere Bewegungsabläufe wie verschiedene Sportübungen. Nach Attal u. a. [2], San-Segundo u. a. [22] und Yazdansebas u. a. [30] erzielt RF dabei eine höhere Genauigkeit als andere Klassifikatoren. Jedoch erst bei der Verwendung von ausreichend gelabelte

5 Klassifizierung von Bewegungen

Trainingsdaten erzielt dieser Klassifikationsalgorithmus die besten Ergebnisse [2]. Trotz des höheren Trainingsaufwands macht die hohe Genauigkeit dieses Klassifikators, auch bei komplexeren Bewegungen, zur bevorzugten Wahl für Aktivitätserkennungssysteme in Sportuhren. Besonders die Erkennungsfähigkeit subtiler Unterschiede in Bewegungsmustern ermöglicht es Sportuhren, zwischen ähnlichen Aktivitäten wie Joggen und Laufen oder verschiedenen Schwimmstilen zu unterscheiden.

5.2.4 Markov Chains and Hidden Markov Models

Hidden Markov Models (HMMs) und Gaussian Mixture Models (GMMs) sind statistische Modelle, die in Sportuhren zur Aktivitätserkennung eingesetzt werden können [2]. In Sportuhren verarbeiten HMMs die Sensordaten als beobachtete Sequenzen und ordnen ihnen verborgene Zustände (Bewegungen wie Gehen, Laufen, Schwimmen) zu [2]. Hierbei wird der Viterbi-Algorithmus zur Berechnung der wahrscheinlichsten Aktivitätssequenz genutzt. GMMs dienen als Emissionsmodelle innerhalb der HMMs. Sie beschreiben die Wahrscheinlichkeitsverteilung der Sensordaten für jede Aktivität, wodurch komplexe Bewegungsmuster erfasst werden können [22]. Diese Kombination eignet sich besonders für die sequentielle Natur menschlicher Bewegungen, da Übergänge zwischen Aktivitäten modelliert werden können, beispielsweise wie wahrscheinlich ein Wechsel vom Gehen zum Laufen ist [22]. Beide Modelle können auch für unüberwachtes Lernen genutzt werden, was für die Erkennung neuer Aktivitätsmuster ohne vorherige Kennzeichnung von Vorteil ist [22].

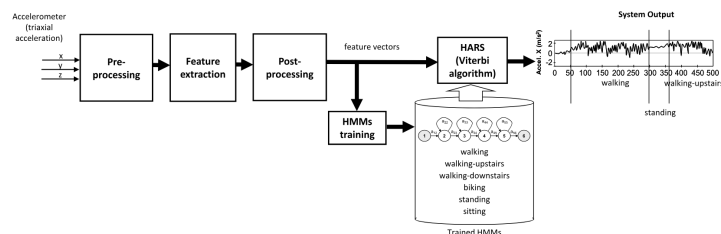


Abbildung 5.2: HMM Architektur

Quelle: [22]

5.2.5 Neuronale Netzwerke und Deep Learning Architektur

Durch die Fortschritte von Hardware, Software und Maschine Learning eröffnen sich im Bereich von HAR und somit von Sportuhren neue Möglichkeiten. Sedaghati, Ardebili und Ghaffari [23] zeigt auf, dass mehr und mehr *Neuronale Netzwerke* zur Bestimmung von Aktivitäten erforscht und eingesetzt werden. Die Fortschritte in der Rechenleistung und Energieeffizienz haben den Einsatz komplexerer Algorithmen wie neuronale Netze auch in ressourcenbeschränkten Geräten wie Wearables ermöglicht. Verschiedene Deep Learning

Architekturen werden im HAR-Kontext erforscht, wobei sich die Forschung primär auf Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), flache Feed-Forward-Netzwerke (MLP) [30] sowie hybride Ansätze wie CNN-LSTM-Architekturen konzentriert [23, 6]. Einige dieser Ansätze finden bereits Anwendung in kommerziellen Sportuhren.

Neuronale Netze zeichnen sich durch ihre automatisierte Merkmalsextraktion aus, erklärt Chen u. a. [6]. Anders als bei traditionellen Ansätzen müssen keine manuellen Feature-Engineering-Entscheidungen getroffen werden, sondern das Modell erlernt selbstständig, welche Features relevant sind. Dadurch gestaltet sich der Entwicklungsprozess effizienter [6]. Die Netzwerke lernen eigenständig eine Hierarchie von Merkmalen - von einfachen Mustern in unteren Schichten bis hin zu komplexen, abstrakten Konzepten in höheren Schichten. Diese Vorgehensweise ermöglicht ihnen das Finden von Features, die dem Menschen unbekannt sind. Zusätzlich erfassen sie durch diese Schichtarchitektur komplexere und subtilere Bewegungsmuster als klassische Ansätze [23]. Ihre Genauigkeit bei komplexeren Aktivitätserkennungen wie Tennisschläge oder Schwimmszüge sticht heraus. Weiters besitzen diese moderneren Algorithmen eine bessere Fähigkeit, sich an den Sporttreibenden oder an neue Aktivitätstypen anzupassen durch Techniken wie *Transfer Learning* [6].

Grundkonzept Neuronaler Netzwerke

Neuronale Netze sind mathematische Modelle, deren Aufbau vom menschlichen Gehirn inspiriert wurde. Wie im Gehirn Milliarden von Nervenzellen (Neuronen) zusammenarbeiten, bestehen neuronale Netze aus Schichten von miteinander verbundenen *künstlichen Neuronen*. Man kann sich die Schichten wie Teammitglieder vorstellen, die miteinander arbeiten. Folgendermaßen sind neuronale Netze aufgebaut [22, 8]:

- Eingabeschicht (Input Layer): Diese Schicht nimmt die Rohdaten der Sensoren auf und dient als Schnittstelle zwischen Rohdaten und weiteren Verarbeitungsschichten. Bei einer Sportuhr handelt es sich um Daten von Beschleunigungs- und anderen Sensoren sowie Gyroskopen.
- Verborgene Schichten (Hidden Layers): Diese Schichten verarbeiten die Informationen weiter und extrahieren zunehmend komplexere Merkmale. Je mehr Schichten, desto *tiefer* ist das neuronale Netz. Von der Architektur leitet sich der Begriff *Deep Learning* ab. In der ersten verborgenen Schicht werden einfache Muster erkannt (z.B. eine kurze Aufwärtsbewegung). In tieferen Schichten werden komplexere Muster analysiert (z.B. eine Kombination von Bewegungen, die einen Tennisaufschlag ausmachen).
- Ausgabeschicht (Output Layer): Diese letzte Schicht liefert das Ergebnis, bei Sportuhren die erkannte Aktivität. Sie könnte zum Beispiel acht verschiedene Neuronen enthalten, die jeweils für eine bestimmte Sportart stehen (Laufen, Schwimmen, Radfahren usw.).

5 Klassifizierung von Bewegungen

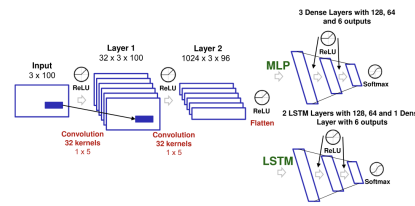


Abbildung 5.3: Deep Learning Architektur
Quelle: [22]

Convolutional Neural Networks

Convolutional Neural Networks (CNNs) sind spezielle neuronale Netzwerke, die ursprünglich für die Bildverarbeitung entwickelt wurden, aber auch bei der Aktivitätserkennung erfolgreich eingesetzt werden. Sie zeichnen sich durch ihre Fähigkeit aus, lokale Muster in Daten zu erkennen und Gewichte zu teilen [8]. CNNs bestehen aus mehreren Schichten. Dazu gehören Faltungsschichten (Convolutional Layers), die mittels Neuronen und tieferer Schichten Eingabemerkmale lernen und optimieren [8]. Jedes Neuron fungiert als rezeptives Feld, das es mit vorherigen Schichten verbindet [8]. Bei der Aktivitätserkennung können Zeit-Frequenz-Spektral-Merkmale von Sensorsignalen durch CNNs verarbeitet werden, um alltägliche Aktivitäten wie Gehen und Stehen zu erkennen [6].

Um die Funktionsweise von CNNs zu verstehen, ist es hilfreich, sie mit einem Bilderkennungssystem zu vergleichen. Ein CNN betrachtet ein Bild, nicht indem es jedes einzelne Pixel analysiert, sondern indem es nach charakteristischen Eigenschaften eines Hundes wie Ohren, Schwanz oder Schnauze sucht. In Sportuhren ist die Funktionsweise eines CNNs ähnlich. In dem Datenstrom des Beschleunigungssensors sucht das CNN nach charakteristischen Bewegungsmustern, die spezifische Bewegungen wie den Kraulschwimmzug oder Laufen kennzeichnen [22].

