HAW HAMBURG

BACHELOR THESIS
Jean Carlos Herzog Gomez

Entwicklung eines Multisensorsystems zur optischen Erkennung muskulärer Aktivität im Unterarm

FAKULTÄT TECHNIK UND INFORMATIK
Department Informatik

Faculty of Engineering and Computer Science
Department Computer Science

HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN HAMBURG
Hamburg University of Applied Sciences

Jean Carlos Herzog Gomez

Entwicklung eines Multisensorsystems zur optischen Erkennung muskulärer Aktivität im Unterarm

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung
im Studiengang *Bachelor of Science Informatik Technischer Systeme*
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. Roman Kusche
Zweitgutachter: Prof. Dr. Peer Stelldinger

Eingereicht am: 11. August 2025

Jean Carlos Herzog Gomez

Thema der Arbeit

Entwicklung eines Multisensorsystems zur optischen Erkennung muskulärer Aktivität im Unterarm

Stichworte

Gestenerkennung, Maschinelles Lernen, CNN, Klassifikation, Mustererkennung, HCI, Signalverarbeitung, Sensordatenverarbeitung, Optomyographie, Photoplethysmographie, PPG-Sensor, MicroPython, muskuläre Aktivität, Unterarm

Kurzzusammenfassung

Herkömmliche Elektromyographie-Verfahren zur Muskelaktivitätsmessung sind durch Klebeelektroden und elektrische Störanfälligkeit eingeschränkt. Die vorliegende Bachelorarbeit ist der Entwicklung eines nicht-invasiven Optomyographie-Systems zur optischen Erfassung von Unterarm-Muskelaktivität mittels acht handelsüblichen Photoplethysmographie-Sensoren und einem Raspberry Pi Pico gewidmet. Das System nutzt die Reflexionseigenschaften von rotem und infrarotem Licht an der Haut zur Detektion gewebebedingter Veränderungen bei Muskelkontraktionen. Durch CNN-basierte Live-Gestenerkennung werden sechs verschiedene Handgelenkgesten mit einer Genauigkeit von 99,56% unter kontrollierten Testbedingungen bei acht Probanden klassifiziert. Diese Arbeit zeigt, dass Optomyographie mit Einschränkungen als eine praktikable Alternative zur Elektromyographie-Methode umsetzbar ist. Das System trägt somit zur Forschung im Bereich von Prothesen-Steuerung und HCI-Anwendungen bei.

Jean Carlos Herzog Gomez

Title of Thesis

Development of an multi-sensor system for optical data analysis of muscular activity in the forearm

Keywords

Gesture Recognition, Machine Learning, CNN, Classification, Pattern Recognition, HCI, Signal Processing, Sensor Data Processing, Optomyography, Photoplethysmography, PPG Sensor, MicroPython, Muscle Activity, Forearm

Abstract

Conventional electromyography methods for measuring muscle activity are limited by adhesive electrodes and susceptibility to electrical interference. This work is dedicated to the development of a non-invasive optomyography system for optical detection of forearm muscle activity using eight commercial photoplethysmography sensors and a Raspberry Pi Pico. The system utilizes the reflection properties of red and infrared light on the skin to detect tissue-related changes during muscle contractions. Through CNN-based live gesture recognition, six different wrist gestures are classified with an accuracy of 99.56% under controlled test conditions with eight subjects. This work demonstrates that optomyography can be implemented as a viable alternative to electromyography methods, albeit with certain limitations. The system thus contributes to research in the field of prosthetic control and HCI applications.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Akronymverzeichnis