Recurrent Neural Networks und Long Short-Term Memory

RNNs sind wie ein Mensch mit Gedächtnis. Sie sind spezialisierte neuronale Netzwerke, die für die Verarbeitung sequentieller Daten entwickelt wurden [8]. Im Gegensatz zu CNNs, die Daten unabhängig voneinander betrachten, können sich RNNs an vorherige Daten *erinnern*, da sie über rekurrente Verbindungen (Rückkopplungsschleifen) verfügen [8]. Diese Struktur ermöglicht Informationen über Zeitschritte hinweg zu speichern und zu nutzen. Dies macht sie besonders wertvoll für die Aktivitätserkennung, da menschliche Bewegungen zeitliche Abhängigkeiten aufweisen.

Bei der Aktivitätserkennung werden RNNs eingesetzt, um zeitliche Muster in kontinuierlichen Sensordatenströmen zu erfassen [8]. Jedoch vergessen traditionelle RNNs schnell wichtige Informationen oder überbewerten unwichtige Details. Aufgrund dessen wurden

verbesserte Programme namens LSTMs entwickelt [6]. Diese funktionieren wie ein kluges Notizbuch, das wichtige Informationen speichert und unwichtige vergisst, weshalb sie Bewegungsmuster viel besser erkennen können. Eine effektive LSTM-basierte RNN-Architektur benötigt mindestens zwei Schichten für die Verarbeitung sequentieller Daten. Moderne Implementierungen nutzen oft Varianten wie bidirektionale LSTMs, die Informationen sowohl aus der Vergangenheit als auch der Zukunft verarbeiten können, was sich positiv auf die Erkennungsgenauigkeit auswirkt [6].

Hybride Architekturen

CNN-LSTM Hybridmodelle vereinen die Stärken beider Netzwerktypen in einer gemeinsamen Architektur [14]: Die CNN-Komponente extrahiert zunächst räumliche und lokale Merkmale zur Erkennung von Mustern in den Sensordaten. Anschließend werden diese extrahierten Merkmale an LSTM-Schichten weitergegeben, die zeitliche Abhängigkeiten und langfristige Muster in den Sequenzdaten erkennen können [14]. Diese Kombination ist besonders leistungsfähig bei der Aktivitätserkennung, da menschliche Aktivitäten sowohl räumliche Muster (z.B. spezifische Bewegungsabläufe) als auch zeitliche Abhängigkeiten (z.B. die Reihenfolge von Teilbewegungen) aufweisen. Durch die Verwendung eines *time distributed wrapper* kann das Modell den CNN-Teil effizient für jede Teilsequenz wiederverwenden, was die Berechnungseffizienz verbessert. Daraus folgt, dass die Hybridarchitektur eine präzisere Erkennung komplexer Aktivitätssequenzen als CNN oder LSTM allein ermöglicht [14].

5.3 Zusammenfassung

Die Entwicklung der Klassifikationsalgorithmen in Sportuhren spiegelt den technologischen Fortschritt im Bereich des maschinellen Lernens wider. Die Aktivitätserkennung hat sich von einfachen, ressourcenschonenden Methoden hin zu fortgeschrittenen maschinellen Lernverfahren entwickelt, die eine präzisere Bewegungsanalyse ermöglichen. Während frühe Systeme eine Erkennungsgenauigkeit von 75-85% bei 4-6 Grundaktivitäten erreichten, erzielen moderne Algorithmen Erkennungsraten von 85-95% bei 10-15 verschiedenen Aktivitätstypen [23, 26, 2, 21]. In der Anfangsphase von etwa 2000 bis 2010 dominierten traditionelle Algorithmen wie Support Vector Machines (SVM) und K-Nearest Neighbors (KNN). Diese ersten Klassifikatoren waren speziell für die begrenzten Ressourcen erster Smartwatches ausgelegt. Die Weiterentwicklungsphase von 2010 bis 2020 brachte wichtige Verbesserungen mit sich. Random Forest etablierte sich als robuster Ensemble-Ansatz und Hidden Markov Models ermöglichten die Erkennung sequenzieller Bewegungsmuster [2]. Parallel dazu wurden erste einfache neuronale Netzwerke in der Forschung entwickelt.

Seit etwa 2015 entstehen zunehmend fortgeschrittene Ansätze in der Bewegungsklassifikation. In der Forschung haben sich Deep Learning-Architekturen wie Convolutional

5 Klassifizierung von Bewegungen

Neural Networks (CNN) und Long Short-Term Memory (LSTM) Netzwerke als vielversprechend erwiesen, wobei diese hauptsächlich in kontrollierten Studien Genauigkeiten von über 90% bei 15+ Aktivitäten erreichen. Hybride Architekturen wie CNN-LSTM zeigen in Laborumgebungen besonders gute Ergebnisse [23].

Der Übergang zu moderneren Ansätzen wurde durch drei Faktoren ermöglicht: die steigende Rechenleistung mobiler Prozessoren, verbesserte Energieeffizienz und den Bedarf nach genauerer Erkennung komplexer Bewegungsmuster. Dies ermöglicht es aktuellen Sportuhren, zwischen verschiedenen Sportarten und teilweise sogar spezifischen Bewegungsmustern zu differenzieren.

Heutzutage setzen Hersteller von Sportuhren überwiegend auf proprietäre Kombinationen aus traditionellen und modernen Algorithmen. Diese Hybrid-Systeme nutzen beispielsweise die Effizienz von Random Forests oder SVMs für grundlegende Aktivitätserkennung und ergänzen diese bei Bedarf um fortgeschrittene Verfahren für spezifische Anwendungen. Durch die intelligente Kombination verschiedener Ansätze können diese Systeme den Kompromiss zwischen Genauigkeit, Recheneffizienz und Batterielebensdauer optimal ausbalancieren. Der erhöhte Rechenaufwand fortgeschrittener Algorithmen stellt jedoch weiterhin eine Herausforderung für die Batterielaufzeit dar. Moderne Systeme lösen dieses Problem durch adaptive Algorithmen, die nur bei Bedarf auf ressourcenintensive Modelle umschalten, während für einfache Aktivitäten effizientere Klassifikatoren ausreichen. Bei günstigeren Sportuhrenmodellen werden aufgrund begrenzter Rechenkapazität weiterhin hauptsächlich traditionelle ML-Algorithmen implementiert. Frühe Systeme konnten lediglich zwischen grundlegenden Aktivitäten wie Gehen und Laufen unterscheiden und einfache Metriken wie Schrittzahl und Tempo berechnen. Moderne Algorithmen auf High-End-Geräten ermöglichen dagegen erweiterte Analysen. Sie können verschiedene Laufstile unterscheiden, Schrittfrequenz und -länge schätzen sowie grundlegende biomechanische Parameter wie Bodenkontaktzeit approximieren.

6 Der Weg der Rohdaten bis zur Klassifizierung - Am Beispiel Laufen

In den vorangegangenen Kapiteln wurden die Verarbeitungsschritte ausführlich vorgestellt, die von der Sportuhr aufgenommenen Beschleunigungsdaten bis zur Klassifizierung und Bewegungsanalyse durchlaufen. Der Sporttreibende wählt auf der Uhr ein Workout aus. Das analoge Signal der Bewegung wird dann durch das Abtasten des Accelerometers mit einer Rate von üblicherweise 50-200 Hz [24] in ein digitales Signal umgewandelt. Anschließend wird der kontinuierliche Datenstrom vorverarbeitet (gefiltert und normalisiert) und in überlappende Fenster (meist 1-10 Sekunden) segmentiert, bevor Merkmale in Zeit- und Frequenzdomäne extrahiert werden. Die Klassifizierung erfolgt durch einen Klassifikationsalgorithmus (z.B. Random Forest, neuronale Netze), der mit den Feature-Vektoren jedes Fensters gespeist wird.



Abbildung 6.1: Garmin Sportmodusauswahl

Quelle: Eigene Darstellung

Um das Verständnis dieses komplexen Prozesses zu erleichtern, wird in diesem Kapitel der vollständige Verarbeitungsweg anhand eines durchgerechneten Beispiels für die Laufaktivität durchgeführt und veranschaulicht.

6.1 Methodik

Nachdem die Möglichkeit, auf Rohdaten zuzugreifen, bei den meisten Sportuhren nicht besteht, wird mit dem Beschleunigungssensor **WT901BLECL5.0** von *Witmotion* gearbeitet. Mit Hilfe einer App können Rohdaten aufgenommen und in Form einer txt-Datei gespeichert werden. Der Sensor enthält ein Gyroskop, ein Accelerometer und einen geomagnetischen Sensor. Für die Berechnungen dieses Kapitels werden ausschließlich die Beschleunigungsdaten herangezogen. Der Sensor ist auf eine Ausgabefrequenz von 50 Hz

6 Der Weg der Rohdaten bis zur Klassifizierung - Am Beispiel Laufen

eingestellt. Zur Aufnahme von Rohdaten während des Laufens wird der **WT901BLECL5.0** mithilfe von Tape wie eine Sportuhr am Handgelenk platziert.



Abbildung 6.2: Platzierung des Sensors
Quelle: Eigene Darstellung

Wiederholt werden Rohdaten von den Aktivitäten Laufen, Gehen und Stehen gesammelt und im Programm *MatLab* weiterverarbeitet. Die Daten der Läufe auf zwei unterschiedlichen Bodenbelägen und mit verschiedenen Laufgeschwindigkeiten dienen als Trainings- und Testdaten für ein Klassifikationsmodell. Die Vorgehensweise des Trainierens von einem Random Forest-Klassifikator wird im nächsten Abschnitt näher erläutert. Nachdem der Algorithmus trainiert ist, wird anhand der Beschleunigungsdaten einer Laufeinheit, bei der sowohl gelaufen als auch gegangen und gestanden wurde, der Weg von Rohdaten zur Klassifikation und Laufanalyse gezeigt und der Aufnahme meiner Sportuhr *Garmin Vivoactive 4s* gegenübergestellt. Der vollständige Code sowie alle verwendeten Datensätze sind im begleitenden GitHub-Repository verfügbar (<https://github.com/sabigrub/master-thesis>). Teile des Codes wurden unter Zuhilfenahme von Claude Sonnet 3.7, einem KI-Werkzeug entwickelt bzw. angepasst. Die finale Verantwortung für Inhalt und Korrektheit liegt bei der Verfasserin.

6.1.1 Training eines Klassifikators

Um die Klassifizierung durch Beschleunigungsdaten vollständig zeigen zu können, wurde ein binärer Klassifikator auf Matlab trainiert. Die Wahl fiel auf einen *Random Forest*.

Im ersten Schritt wurden Rohdaten von den Aktivitäten Laufen, Gehen und Stehen aufgenommen, um Trainings- und Testdaten zu sammeln. Dazu zeichnete ich mithilfe des Sensors WT901BLECL5.0 am Handgelenk (siehe Abbildung 6.2) diese Bewegungen wiederholt jeweils mehrere Minuten lang auf. Der resultierende Datensatz umfasste insgesamt 139.144 Messungen für das Training des Klassifikators sowie 11.400 Datenpunkte für

die anschließende Modellvalidierung. Bei der Lauferkennung durch einen binären Klassifikator wird zwischen zwei Aktivitätstypen unterschieden *Laufen* und *Nicht-Laufen*. Laufen wird mit 1 und Nicht-Laufen mit 0 beschriftet. Die Rohdaten wurden so aufgenommen, dass jede Aufzeichnung nur eine Aktivität enthielt - entweder Laufen oder Nicht-Laufen (Gehen oder Stehen). Dadurch erspart man sich eine aufwändige nachträgliche Kategorisierung.

Zunächst werden alle Lauf- bzw. Nicht-Laufdaten gelabelt und anschließend kombiniert. Bevor die Daten für das Training und die Validierung genutzt werden können, werden die Daten vorverarbeitet und die Feature-Vektoren jedes Fensters bestimmt. Die Trainingsdaten durchlaufen den gleichen Vorverarbeitungsprozess wie die Beschleunigungsdaten des Sporttreibenden, die von einer Sportuhr bzw. nach der Implementierung des Random Forest, klassifiziert werden sollen. Im nächsten Abschnitt wird der Vorverarbeitungs- und Merkmalsextraktionsprozess genau erläutert.

Die vorbearbeiteten Daten, die Feature-Vektoren für jedes Fenster, werden aufgeteilt in Trainings- und Testdaten. Ersteres wird zum Training des Random Forest herangezogen und letzteres für die Bewertung und Bestimmung der Genauigkeit des Modells. Anschließend wird die Anzahl der Samples, Features und die Klassenverteilung sowie das Klassenungleichgewicht abgefragt.

Im Rahmen der Datenaufbereitung werden zunächst sowohl die Trainings- als auch die Testdaten zufällig durchgemischt, um eine gleichmäßige Verteilung der Samples zu gewährleisten. Dabei wird ein fester Seed-Wert von 42 verwendet, um die Reproduzierbarkeit der Ergebnisse sicherzustellen. Anschließend werden die Daten in Features (X) und Labels (y) aufgeteilt - alle Spalten außer der letzten dienen als Eingabedaten, die letzte Spalte enthält die Zielvariable. Diese Trennung ist erforderlich, weil der Algorithmus wissen muss, welche Daten als Eingabe dienen und was vorhergesagt werden soll. Im nächsten Schritt wird das Random Forest-Modell trainiert, indem zunächst 100 Entscheidungsbäume definiert werden. Die TreeBagger-Funktion erstellt dann das Modell basierend auf den Trainingsdaten (X_{train}) und den entsprechenden Labels (y_{train}).

Abschließend werden Vorhersagen für die Testdaten erstellt und von Text- in Zahlenformat konvertiert. Zusätzlich wird die Gesamtgenauigkeit des Random-Forest-Modell berechnet, indem korrekte Vorhersagen gezählt und als Prozentsatz ausgegeben werden. Zur detaillierten Analyse lässt man eine Konfusionsmatrix erstellen und visualisieren, gefolgt von zusätzlichen Metriken wie der Anzahl korrekter/falscher Vorhersagen und klassenspezifischen Genauigkeiten für jede einzelne Klasse. Der folgende Code repräsentiert das Training des Klassifikators.

```

baseDir = ".";
dataDir = baseDir + "/data";
trainDataDir = dataDir + '/train';
testDataDir = dataDir + '/test';

trainDataFile = trainDataDir + '/processed/preprocessed_data.csv';
testDataFile = testDataDir + '/processed/preprocessed_data.csv';

%% 1. Dataloading
trainData = readtable(trainDataFile, 'VariableNamingRule', 'preserve');
testData = readtable(testDataFile, 'VariableNamingRule', 'preserve');

fprintf('Data analysis:\n');
fprintf('Train samples: %d\n', height(trainData));
fprintf('%d features and 1 target \n', width(trainData)-1);

% Class Imbalance
classDistribution = tabulate(trainData.Target);
classCounts = classDistribution(:, 2);
imbalanceRatio = max(classCounts) / min(classCounts);
fprintf('Imbalance ratio of train data: %.2f\n', imbalanceRatio);

%% 2. Data Shuffling
rng(42);
randomTrainIdx = randperm(height(trainData));
randomTestIdx = randperm(height(testData));
shuffledTrainData = trainData(randomTrainIdx, :);
shuffledTestData = testData(randomTestIdx, :);

% Split data in features and labels
X_train = shuffledTrainData(:, 1:end-1); % Features
y_train = shuffledTrainData.Target;      % Labels
X_test = shuffledTestData(:, 1:end-1);   % Features
y_test = shuffledTestData.Target;        % Labels

fprintf('Training samples: %d\n', length(y_train));
fprintf('Test samples: %d\n', length(y_test));

%% 4. Training des Random Forest Klassifikators
fprintf('Training Random Forest Classifier...\n');
numTrees = 100;
RF_model = TreeBagger(numTrees, X_train, y_train, ...

```

```

        'Method', 'classification', ...
        'OOBPrediction', 'on');

fprintf('Random Forest training completed.\n');

%% 5. Make Predictions on Test Set
fprintf('Making predictions on test set...\n');
RF_predictions_cell = predict(RF_model, X_test);
RF_predictions = str2double(RF_predictions_cell);

%% 6. Calculate Accuracy
RF_accuracy = sum(RF_predictions == y_test) / length(y_test) * 100;
fprintf('Random Forest Test Accuracy: %.2f%%\n', RF_accuracy);

%% 7. Display Confusion Matrix
uniqueClasses = unique(y_test);
classNames = cellstr(string(uniqueClasses));

figure;
cm_RF = confusionmat(y_test, RF_predictions);
confusionchart(cm_RF, classNames);
title('Random Forest Confusion Matrix');

%% 8. Additional Metrics
fprintf('\nDetailed Results:\n');
fprintf('Correct predictions: %d out of %d\n', sum(RF_predictions ==
↪ y_test), length(y_test));
fprintf('Incorrect predictions: %d\n', sum(RF_predictions ~= y_test));

for i = 1:length(uniqueClasses)
    classIdx = (y_test == uniqueClasses(i));
    classCorrect = sum(RF_predictions(classIdx) == y_test(classIdx));
    classTotal = sum(classIdx);
    if classTotal > 0
        classAccuracy = classCorrect / classTotal * 100;
        fprintf('Class %d accuracy: %.2f%% (%d/%d)\n', ...
            uniqueClasses(i), classAccuracy, classCorrect, classTotal);
    end
end

save(baseDir + '/classifier-rf', 'RF_model');
```

6.1.2 Limitationen

Es wäre möglich, den trainierten Random-Forest-Algorithmus auf einer Sportuhr für die Lauferkennung zu implementieren. Jedoch ist das Trainingsdatenset beschränkt auf zwei Personen und zwei verschiedene Untergründe (Asphalt, Waldboden). Für eine kommerzielle Sportuhr müssten die Daten, mit denen der RF trainiert ist, umfangreicher und vielfältiger (Bodenbelag, Tempo, Laufstile) sein, da jede Person einen anderen Laufstil und somit ein anderes Bewegungsmuster erzeugt. Zusätzlich beeinflusst das Gelände, in dem gelaufen wird, das Beschleunigungsmuster.