CNN Convolutional Neural Network

ECG Elektrokardiogramm

EEG Elektroenzephalogramm

EIM elektrische Impedanzmyographie

EMG Elektromyographie

EOT End of Transmission

FFT Fast-Fourier-Transformation

GPIO General Purpose Input/Output

HAL Hardware Abstraction Layer

HAW Hochschule für Angewandte Wissenschaften

HCI Human Computer Interface

MCU Microcontroller Unit

MMG Mechanomyogramm

MVC Model-View-Controller

NIR Nahinfrarot

OMG Optomyogaphie

PPG Photoplethysmographie

ReLU Rectified Linear Unit

REPL Read-Eval-Print-Loop

SDA Serial Data Line

SDL Serial Clock Line

SWN Sliding Window Normalisation

1 Einleitung

Die Steuerung von Prothesen und Computer-Interfaces durch Muskelsignale ist ein aktives Forschungsfeld mit erheblichem praktischen Potenzial. Während etablierte Messverfahren bereits erfolgreich eingesetzt werden, bestehen weiterhin Herausforderungen hinsichtlich Alltagstauglichkeit und Anwenderkomfort [23]. Diese Arbeit untersucht das Optomyographie-Verfahren als alternativen Ansatz zur Erfassung von Muskelaktivität mittels optischer Sensoren am Unterarm. Die folgenden Abschnitte erläutern den aktuellen Forschungsstand, die spezifische Motivation für optische Verfahren und die konkreten Ziele der vorliegenden Arbeit.

1.1 Stand der Forschung

Ein klassischer Ansatz zur Messung an Muskeln basiert darauf, zwei Elektrodenpaare am Unterarm zu platzieren, die Signale von entgegengesetzten Muskeln erfassen. So kann beispielsweise das Öffnen und Schließen einer Hand untersucht werden [30]. Während Machine-Learning- und Mustererkennung-Methoden unter Laborbedingungen solide Ergebnisse zeigen, sind Versuche unter alltäglichen, nicht-stationären Bedingungen problematisch durchzuführen [30].

Die Verwendung von Elektromyographie (EMG) allein oder in Kombination mit elektrischer Impedanzmyographie (EIM) zur Steuerung von Handprothesen wurde in mehreren Forschungarbeiten [19, 31] untersucht. Während EMG-basierte Klassifikatoren bereits gute Ergebnisse für verschiedene Armpositionen erzielen, zeigen die Untersuchungen, dass die Kombination beider Verfahren die Klassifikationsleistung deutlich verbessert. Die Integration von EIM-Features führt zu einer merklichen Reduzierung der Klassifikationsfehler, wobei dieser Effekt besonders bei freien Handbewegungen erkennbar ist [19]. EMG und EIM ergänzen sich gegenseitig bei der Erfassung verschiedener Aspekte der Muskelaktivität. Allerdings muss berücksichtigt werden, dass EIM-Signale eine stärkere Sensitivität gegenüber Armpositionen als EMG-Signale aufweisen. Dies macht bei

der Systemauslegung entsprechende Trainingsstrategien mit mehreren Armpositionen erforderlich [19]. Ergänzend zeigen auch Ansätze mit EMG- und Inertialsensoren, welche Beschleunigung und Orientierung im Raum erfassen, vielversprechende Ergebnisse mit hohen Klassifikationsgenauigkeiten bei Prothesensteuerungsaufgaben [48].

Das Mechanomyogramm (MMG)-Verfahren, eine andere Art der Erfassung von Muskelkontraktionen, dient zur mechanischen Untersuchung der Muskelaktivität. Muskelfaserzuckungen erzeugen mechanische Schwingungen basierend auf verschiedenen Mechanismen. Analysen mittels beschleunigungsbasierter MMG-Messungen können Einblicke in Ermüdungsprozesse bei Muskelarbeit ermöglichen. Für praktische Anwendungen erweist sich MMG jedoch als problematisch, da die Methode sehr empfindlich gegenüber Bewegungsartefakten und Störungen ist, was zuverlässige Messungen in realen Anwendungsszenarien erschwert [52].

Optomyographie (OMG), als nicht-invasive Messtechnik, verwendet Nahinfrarot (NIR) - Photoelektriksensoren zur Messung von Hautoberflächenverschiebungen, die durch Muskelaktivität verursacht werden. In [20] werden zwei wesentliche Komponenten zur Untersuchung eingesetzt: eine NIR-emittierende Diode (940 nm Spitzenwellenlänge) und ein hochempfindlicher Phototransistor (800 nm maximale Empfindlichkeit) mit integriertem Filter für sichtbares Licht. Nach [9] gilt der Bereich um 800 nm als optimal für die Messung von Oberflächenveränderungen der oberen 2 mm bis 3 mm dicken Hautschicht. In [41] und [20] werden bis zu fünf Sensoren zur Auswertung der Muskelkontraktionen eingesetzt. Anhand der Messergebnisse konnten 14 unterschiedliche Handgesten gemessen werden. Zum Beispiel das Bewegen des Handgelenks und der Finger sowie das Öffnen und Schließen der Hand. Signale mussten dafür mit Tiefpassfiltern von 20 Hz verarbeitet werden, um hochfrequente Störsignale zu beseitigen.