6.2 Klassifikation von Aktivität *Laufen* - Ein Beispiel

Der Beschleunigungssensor der Sportuhr erfasst die Bewegungsdaten. Leistungsfähige Geräte verarbeiten diese direkt auf der Uhr für Echtzeitanalyse, während Geräte mit begrenzter Rechenkapazität die Daten nach der Synchronisation mit einer Smartphone-App in die Cloud übertragen.

Der normalerweise auf der Uhr oder in der Cloud stattfindende Verarbeitungsprozess wird hier in MATLAB nachgestellt. Jede Verarbeitungsstufe der Rohdaten wird durch Grafiken und MATLAB-Code erläutert. Die Parameter wie Fenstergröße und Filtereinstellungen basieren auf der Literaturrecherche aus den vorangegangenen Kapiteln. Die konkreten Implementierungen variieren zwischen den Herstellern.

Für das Beispiel werden Aufnahmen einer 10-minütigen Laufeinheit verwendet, die sowohl mit dem Sensor WT901BLECL5.0 als auch zur Kontrolle mit der Garmin vivoactive 4s aufgezeichnet wurden. Die Einheit bestand aus alternierenden 30-Sekunden-Intervallen von Laufen und Gehen.

6.2.1 Rohdatenimport

Im ersten Schritt müssen die aufgezeichneten Rohdaten der txt-Datei in Matlab importiert und die Dateipfade festgelegt werden. Der folgende Code wurde dafür verwendet.

```
dataDir = "./data/inference";
dataFilePath = dataDir + "/inference.txt";

% Data loading
data = readtable(dataFilePath, 'Delimiter', '\t', 'VariableNamingRule',
    ↪ 'preserve');
assert(all(sum(ismissing(data)) == 0));
```

6.2 Klassifikation von Aktivität Laufen - Ein Beispiel

In der vom Witmotion-Sensor übertragenden txt-Datei befinden sich Daten des Beschleunigungssensors, des Gyroskops und des geomagnetischen Sensors aller Achsen sowie Zeit, Winkelveränderungen und Temperatur. Für den weiteren Verlauf werden ausschließlich die Beschleunigungsdaten verwendet, da sich die Arbeit auf die Verarbeitung der Accelerometerdaten beschränkt. Die Beschleunigungsdaten werden, wie auf den Abbildungen (6.3) zu sehen, in der *Witmotion-App* direkt dargestellt. Bei Sportuhren werden die Daten aller Sensoren für die Klassifizierung genutzt, was genauere Analysen und Klassifizierungen von komplexeren Aktivitäten ermöglicht. Zum Vergleich sieht man auf der linken



Abbildung 6.3: Links: Gehen, rechts: Laufen
Quelle: Eigene Darstellung

Seite die Aufnahmen des Beschleunigungssensors beim Gehen und rechts beim Laufen.

6.2.2 Vorverarbeitung

Für die Vorverarbeitung werden die Beschleunigungsdaten der einzelnen Achsen aus der Datei extrahiert und daraus die Gesamtbeschleunigung berechnet. Anschließend werden die Daten zur Rauschunterdrückung und zum Entfernen der Erdbeschleunigung gefiltert. Dafür wird MatLab die Abtastfrequenz des Beschleunigungssensors von 50 Hz angegeben. Der Butterworth-Tiefpassfilter 4.Ordnung wird mit einer Grenzfrequenz von 20 Hz angewendet, da menschliche Bewegungen wie Gehen und Laufen eine niedrigere Frequenz haben. Hochfrequentes Rauschen über 20 Hz entfernt dieser Filter. Die Funktion *filtfilt* wendet den Filter zur Elimination von Phasenverschiebungen vorwärts und rückwärts auf die Daten an. Dieser Schritt ist bedeutsam für das Erkennen von zeitlichen Mustern (z.B. bei der Schritterkennung oder Aktivitätsklassifikation). Im nächsten Schritt

dient ein weiterer Butterworth-Hochpassfilter zur Entfernung der Schwerkraft aus den Rohdaten. Als letzten Schritt der Vorverarbeitung wird der Datenstrom in Fenster von 2 Sekunden mit 50% Überlappung segmentiert.

```
function data = preprocessData(rawData)
    % Extracting acceleration data
    accX = rawData(:, 'AccX(g)');
    accY = rawData(:, 'AccY(g)');
    accZ = rawData(:, 'AccZ(g)');

    % Calculate acceleration magnitude
    accMag = sqrt(accX.^2 + accY.^2 + accZ.^2);

    % Sampling rate
    fs = 50; % Hz

    % Low-pass filter
    cutoffFreq = 20; % Hz
    [b, a] = butter(4, cutoffFreq/(fs/2), 'low');
    accXFiltered = filtfilt(b, a, accX);
    accYFiltered = filtfilt(b, a, accY);
    accZFiltered = filtfilt(b, a, accZ);
    accMagFiltered = filtfilt(b, a, accMag);

    % High-pass filter
    cutoffFreqHigh = 0.1; % Hz
    [b, a] = butter(4, cutoffFreqHigh/(fs/2), 'high');
    accXHighpass = filtfilt(b, a, accXFiltered);
    accYHighpass = filtfilt(b, a, accYFiltered);
    accZHighpass = filtfilt(b, a, accZFiltered);

    % Define windowing parameters
    windowSize = 2 * fs; % 2 seconds
    overlap = 0.5; % 50% overlap
    step = round(windowSize * (1 - overlap));
    numWindows = floor((length(accXHighpass) - windowSize) / step) + 1;
```

6.2.3 Feature-Extraktion

In jedem Fenster werden Merkmale in der Frequenz und auch in der Zeitdomäne extrahiert. Insgesamt entschied ich mich, 15 Features pro Fenster aus den Daten zu extrahieren. Darunter Durchschnitt, Standardabweichung, Maximum und Minimum für alle Achsen

im Zeitbereich. Vor der FFT-Transformation des Signals in den Frequenzbereich wird der restliche Direct-Current Anteil (DC-Anteil) in den Beschleunigungsdaten zur Verhinderung von spektralen Verzerrungen entfernt. Ein DC-Offset manifestiert sich als dominante Komponente bei 0 Hz und kann durch seine Amplitude andere physiologisch relevante Frequenzen im Spektrum überdecken. Zusätzlich wird die FFT durch Zero-Padding mit 512 Punkten für eine verbesserte Frequenzauflösung von 0.098 Hz durchgeführt. Diese höhere Frequenzauflösung ist besonders wichtig für die Erkennung der dominanten Frequenz, da sie es ermöglicht, zwischen eng beieinanderliegenden Frequenzkomponenten zu unterscheiden. Aus dem resultierenden Magnitude-Spektrum wird die dominante Frequenz im physiologisch relevanten Bereich von 0.5-10 Hz identifiziert. Dieser Bereich deckt typische menschliche Bewegungsfrequenzen ab, während Frequenzen unterhalb von 0.5 Hz meist nur Rauschen oder Sensordrift repräsentieren. Zusätzlich wird die spektrale Energie berechnet, die ein Maß für die Gesamtaktivität im Frequenzbereich ist. Als 15. und letztes Feature wird die Signal Magnitude Area berechnet. Dies ergibt einen 15-dimensionalen Feature-Vektor pro Fenster.

```
features = zeros(numWindows, 15);

for i = 1:numWindows
    startIdx = (i-1)*step + 1;
    endIdx = startIdx + windowSize - 1;

    % Window extraction
    xWindow = accXHighpass(startIdx:endIdx);
    yWindow = accYHighpass(startIdx:endIdx);
    zWindow = accZHighpass(startIdx:endIdx);
    magWindow = accMagFiltered(startIdx:endIdx);

    % Timebased features
    features(i, 1:3) = [mean(xWindow), mean(yWindow),
        ↪ mean(zWindow)];
    features(i, 4:6) = [std(xWindow), std(yWindow), std(zWindow)];
    features(i, 7:9) = [max(xWindow), max(yWindow), max(zWindow)];
    features(i, 10:12) = [min(xWindow), min(yWindow), min(zWindow)];

    % Frequencybased features
    % Remove DC offset to avoid bias
    magWindow = magWindow - mean(magWindow);

    % FFT for frequency analysis
    nfft = 512;
    fftResult = fft(magWindow, nfft);
    fftMag = abs(fftResult);
    fftMag = fftMag(1:floor(nfft/2)+1);
```

```

% Create frequency vector with improved resolution
freqs = (0:length(fftMag)-1) * fs / nfft;

% Find dominant frequency
minFreq = 0.5;
maxFreq = 10;
validIdx = (freqs >= minFreq) & (freqs <= maxFreq);

if any(validIdx)
    validFftMag = fftMag(validIdx);
    validFreqs = freqs(validIdx);
    [~, maxIdx] = max(validFftMag);
    dominantFreq = validFreqs(maxIdx);
else
    dominantFreq = 0; % Fallback if no valid frequency found
end

features(i, 13) = dominantFreq;

% Spectral energy (exclude DC component at index 1)
features(i, 14) = sum(fftMag(2:end).^2);

% Debug output for first few windows
if i <= 3
    fprintf('Window %d: Dominant frequency = %.2f Hz\n', i,
        ↪ dominantFreq);
end

% Signal Magnitude Area (SMA)
features(i, 15) = sum(abs(xWindow)) + sum(abs(yWindow)) +
    ↪ sum(abs(zWindow));
end

% 9. Build final dataset
featureNames = {'mean_x', 'mean_y', 'mean_z', 'std_x', 'std_y',
    ↪ 'std_z', ...
        'max_x', 'max_y', 'max_z', 'min_x', 'min_y',
        ↪ 'min_z', ...
        'dominant_freq', 'spectral_energy', 'sma'};
data = array2table(features, "VariableNames", featureNames);
end

```

```
preprocessedData = preprocessData(data);
```

Die vorverarbeiteten Daten werden in Form einer CSV-Datei gespeichert.

```
writetable(preprocessedData, dataDir + '/preprocessed_data.csv');
```

Eine Min-Max-Normalisierung wird nicht implementiert, da Random Forest Klassifikatoren aufgrund ihrer schwellenwertbasierten Entscheidungsstruktur weitgehend skalenunabhängig operieren. Ebenso wird auf PCA verzichtet, da der 15-dimensionale Merkmalsvektor bereits eine angemessene Kompaktheit aufweist und die Interpretierbarkeit der Features erhalten bleiben soll.

6.2.4 Klassifizierung

Die gespeicherten Feature-Vektoren dienen als Eingabe für den Klassifikator. Für die Klassifizierung werden die vorverarbeiteten Inferenz-Daten und das zuvor trainierte Random Forest-Modell geladen. Anschließend wird der Klassifikator auf die neuen Daten angewendet, um Vorhersagen zu generieren, ob und wann eine Person läuft (Klasse 1) oder nicht läuft (Klasse 0). Das System wandelt die Ergebnisse in ein einheitliches Zahlenformat um und erstellt dann eine Statistik darüber, wie oft jede Aktivität erkannt wurde (als absolute Zahlen und in Prozentsätzen). Abschließend werden die Klassifikationsergebnisse als CSV-Datei gespeichert, um eine weitere Analyse oder Visualisierung zu ermöglichen.

```
% Define the directory containing data files
baseDir = ".";
dataFilePath = baseDir + "/data/inference/preprocessed_data";
modelPath = baseDir + '/classifier-rf';

data = readtable(dataFilePath, 'VariableNamingRule', 'preserve');
model = load(modelPath).RF_model;
X_inference = data{:, :};
predictions = predict(model, X_inference);

% Convert cell array of strings to numeric if needed
if iscell(predictions)
    predictions_numeric = str2double(predictions);
else
    predictions_numeric = predictions;
end

% Count the predictions
count_0 = sum(predictions_numeric == 0);
```

```
count_1 = sum(predictions_numeric == 1);
total_predictions = length(predictions_numeric);

% Display the counts
fprintf('Prediction counts:\n');
fprintf('Class 0 (not running): %d predictions (%.1f%%)\n', count_0,
    ↪ (count_0/total_predictions)*100);
fprintf('Class 1 (running): %d predictions (%.1f%%)\n', count_1,
    ↪ (count_1/total_predictions)*100);
fprintf('Total predictions: %d\n', total_predictions);

fprintf('\nDetailed breakdown:\n');
tabulate(predictions_numeric)

writetable(array2table(predictions), baseDir +
    ↪ '/data/inference/results.csv')
```

Die Abbildung (6.4) zeigt die Ausgabe des RF-Klassifikators in Matlab. Von den 593 Fenstern werden 259 als *Nicht-Laufen* und 334 als *Laufen* klassifiziert. Das Ergebnis der

```
>> inference_classification
Prediction counts:
Class 0 (not running): 259 predictions (43.7%)
Class 1 (running): 334 predictions (56.3%)
Total predictions: 593

Detailed breakdown:
  Value  Count  Percent
    0      259   43.68%
    1      334   56.32%
```

Abbildung 6.4: Klassifizierungsausgabe in MatLab

Klassifizierung zeigt, dass 56.32% der Zeit gelaufen und 43.68% der Zeit nicht gelaufen wurde. Das entspricht einer Laufzeit von 5.6 Minuten und einer Nicht-Laufzeit von 4.4 Minuten. Die 30 Sekunden-Intervalle sind in einem 4 : 00 – 4 : 50 Schnitt gelaufen worden und es benötigt einige Sekunden, von schnellem Lauftempo ins Gehen zu wechseln. Durch diese Tatsache lässt sich erklären, warum 56% als Laufen klassifiziert werden. In der Grafik (6.5) wird die Klassifikation in Abhängigkeit der Zeit sowie die Anteile von Laufen vs. Nicht-Laufen veranschaulicht.

6.2 Klassifikation von Aktivität Laufen - Ein Beispiel

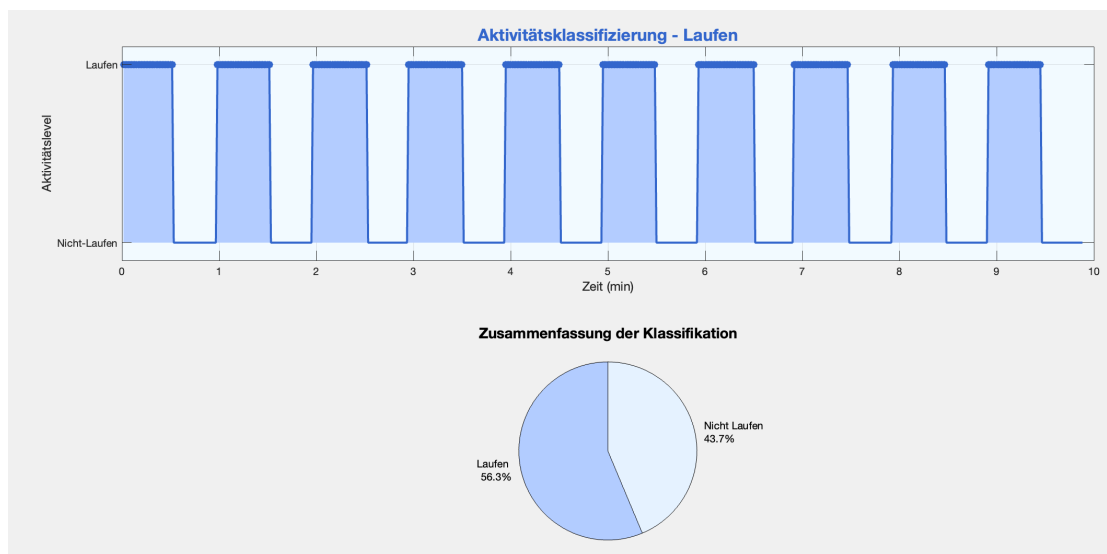


Abbildung 6.5: Klassifizierung durch trainierten RF

6.2.5 Kadenz und weitere Laufanalysen

Moderne Sportuhren erfassen weit mehr als nur grundlegende Schrittdaten – sie liefern detaillierte Laufanalysen, die entscheidende Einblicke in Technik und Effizienz bieten. Die Kadenz, gemessen als Schritte pro Minute, stellt dabei einen zentralen Leistungsindikator dar. Während eine Schrittzählung mittlerweile Standard in jedem Wearable ist, ermöglicht die präzise Kadenzanalyse während des Lauftrainings eine gezielte Optimierung des Laufstils. Eine erhöhte Schrittfrequenz wirkt sich nachweislich positiv auf die Gelenkgesundheit aus: Durch kürzere, häufigere Schritte reduzieren sich die Aufprallkräfte auf Fuß- und Kniegelenke erheblich. Diese biomechanische Optimierung minimiert nicht nur das Verletzungsrisiko, sondern steigert auch die Laufökonomie.

Die zwei untenstehenden Grafiken zeigen die Kadenz des 10-minütigen Laufes in Abhängigkeit von der Zeit. Die obenstehende Abbildung, die in der Garmin-App generiert worden ist, ist der Visualisierung der Berechnungen in MatLab gegenübergestellt, wobei die y -Achse bei der Garmin-Darstellung gestaucht als bei der MatLab-Version ist.