1.2 Motivation

Im Bereich der Erfassung von Muskelaktivität bietet das Verfahren der EMG Vorteile durch seine hohe Sensitivität, direkte Messung elektrischer Signale und schnelle Reaktionszeiten [59]. Auf der anderen Seite bringt es auch einige praktische Nachteile mit sich. Klebeelektroden müssen auf der Haut angebracht werden und elektrische Störungen können die Messungen beeinflussen. Als Methode ist sie zudem stark von der körpereigenen Energieerzeugung abhängig, wodurch nur aktive Muskelkontraktionen erfasst werden können. Passive Bewegungen, wie beispielsweise das Heruntergleiten eines Arms

vom Tisch, bleiben unerkannt [19]. Eine weitere Herausforderung stellt die Unberechenbarkeit des EMG-Signals im Frequenzbereich zwischen 20 Hz und 200 Hz dar, da typische Bewegungsartefakte in einem ähnlichen Frequenzbereich auftreten [31].

EIM bietet durch die Messung zeitlicher Veränderungen der Bioimpedanz einen alternativen Ansatz zur Muskelaktivitätsdetektion. EIM basiert auf der Anwendung und Messung hochfrequenter, niedrigintensiver elektrischer Ströme zur nicht-invasiven Bewertung von Muskeln, wobei die resultierenden Spannungen gemessen werden [19]. Im Gegensatz zu mechanischen Störungen verursachen tatsächliche Muskelkontraktionen spezifische Veränderungen in der Phasenresponse: negativere Phasen bei niedrigen Frequenzen und positivere Phasen bei höheren Frequenzen [31]. EIM untersucht die Veränderungen der elektrischen Impedanz während Muskelaktivitäten und kann im Vergleich zu EMG potenziell resistenter gegen Artefakte sein, da es Informationen nicht nur über die elektrische Aktivität der Muskelfasern, sondern auch über die Verdickung und Bewegung der Muskeln selbst liefert [48]. Allerdings zeigt EIM ähnliche Herausforderungen wie EMG bezüglich der praktischen Anwendung, insbesondere eine stärkere Beeinflussung durch Änderungen der Armposition als EMG [19].

Messungen an Muskeln anhand der Lichtreflexion, auch als OMG bekannt, ist ein alternativer Ansatz, welcher dieser Problematik entgegenwirken kann [41]. Durch die Messung optischer Gewebeeigenschaften wird auf direkten Hautkontakt mit Klebeelektroden verzichtet und somit elektrische Störanfälligkeiten eliminiert [20]. Interessant ist dabei der Einsatz handelsüblicher Sensoren, die das System sowohl technisch als auch wirtschaftlich attraktiv machen. Die Entwicklung fortschrittlicher Prothesen und intuitiver Human-Computer-Interface (HCI)-Systeme kann dadurch begünstigt werden [41, 21].

1.3 Ziel der Arbeit

Ziel dieser Arbeit ist die Entwicklung eines Systems zur optischen Erfassung von Muskelaktivität im Unterarm unter Berücksichtigung der zur Verfügung gestellten Mitteln des *Medical Sensor Labs*. Das entwickelte Sensorsystem soll in der Lage sein, Muskelaktivität im Unterarm optisch zu detektieren und Messdaten für weitere Untersuchungen zu liefern. Mittels Signalverarbeitung und Convolutional Neural Network (CNN)-basierte Gestenerkennung soll gezeigt werden, dass OMG eine Alternative oder Ergänzung zum EMG-Verfahren bieten kann. Die Reflexionseigenschaften von rotem und infrarotem Lichtspektrum stehen im Mittelpunkt. Dafür steht unter den verfügbaren Mitteln der PPG-Sensor

MAX30102 [38]. Dieser Sensor kann die Reflexionen des roten und infraroten Lichtspektrums erfassen, um Veränderungen am Gewebe zu messen.