6 Der Weg der Rohdaten bis zur Klassifizierung - Am Beispiel Laufen

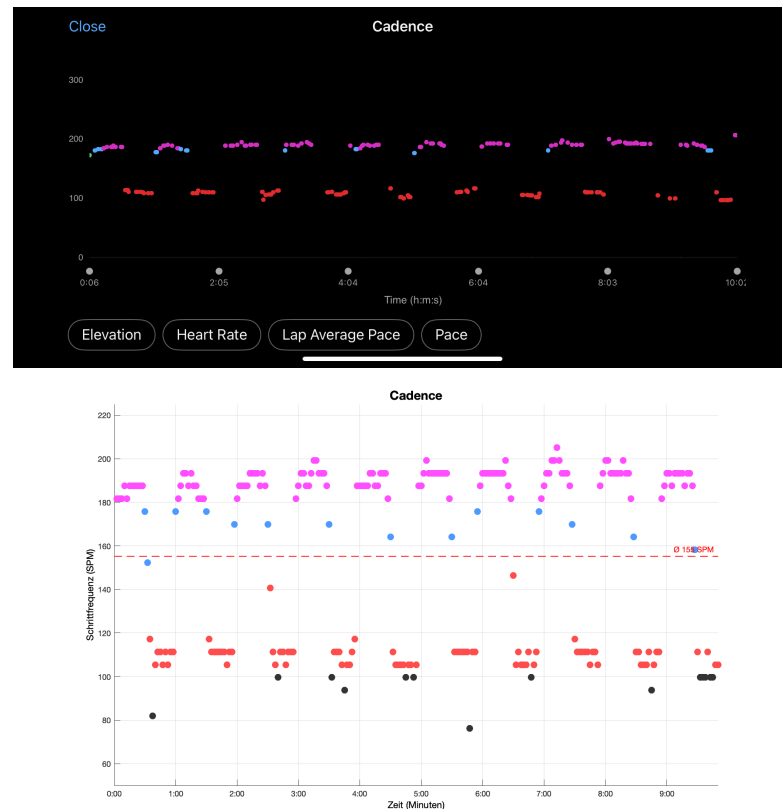


Abbildung 6.6: Schritt看率: Garmin(oben) und MatLab(unten)

Quelle: Garmin App; Eigene Darstellung

Die FFT kommt auch zur Berechnung der Schritt看率 zum Einsatz, da sie über die *Dominante Frequenz* bestimmt wird. Folgendermaßen wurde die Schritt看率 in MatLab berechnet.

```
baseDir = "./data/inference";
dataFilePath = baseDir + "/inference.txt";
data = readtable(dataFilePath, 'Delimiter', '\t', 'VariableNamingRule',
    ↪ 'preserve');
assert(all(sum(ismissing(data)) == 0));

% Extracting acceleration data
accX = data(:, 'AccX(g)');
accY = data(:, 'AccY(g)');
accZ = data(:, 'AccZ(g)');

% Calculate acceleration magnitude
accMag = sqrt(accX.^2 + accY.^2 + accZ.^2);
```

```

% Remove DC offset
accMag = accMag - mean(accMag);

% Parameters
fs = 50;
windowSize = 5;
overlap = 0.5;

% Sliding window analysis
windowSamples = windowSize * fs;
stepSize = windowSamples * (1 - overlap);
numWindows = floor((length(accMag) - windowSamples) / stepSize) + 1;

stepFrequencies = zeros(numWindows, 1);
timeWindows = zeros(numWindows, 1);

for i = 1:numWindows
    % Window extraction
    startIdx = (i-1) * stepSize + 1;
    endIdx = startIdx + windowSamples - 1;
    window = accMag(startIdx:endIdx);

    % Time for this window (center point)
    timeWindows(i) = (startIdx + endIdx) / (2 * fs);

    % FFT for frequency analysis
    nfft = 512;
    fftResult = abs(fft(window, nfft));
    fftResult = fftResult(1:floor(nfft/2)+1);

    % Frequency vector
    freqs = (0:length(fftResult)-1) * fs / nfft;

    % Step frequency in range 1-4 Hz
    validIdx = (freqs >= 1) & (freqs <= 4);

    if any(validIdx)
        [~, maxIdx] = max(fftResult(validIdx));
        validFreqs = freqs(validIdx);
        stepFrequencies(i) = validFreqs(maxIdx) * 60; % Umrechnung in
        ↳ Schritte/Min
    else

```



```

        stepFrequencies(i) = 0;
    end
end

% Convert time to minutes
timeMinutes = timeWindows / 60;

```

Der Code für die Darstellung der Schrittfrequenz in Abhängigkeit von der Zeit lautet folgendermaßen:

```

% Plot with white background
figure('Position', [100, 100, 1000, 600], 'Color', 'w');
ax = axes('Color', 'w', 'XColor', 'k', 'YColor', 'k');

% Color coding of points
colors = zeros(length(stepFrequencies), 3);
for i = 1:length(stepFrequencies)
    if stepFrequencies(i) < 100
        colors(i,:) = [0.2, 0.2, 0.2];
    elseif stepFrequencies(i) < 150
        colors(i,:) = [1, 0.3, 0.3];
    elseif stepFrequencies(i) < 180
        colors(i,:) = [0.3, 0.6, 1];
    else
        colors(i,:) = [1, 0.3, 1];
    end
end

% Scatter plot with color coding
scatter(timeMinutes, stepFrequencies, 80, colors, 'filled',
    ⇨ 'MarkerEdgeColor', 'none');

% Axis styling
grid on;
ax.GridColor = [0.7, 0.7, 0.7];
ax.GridAlpha = 0.5;
xlabel('Zeit (Minuten)', 'Color', 'k', 'FontSize', 12);
ylabel('Schrittfrequenz (SPM)', 'Color', 'k', 'FontSize', 12);
title('Cadence', 'Color', 'k', 'FontSize', 16, 'FontWeight', 'bold');

% Y-axis limits
ylim([50, max([200, max(stepFrequencies)+20])]);
xlim([0, max(timeMinutes)]);

```

```

ax.XTickLabel = string(ax.XTick);
ax.YTickLabel = string(ax.YTick);

% Time format for X-axis (mm:ss)
xticks_minutes = 0:1:ceil(max(timeMinutes));
xticks(xticks_minutes);
xticklabels(arrayfun(@(x) sprintf('%d:%02d', floor(x), mod(x*60, 60)),
    ↪ xticks_minutes, 'UniformOutput', false));

% Display average value
avgStepFreq = mean(stepFrequencies(stepFrequencies > 0));
hold on;
yline(avgStepFreq, 'r--', sprintf('Ø %.0f SPM', avgStepFreq),
    ↪ 'LineWidth', 1.5);
hold off;

```

6.3 Zusammenfassung

Die praktische Umsetzung des gesamten Verarbeitungsweges von Rohdaten bis zur Aktivitätsklassifikation in diesem Kapitel demonstriert die Komplexität und gleichzeitig die Machbarkeit moderner Bewegungsanalyse. Der trainierte Random Forest-Algorithmus unterschied Lauf- und Nicht-Laufaktivitäten mit zufriedenstellender Genauigkeit. Die Klassifizierungsergebnisse (56.32% Laufen, 43.68% Nicht-Laufen) weichen erwartungsgemäß von den nominell gleichmäßigen 30-Sekunden-Intervallen ab, was auf die physiologischen Übergangsphasen zwischen den Aktivitätszuständen zurückzuführen ist. Zusätzlich bestätigt die quantitative Validierung durch den Vergleich der FFT-basierten Kadenzberechnung mit den Referenzdaten der Garmin-Sportuhr die methodische Korrektheit der implementierten Algorithmen. Weiters demonstriert die erfolgreiche Extraktion von 15 charakteristischen Merkmalen sowohl im Zeit- als auch im Frequenzbereich, dass bereits mit einem gezielt ausgewählten kompakten Feature-Set robuste Bewegungsklassifikationen realisierbar sind.

Die aufgezeigten Limitationen – insbesondere das begrenzte Trainingsdatenset mit nur zwei Personen und zwei Bodenbelägen – verdeutlichen jedoch den erheblichen Aufwand, der für kommerzielle Anwendungen erforderlich ist. Für eine robuste, personenübergreifende Klassifikation müssten deutlich umfangreichere und diversifizierte Datensätze verwendet werden, die verschiedene Laufstile, Geschwindigkeiten und Umgebungsbedingungen abdecken. Zusammengefasst bietet dieses Kapitel einen wertvollen Einblick in die tatsächlichen Verarbeitungsschritte moderner Sportuhren und zeigt auf, wie aus einfachen Beschleunigungssignalen komplexe Bewegungsanalysen entstehen.

7 Fazit

Im Laufe dieser Arbeit wurden die mathematischen Grundlagen und praktischen Implementierungen für die effiziente Verarbeitung von Beschleunigungssensordaten in Sportuhren systematisch untersucht, um aussagekräftige Trainingsanalysen zu ermöglichen.

7.1 Erkenntnisse zur Signalverarbeitung

Menschliche Bewegungen lassen sich als komplexe Schwingungen verstehen, die mathematisch durch Sinusfunktionen oder deren Superposition beschrieben werden können. Zur Quantifizierung dieser Bewegungen muss das kontinuierliche analoge Sensorsignal zunächst durch Sampling in ein diskretes digitales Signal umgewandelt werden. Das digitalisierte Signal wird anschließend mittels Hoch- und Tiefpassfiltern von Rauschen befreit und in zeitliche Fenster segmentiert. Die Kombination aus Fensterlänge und Abtastrate erweist sich als entscheidender Faktor für die Genauigkeit der Bewegungsklassifizierung. Bei der Optimierung dieser Parameter müssen die begrenzten Rechenkapazitäten und der Energieverbrauch der Sportuhr sowie die spezifischen Charakteristika der zu klassifizierenden Bewegungen berücksichtigt werden.