1.4 Aufbau der Arbeit

Die vorliegende Arbeit gliedert sich in sechs Hauptkapitel. Kapitel 2 legt die theoretischen Grundlagen, indem zunächst die biologischen und medizinischen Aspekte optischer Biosensorik erläutert werden, einschließlich dem anatomischen Aufbau der Haut und des Unterarmes. Anschließend werden die technischen Grundlagen sowie Methoden der Datenverarbeitung, insbesondere Median-Glättung, Normalisierung und CNNs, behandelt.

Kapitel 3 beschreibt die methodische Herangehensweise, beginnend mit den Anforderungen an das System und dem entwickelten Konzept. Es folgt die Darstellung der verwendeten Hardware sowie der Konstruktion des Armbandes. Abschließend werden die Implementierung von Firmware und Software sowie die Signalverarbeitung und Gestenerkennung mittels CNNs vorgestellt.

In Kapitel 4 wird die Charakterisierung des Systems durchgeführt, wobei unter anderem Einstellungsparameter, Stabilitätsmessungen und Signalverifikation untersucht werden.

Kapitel 5 präsentiert die durchgeführten Probandenmessungen und die daraus resultierenden Ergebnisse. Die charakteristischen Merkmale der Datensätze werden analysiert und die Performance des entwickelten Klassifikationsmodells evaluiert.

In Kapitel 6 werden die Messergebnisse und die Systemperformance diskutiert sowie die wichtigsten Erkenntnisse zusammengefasst. Dieses Kapitel geht auf die Erfüllung der definierten Anforderungen ein und gibt einen Ausblick auf mögliche Weiterentwicklungen des Systems.

2 Grundlagen

Dieses Kapitel stellt die theoretischen Grundlagen aus Medizin, Biologie, Signalverarbeitung und technische Aspekte dar, die für die Entwicklung und Bewertung der optischen Messmethoden notwendig sind. Die hier zusammengefassten Erkenntnisse bilden die Basis für die in den folgenden Kapiteln beschriebene Methodik und Ergebnisse.

2.1 Biologische und medizinische Aspekte

Biologische Aspekte umfassen die physiologischen Eigenschaften und Prozesse des menschlichen Körpers, die für diese Arbeit relevant sind, wie beispielsweise Herzfrequenz, Blutdruck, Hautleitfähigkeit oder Bewegungsmuster. Medizinische Aspekte beziehen sich auf gesundheitsbezogene Überlegungen bei der Datenerfassung, einschließlich Sicherheitsstandards, ethischer Richtlinien und der Interpretation von Messwerten im Kontext der Gesundheit. Diese Arbeit befasst sich mit der Messung biologischer Eigenschaften an Menschen und muss verschiedene Aspekte beachten, die nicht zur Informatik gehören. In diesem Abschnitt werden grundlegende Informationen zusammengefasst, die relevant für die Durchführung dieser Arbeit sind [42].

2.1.1 Sicherheitsaspekte medizinischer Sensorsysteme

Bei der Entwicklung und Anwendung medizinischer Sensorsysteme stellt die Betriebssicherheit einen zentralen Aspekt dar. Die Gefährdung durch elektrischen Strom erfordert besondere Beachtung, da dieser sowohl Verbrennungen verursachen als auch erregbares Gewebe wie Nerven und Muskeln stimulieren kann. Im Bereich der biomedizinischen Telemetrie und bei stromabhängigen biomedizinischen Geräten müssen daher umfassende Sicherheitsmaßnahmen implementiert werden, die elektrische, wärmebezogene und