Aus den segmentierten Daten werden verschiedene Merkmale sowohl aus dem Zeit- als auch aus dem Frequenzbereich extrahiert. Die Arbeit demonstriert erfolgreich die Integration von Zeit- und Frequenzdomänen-Features durch hybride Algorithmen. Während Zeitdomänen-Merkmale (Mittelwerte, Standardabweichungen, Signal Magnitude Area) durch ihre geringe Rechenintensität eine erste schnelle Aktivitätserkennung ermöglichen, gewährleisten Frequenzdomänen-Features (dominante Frequenz, spektrale Energie, spektrale Entropie) die präzise Klassifizierung periodischer Bewegungsmuster. Diese komplementäre Nutzung erreicht eine Erkennungsgenauigkeit von 92.38% im Frequenzbereich gegenüber 83.48% im reinen Zeitbereich. Das zu extrahierende Feature-Set variiert je nach Betriebsmodus der Sportuhr – abhängig davon, ob ein spezifisches Workout getrackt oder zwischen verschiedenen Tagesaktivitäten wie Gehen, Laufen und Radfahren unterschieden werden soll.

Für die Berechnung der Frequenzmerkmale müssen die Zeitbereichsdaten zunächst in den Frequenzbereich transformiert werden, wobei die Fast-Fourier-Transformation (FFT) eine zentrale Rolle spielt. Je nach verwendetem Maschinenlernalgorithmus kann anschließend eine Dimensionsreduktion der Merkmalsvektoren durchgeführt werden, bevor diese zur Klassifizierung in einen trainierten Klassifikator eingegeben werden. Die praktische

Implementierung einer vollständigen Verarbeitungskette von Rohdaten bis zur erfolgreichen Bewegungsklassifizierung validiert die vorgestellten theoretischen Konzepte. Klassische Maschinenlernalgorithmen wie Random Forest erweisen sich als effektiv für die gegebenen Anforderungen der Aktivitätserkennung, insbesondere bei der Erkennung von alltäglichen Bewegungsmustern wie Gehen, Laufen und Radfahren.

7.2 Spezifische Forschungsergebnisse zur Fourier-Transformation

Die zentrale Forschungsfrage nach den Auswirkungen der Fourier-Transformation auf Rechenaufwand, Energieverbrauch und Erkennungsgenauigkeit konnte umfassend beantwortet werden.

7.2.1 Recheneffizienz

Einerseits ermöglicht die FFT als Berechnung der Diskreten-Fourier-Transformation eine immense Effizienzsteigerung bezüglich Rechenaufwand. Die Komplexitätsreduktion von $O(N^2)$ auf $O(N \log N)$ macht erst die Anwendung der Echtzeitverarbeitung von Sensordaten im Frequenzbereich möglich. Andererseits reduzieren die Symmetrieeigenschaften der trigonometrischen Basisfunktionen den Rechenaufwand systematisch um 50%, da bei Signalen mit bekannter Symmetrie nur die Hälfte der Fourierkoeffizienten berechnet werden muss. Diese Eigenschaft ist besonders relevant, da viele menschliche Bewegungsmuster symmetrische Komponenten aufweisen.

7.2.2 Energieeffizienz

Diese Rechenoptimierungen wirken sich positiv auf die Energieeffizienz aus. Die drastische Reduktion der benötigten arithmetischen Operationen führt zu proportional geringerem Energieverbrauch, was besonders relevant für batteriebetriebene Wearables wie Sportuhren ist. Zusätzlich ist die Halbierung des Speicherbedarfs durch Symmetrienausnutzung vorteilhaft, da Speicherzugriffe erheblich zum Energieverbrauch beitragen.

7.2.3 Erkennungsgenauigkeit

Drei fundamentale mathematische Eigenschaften der Fourieranalyse gewährleisten die Präzision der Aktivitätserkennung. Die eindeutige Darstellung periodischer Funktionen als Summe trigonometrischer Komponenten garantiert reproduzierbare und zuverlässige Ergebnisse. Das Parseval-Theorem stellt sicher, dass bei der Transformation vom Zeit- in den Frequenzbereich keine Informationen verloren gehen, wodurch alle charakteristischen

Merkmale der Originalsignale erhalten bleiben. Die Fähigkeit der Fouriertransformation, verborgene Periodizitäten zu identifizieren, die im Zeitbereich nicht erkennbar sind, ermöglicht zuverlässige Unterscheidungen zwischen verschiedenen Aktivitäten wie Laufen, Gehen und Radfahren anhand ihrer charakteristischen Frequenzmuster.

7.3 Limitationen und Ausblick

Die vorliegende Arbeit gibt Einblick in die Welt der sensorbasierten Bewegungserkennung. Die Fourier-Transformation hat sich aufgrund ihrer dokumentierten Rechen- und Energieeffizienz (50% Reduktion durch Symmetrie) als weitverbreitetes Werkzeug der Signalverarbeitung in Wearable-Systemen etabliert. Dennoch eröffnen moderne Entwicklungen komplementäre Forschungsrichtungen: Deep-Learning-Ansätze mit automatischer Feature-Extraktion können die hier präsentierte manuelle Merkmalsselektion durch End-to-End-Learning ersetzen, während Wavelet Transformationen die Limitationen der Fourier Analyse bei nicht-stationären Signalen durch simultane Zeit-Frequenz-Lokalisierung überwinden. Die rasante Entwicklung von Hardware und Software bei Wearables verbessert nicht nur die Genauigkeit der Aktivitätserkennung, sondern ermöglicht auch die Klassifizierung und Analyse komplexerer Bewegungsmuster. Die Fourier-Analyse schafft aufgrund ihrer mathematischen Eigenschaften eine solide Grundlage für die Schrittfrequenzbestimmung und Klassifizierung periodischer Bewegungen. Zukünftige Forschung wird jedoch zeigen, inwieweit effizientere Algorithmen und Deep-Learning-Ansätze die traditionelle Fourier-Analyse und manuelle Merkmalsextraktion ergänzen oder (teilweise) ersetzen können.

Literatur

- [1] S. W. Amos und Roger Amos. *Newnes Dictionary of Electronics*. en. Google-Books-ID: c4qHqtC9JkgC. Elsevier, März 2002. ISBN: 978-0-08-052405-4.
- [2] Ferhat Attal u. a. “Physical Human Activity Recognition Using Wearable Sensors”. en. In: *Sensors* 15.12 (Dez. 2015). Number: 12 Publisher: Multidisciplinary Digital Publishing Institute, S. 31314–31338. ISSN: 1424-8220. DOI: 10.3390/s151229858. URL: <https://www.mdpi.com/1424-8220/15/12/29858> (besucht am 05.03.2025).
- [3] O. Baños u. a. “Window Size Impact in Human Activity Recognition”. In: *Sensors (Basel, Switzerland)* 14 (2014), S. 6474–6499. DOI: 10.3390/s140406474. URL: <https://consensus.app/papers/window-size-impact-in-human-activity-recognition-ba%C3%B1os-g%C3%A1lvez/b9be7f2965fa5d6cb87d7c4c21d0fa58/> (besucht am 10.03.2025).
- [4] René Bocksch. *Infografik: Nutzung von Wearables in Europa deutlich gestiegen*. de. Dez. 2023. URL: <https://de.statista.com/infografik/31378/nutzung-von-wearables-in-europa> (besucht am 03.03.2025).
- [5] Andreas Bulling, Ulf Blanke und Bernt Schiele. “A tutorial on human activity recognition using body-worn inertial sensors”. In: *ACM Comput. Surv.* 46.3 (Jan. 2014), 33:1–33:33. ISSN: 0360-0300. DOI: 10.1145/2499621. URL: <https://dl.acm.org/doi/10.1145/2499621> (besucht am 07.03.2025).
- [6] Kaixuan Chen u. a. “Deep Learning for Sensor-based Human Activity Recognition: Overview, Challenges, and Opportunities”. en. In: *ACM Computing Surveys* 54.4 (Mai 2022), S. 1–40. ISSN: 0360-0300, 1557-7341. DOI: 10.1145/3447744. URL: <https://dl.acm.org/doi/10.1145/3447744> (besucht am 27.02.2025).
- [7] James W. Cooley, Peter A. W. Lewis und Peter D. Welch. “The Fast Fourier Transform and Its Applications”. In: *IEEE Transactions on Education* 12.1 (März 1969). Conference Name: IEEE Transactions on Education, S. 27–34. ISSN: 1557-9638. DOI: 10.1109/TE.1969.4320436. URL: <https://ieeexplore.ieee.org/document/4320436> (besucht am 27.02.2025).
- [8] Vincenzo Dentamaro u. a. “Human activity recognition with smartphone-integrated sensors: A survey”. In: *Expert Systems with Applications* 246 (Juli 2024), S. 123143. ISSN: 0957-4174. DOI: 10.1016/j.eswa.2024.123143. URL: <https://www.sciencedirect.com/science/article/pii/S0957417424000083> (besucht am 05.03.2025).

- [9] FFT. URL: <https://www.nti-audio.com/de/service/wissen/fast-fourier-transformation-fft> (besucht am 24.03.2025).
- [10] Davide Figo u. a. “Preprocessing techniques for context recognition from accelerometer data”. eng ; jpn. In: *Personal and Ubiquitous Computing* 14.7 (2010). Publisher: Springer Science and Business Media LLC, S. 645–662. ISSN: 1617-4909. DOI: 10.1007/s00779-010-0293-9.
- [11] Narit Hnoohom, Sakorn Mekruksavanich und Anuchit Jitpattanakul. “Human Activity Recognition Using Triaxial Acceleration Data from Smartphone and Ensemble Learning”. In: *2017 13th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*. Jaipur: IEEE, Dez. 2017, S. 408–412. ISBN: 978-1-5386-4283-2. DOI: 10.1109/SITIS.2017.73. URL: <https://ieeexplore.ieee.org/document/8334779/> (besucht am 22.04.2025).
- [12] Statista Market Insights. *Smartwatches - Weltweit / Statista Marktprognose*. de. URL: <http://frontend.xmo.prod.aws.statista.com/outlook/hmo/digital-health/digital-fitness-well-being/fitness-tracker/smartwatches/weltweit> (besucht am 03.03.2025).
- [13] Magdalena Jachymek u. a. “Wristbands in Home-Based Rehabilitation—Validation of Heart Rate Measurement”. In: *Sensors (Basel, Switzerland)* 22 (2021). DOI: 10.3390/s22010060. URL: <https://consensus.app/papers/wristbands-in-homebased-rehabilitation%E2%80%94validation-of-jachymek-jachymek/0b77d45f8c8252f088d8ccd1cc25451c/> (besucht am 07.03.2025).
- [14] Milagros Jaén-Vargas u. a. “Effects of sliding window variation in the performance of acceleration-based human activity recognition using deep learning models”. en. In: *PeerJ Computer Science* 8 (Aug. 2022). Publisher: PeerJ Inc., e1052. ISSN: 2376-5992. DOI: 10.7717/peerj-cs.1052. URL: <https://peerj.com/articles/cs-1052> (besucht am 11.03.2025).
- [15] Sebastian Klöß. *Die Zukunft der Consumer Technology 2024*. de. Sep. 2024. URL: <https://www.bitkom.org/Bitkom/Publikationen/Die-Zukunft-der-Consumer-Technology-2024> (besucht am 03.03.2025).
- [16] H.J. Landau. “Sampling, data transmission, and the Nyquist rate”. In: *Proceedings of the IEEE* 55.10 (Okt. 1967). Conference Name: Proceedings of the IEEE, S. 1701–1706. ISSN: 1558-2256. DOI: 10.1109/PROC.1967.5962. URL: <https://ieeexplore.ieee.org/document/1447892/?arnumber=1447892> (besucht am 26.03.2025).
- [17] Christian B. Lang und Norbert Pucker. “Fourierreihe”. de. In: *Mathematische Methoden in der Physik*. Springer Spektrum, Berlin, Heidelberg, 2016, S. 489–512. ISBN: 978-3-662-49313-7. DOI: 10.1007/978-3-662-49313-7_13. URL: https://link-springer-com.uaccess.univie.ac.at/chapter/10.1007/978-3-662-49313-7_13 (besucht am 18.03.2025).

- [18] Marcel Marnitz. “Wie funktionieren eigentlich Fitnesstracker und was hat das mit Mathe zu tun?” de. Diss. Aachen: RWTH Aachen University, Sep. 2017. URL: <https://www.cammp.online/Fitness-thesis.pdf>.
- [19] Sakorn Mekruksavanich, Ponnipa Jantawong und Anuchit Jitpattanakul. “Effect of Sliding Window Sizes on Sensor-Based Human Activity Recognition Using Smart-watch Sensors and Deep Learning Approaches”. In: *2024 5th International Conference on Big Data Analytics and Practices (IBDAP)*. Aug. 2024, S. 124–129. DOI: 10.1109/IBDAP62940.2024.10689691. URL: <https://ieeexplore.ieee.org/document/10689691/?arnumber=10689691> (besucht am 10.03.2025).
- [20] Meinard Müller. *Fundamentals of Music Processing: Using Python and Jupyter Notebooks*. en. Cham: Springer International Publishing, 2021. ISBN: 978-3-030-69807-2 978-3-030-69808-9. DOI: 10.1007/978-3-030-69808-9. URL: <https://link.springer.com/10.1007/978-3-030-69808-9> (besucht am 28.02.2025).
- [21] Samanta Rosati, Gabriella Balestra und Marco Knaflitz. “Comparison of Different Sets of Features for Human Activity Recognition by Wearable Sensors”. en. In: *Sensors* 18.12 (Dez. 2018). Number: 12 Publisher: Multidisciplinary Digital Publishing Institute, S. 4189. ISSN: 1424-8220. DOI: 10.3390/s18124189. URL: <https://www.mdpi.com/1424-8220/18/12/4189> (besucht am 01.04.2025).
- [22] Rubén San-Segundo u. a. “Robust Human Activity Recognition using smartwatches and smartphones”. In: *Engineering Applications of Artificial Intelligence* 72 (Juni 2018), S. 190–202. ISSN: 0952-1976. DOI: 10.1016/j.engappai.2018.04.002. URL: <https://www.sciencedirect.com/science/article/pii/S0952197618300812> (besucht am 22.04.2025).
- [23] Nazanin Sedaghati, Sondos Ardebili und Ali Ghaffari. “Application of human activity/action recognition: a review”. en. In: *Multimedia Tools and Applications* (Jan. 2025). ISSN: 1573-7721. DOI: 10.1007/s11042-024-20576-2. URL: <https://link.springer.com/10.1007/s11042-024-20576-2> (besucht am 27.02.2025).
- [24] Luis Francisco Sigcha u. a. “Occupational Risk Prevention through Smartwatches: Precision and Uncertainty Effects of the Built-In Accelerometer”. In: *Sensors* 18 (Nov. 2018). DOI: 10.3390/s18113805.
- [25] Jan Stettner. *PCA - Eine Methode zur Dimensionsreduktion » Lamarr-Institut*. de-DE. Apr. 2023. URL: <https://lamarr-institute.org/de/blog/dimensionsreduktion-pca/> (besucht am 22.04.2025).
- [26] A. S. Abdull Sukor, A. Zakaria und N. Abdul Rahim. “Activity recognition using accelerometer sensor and machine learning classifiers”. In: *2018 IEEE 14th International Colloquium on Signal Processing & Its Applications (CSPA)*. März 2018, S. 233–238. DOI: 10.1109/CSPA.2018.8368718. URL: <https://ieeexplore.ieee.org/document/8368718> (besucht am 24.02.2025).
- [27] Weijun Tao u. a. “Gait Analysis Using Wearable Sensors”. en. In: *Sensors* 12.2 (Feb. 2012), S. 2255–2283. ISSN: 1424-8220. DOI: 10.3390/s120202255. URL: <https://www.mdpi.com/1424-8220/12/2/2255> (besucht am 24.02.2025).

- [28] Jindong Wang u. a. “Deep learning for sensor-based activity recognition: A survey”. In: *Pattern Recognition Letters*. Deep Learning for Pattern Recognition 119 (März 2019), S. 3–11. ISSN: 0167-8655. DOI: 10.1016/j.patrec.2018.02.010. URL: <https://www.sciencedirect.com/science/article/pii/S016786551830045X> (besucht am 24.05.2025).
- [29] Jialun Xue. “Human activity recognition using wearable sensors: a deep learning approach”. en. In: *IEEE 14th International Colloquium on Signal Processing Its Applications (CSPA)* (2020).
- [30] Delaram Yazdansepas u. a. “A Multi-featured Approach for Wearable Sensor-Based Human Activity Recognition”. In: *2016 IEEE International Conference on Healthcare Informatics (ICHI)*. Okt. 2016, S. 423–431. DOI: 10.1109/ICHI.2016.81. URL: <https://ieeexplore.ieee.org/document/7776399/?arnumber=7776399> (besucht am 01.04.2025).
- [31] Xu Yuntai. “Discrete Chaotic Fuzzy Neural Network (DC-FNN) Based Smart Watch Embedded Devices for Sports and Health Monitoring”. en. In: *Tehnicki vjesnik - Technical Gazette* 30.6 (Dez. 2023). ISSN: 13303651, 18486339. DOI: 10.17559/TV-20230222000374. URL: <https://hrcak.srce.hr/309228> (besucht am 04.03.2025).