allgemeine Sicherheitsaspekte berücksichtigen [42]. Sensoren, wie der Photoplethysmographie (PPG)-Sensor MAX30102, sind für die Verwendung an Menschen in Consumer-Elektronik und Fitness-Anwendungen konzipiert worden und erfüllen die Anforderungen für Wearable Devices und Smartphones, jedoch macht das Datenblatt [38] keine Angaben zur Erfüllung der medizinischen Sicherheitsstandards des IEC 60601. Da der Fokus dieser Arbeit auf der technischen Implementierung und Datenverarbeitung liegt und keine medizinischen Messungen für diagnostische Zwecke durchgeführt werden, ist die Verwendung dieses Sensors für die Zielsetzung angemessen. Wenn Daten vom System per USB-Schnittstelle übertragen werden sollen, so muss ein medizinischer USB-Isolator wie der ISOUSB-Cable-A der IFTOOLS GmbH als Sicherheitsvorkehrung verwendet werden. Dieser USB-Isolator dient dazu, eine galvanische Trennung zwischen dem Messsystem und dem PC zu schaffen, um zu verhindern, dass elektrische Ströme vom Computer über die USB-Verbindung zum Probanden fließen können. Dies ist eine wichtige Sicherheitsmaßnahme beim Umgang mit Biosignalen am Menschen. Das verwendete Gerät ist nach EN 60601-1 zertifiziert [25], einer europäischen Norm für medizinische elektrische Geräte, die Sicherheitsanforderungen zum Schutz von Patienten, Anwendern und Umgebung definiert, und gewährleistet eine Isolationsfestigkeit von 4kVrms.

2.1.2 Photoplethysmographie - Grundlagen der optischen Biosensorik

Photoplethysmographie ist eine optische Methode der Plethysmographie, die zum Erkennen von Blutvolumenänderungen im Gewebe verwendet wird [44]. Plethysmographie bezeichnet allgemein die Techniken zur Messung von Volumen und den damit verbundenen Veränderungen in Körperteilen. PPG ermöglicht Messungen eines Organs mittels eines optischen Sensors, der typischerweise aus einer infrarot-emittierenden LED und einem Photodetektor besteht.

Das PPG-Signal kann durch das Anbringen eines Pulsoximeter-Geräts an der Hautoberfläche nicht-invasiv aufgezeichnet werden [44]. Pulsoximeter dienen zur Überwachung der Durchblutung der Haut und des subkutanen Gewebes [42]. Das Messprinzip des Pulsoximeters basiert auf der Messung der Sauerstoffsättigung des Hämoglobins. Der Sensor, der normalerweise am Finger oder Ohrläppchen platziert wird, sendet Licht zweier unterschiedlicher Wellenlängen aus. Die Sauerstoffsättigung wird anhand der variierenden Lichtabsorption durch das pulsierende Blut bestimmt [42].

Die physikalischen Prinzipien beruhen auf der Wechselwirkung von Licht mit biologischen Geweben, was zu Reflexion, Streuung und Absorption führt. Bei der Wellenlängenauswahl für biomedizinische Anwendungen sind physiologische Faktoren entscheidend. Während ultraviolettes Licht bei längerer Exposition gesundheitsschädliche Wirkungen zeigen kann, weist Infrarotlicht im Bereich von 700 bis 1200 nm deutlich geringere Effekte auf [2]. Dieser Spektralbereich zeichnet sich durch eine minimale optische Absorption im Körpergewebe aus. Außerhalb dieses optimalen Fensters dominiert die Absorption durch Hämoglobin (< 700 nm) beziehungsweise Wasser (> 1200 nm) [42]. Melanin absorbiert stark Licht kürzerer Wellenlängen [2], während rotes und nahinfrarotes Licht leicht wasserreiches Gewebe durchdringen kann [7]. PPG-Sensoren verwenden typischerweise Infrarot-Wellenlängen und können sowohl im Transmissions- als auch im Reflexionsmodus betrieben werden [2]. Klinische Anwendungen der PPG umfassen die Messung der Blutsauerstoffsättigung, Schätzung der Herzfrequenz, Blutdruckmessung und Diagnose von Arterienerkrankungen [44]. PPG-Sensoren enthalten Halbleiter-Photodetektoren, die durch mechanischen Stress aufgrund des piezoresistiven Effekts ihre elektrischen Eigenschaften verändern. Hierbei kann geringer Druck (MPa-Bereich) bereits zu Widerstandsänderungen führen [6]. Ebenso kann Druck auf die PPG-Sensor-Platine mechanische Verformungen verursachen, die Widerstandsänderungen in Leiterbahnen, Mikrorisse in Lötstellen und Verschiebungen von elektronischen Bauteilen zur Folge haben können [32].

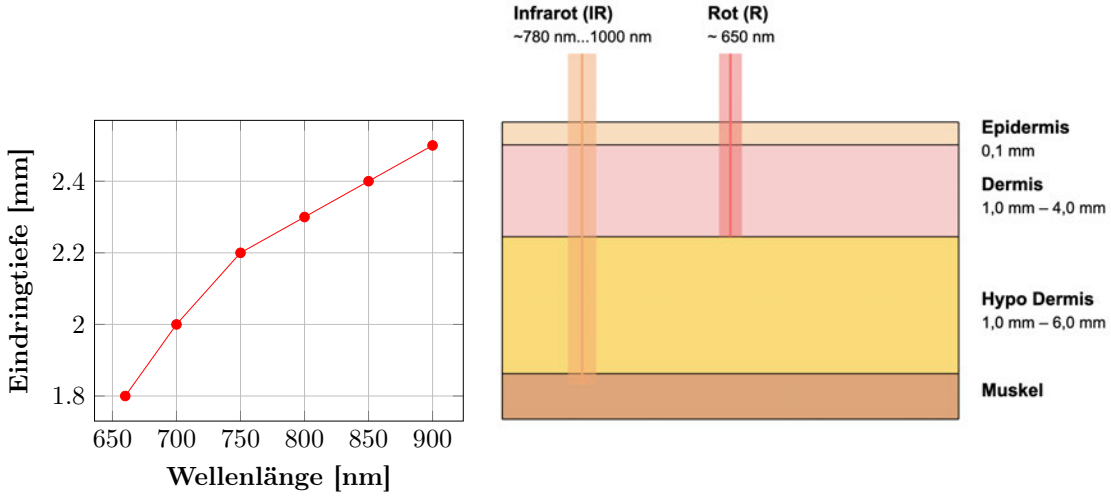
2.1.3 Anatomischer Aufbau und optische Eigenschaften der Haut

Verschiedene Publikationen gehen auf die Lichtdurchlässigkeit der Haut ein, wie zum Beispiel [7, 21, 53, 5]. In Abbildung 2.1a sind experimentell ermittelte Eindringtiefen nach [7] dargestellt. Die Eindringtiefe von Licht ins Gewebe ist wellenlängenabhängig. Im roten und infraroten Spektralbereich dringen längere Wellenlängen tiefer in die Haut ein [2]. Die menschliche Haut weist einen charakteristischen Schichtaufbau auf, wie in Abb. 2.1b dargestellt. Nach [7] besteht sie von der Oberfläche ausgehend aus folgenden drei Schichten:

- Epidermis (0,1 mm dick, blutfreie Schicht)
- Dermis (1 bis 4 mm dick, vaskularisierte Schicht)
- Hypodermis (1 bis 6 mm dick, subkutanes Fett abhängig von der Körperstelle)

Die Haut ist ein komplexes heterogenes Medium, in dem der Blut- und Pigmentgehalt räumlich variabel in der Tiefe verteilt sein kann. Somit lassen sich für den Blut- und Pigmentgehalt nur Näherungswerte darstellen und ermitteln, da die Verteilung nicht homogen ist und von Person zu Person sehr unterschiedlich sein kann. Die Breite der verschiedenen Schichten kann je nach Person stark variieren [57]. Studien wie [26, 27] suggerieren, dass rotes und infrarotes Licht zwischen 1 cm und 3 cm durch die Haut bis zum Muskel dringen können.

Rotes Licht mit einer Wellenlänge von ca. 650 nm kann bis zur Dermis-Schicht durchdringen, während infrarotes Licht (ab ca. 780 nm) bei günstigen Bedingungen wie geringer Hautdicke und niedrigem Melaningehalt durch die Hypodermis bis zum Muskel durchdringen kann [2]. Unter ungünstigen Bedingungen erreicht infrarotes Licht möglicherweise nicht die Muskelschicht (siehe Abb. 2.1b). Obwohl Messungen am Muskel präziser sind, können auch an den Hautschichten in Muskelnähe aussagekräftige Messungen gewonnen werden [53, 57].



(a) Eindringtiefe in Abhängigkeit von der Wellenlänge

(b) Geschätzte Lichtdurchlässigkeit der verschiedenen Hautschichten

Abbildung 2.1: Lichtdurchdringung durch Hautschichten in Abhängigkeit der Wellenlänge. (a) Eindringtiefe des Lichtes durch die Haut abhängig von der Wellenlänge anhand der experimentellen Werte nach [7]. (b) Darstellung der verschiedenen Hautschichten und ihre geschätzte Lichtdurchlässigkeit nach [21].

2.1.4 Bewegungen und Muskulatur des Unterarmes

Der Schwerpunkt dieser Arbeit liegt auf der oberflächlichen Unterarmmuskulatur (siehe Abb. 2.2). Nach [43] kann der Unterarm vier Hauptbewegungen ausführen. Flexion und Extension beschreiben das Heranführen und Strecken des Unterarms zum Oberarm. Pronation und Supination beschreiben das Drehen des Unterarmes, sodass die Handfläche nach unten oder oben zeigt. Diese Bewegungen führen zu Änderungen in der räumlichen Anordnung der Unterarmmuskulatur [39].

Am Handgelenk erzeugen verschiedene Muskeln Kombinationen aus Beugung oder Streckung sowie seitlichen Bewegungen (Abduktion oder Adduktion). Flexoren (Beugemuskeln) befinden sich überwiegend an der Vorderseite des Unterarms, während Extensoren(Streckmuskeln) hauptsächlich an der Rückseite des Unterarms liegen [43].

Die verantwortlichen Muskelgruppen für diese Bewegungen sind in [36] beschrieben. Der *Flexor carpi radialis* und *-ulnaris* beugen das Handgelenk, der *Flexor digitorum superficialis* und *-profundus* die Finger und der *Flexor pollicis longus* den Daumen.

Der *Extensor carpi radialis* und *-ulnaris* strecken das Handgelenk, der *Extensor digitorum* die Finger, der *Extensor pollicis longus* und *-brevis* den Daumen. Der *Anconeus* ist ein dreieckiger Muskel am Ellenbogen, der zur Streckung des Ellenbogengelenks beiträgt und dieses während der Unterarmbewegungen stabilisiert [43, 36].

Die Muskeln des Unterarms ermöglichen auch seitliche Bewegungen des Handgelenks. Abduktion (Bewegung zur Daumenseite) wird durch *Flexor carpi radialis* und *Extensor carpi radialis* ausgeführt. Adduktion (Bewegung zur Kleinfingerseite) wird durch *Flexor carpi ulnaris* und *Extensor carpi ulnaris* bewirkt [43, 36].

In Abbildung 2.3 kann der Querschnitt des Unterarms betrachtet werden. Am äußeren Rand sind die Hautschichten zu erkennen, die in 2.1.3 beschrieben werden. Die an der Haut anliegenden Muskeln sind für Verformungen bei Bewegung, die in dieser Arbeit analysiert werden, verantwortlich [39].

Muskelermüdung kann bei anhaltenden Kontraktionen zu einer verminderten Krafterzeugung führen. Nach [52] wird Ermüdung als Reduktion der maximalen willkürlichen Kontraktionskraft definiert, unabhängig von der ausgeführten Aufgabe. Charakteristisch für den ermüdeten Muskel sind eine verlangsamte Spannungsentwicklung sowie eine prolongierte Relaxationsphase, die von biochemischen Veränderungen begleitet wird. Die

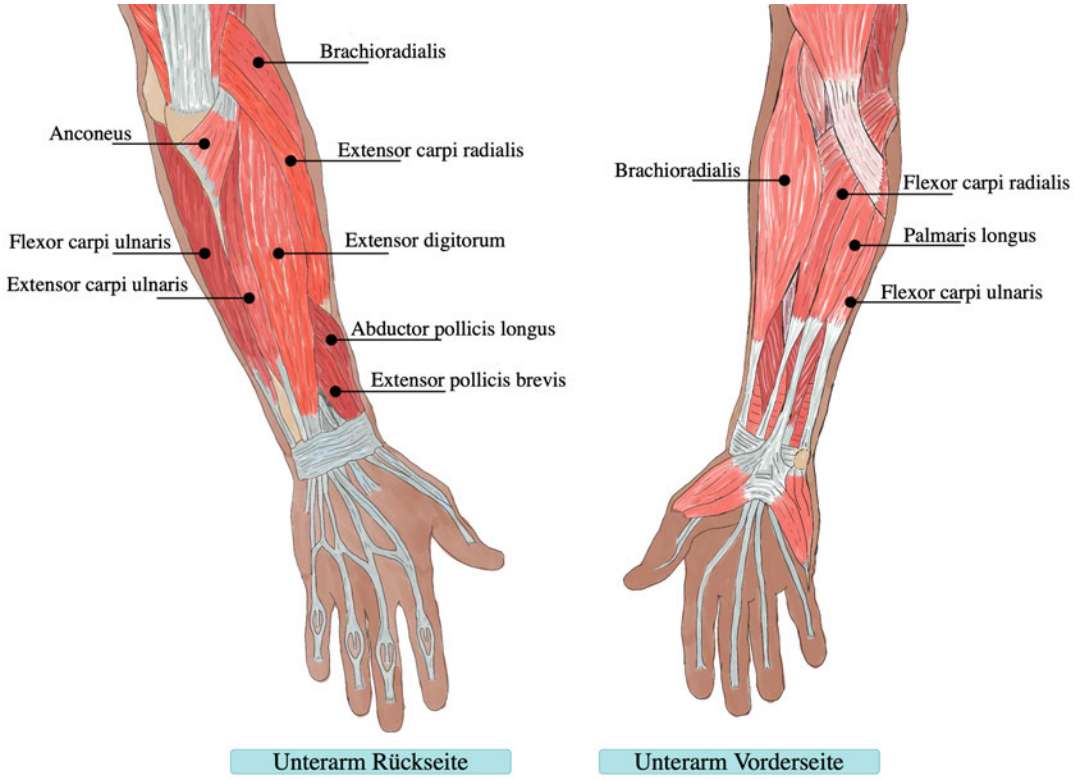


Abbildung 2.2: Oberflächliche Muskelschicht des rechten Unterarms. Auf der Rückseite (links dargestellt) des Unterarms befinden sich hauptsächlich die Extensor-Muskelgruppen und auf der Vorderseite (rechts dargestellt) hauptsächlich die Flexor-Muskelgruppen. Blutgefäße und die meisten Knochen sind in dieser Darstellung nicht enthalten. Vorlage entnommen von [36]

Ausdauergrenze, um motorische Aufgaben aufrechtzuerhalten, ist in dieser Arbeit bei den Probandenmessungen zu beachten.

Für biomedizinische Anwendungen weisen Muskelkontraktionen primär Frequenzkomponenten im Bereich von 0 Hz bis 10 Hz auf, während Frequenzen oberhalb dieses Bereichs als Störsignale oder Artefakte klassifiziert werden können [21]. Untersuchungen von 24 verschiedenen Alltagstätigkeiten wie zum Beispiel Körperpflege, Küchentätigkeiten und Schreibaufgaben zeigen bei Handgelenkbewegungen durchschnittliche Frequenzen von etwa 1 Hz. Handgelenkbewegungen zeigen einen Frequenzbereich von 0,48 bis 12,46 Hz, wobei sich die Mehrzahl (75%) unter 5 Hz konzentriert. Dabei haben Flexion und Extension eine Hauptfrequenz von 1,06 Hz mit einem Bereich von 0,52 bis 2,47 Hz. Die Radial- und Ulnarabweichung liegt bei 1,11 Hz im Bereich von 0,48 bis 2,25 Hz [35]. Die-

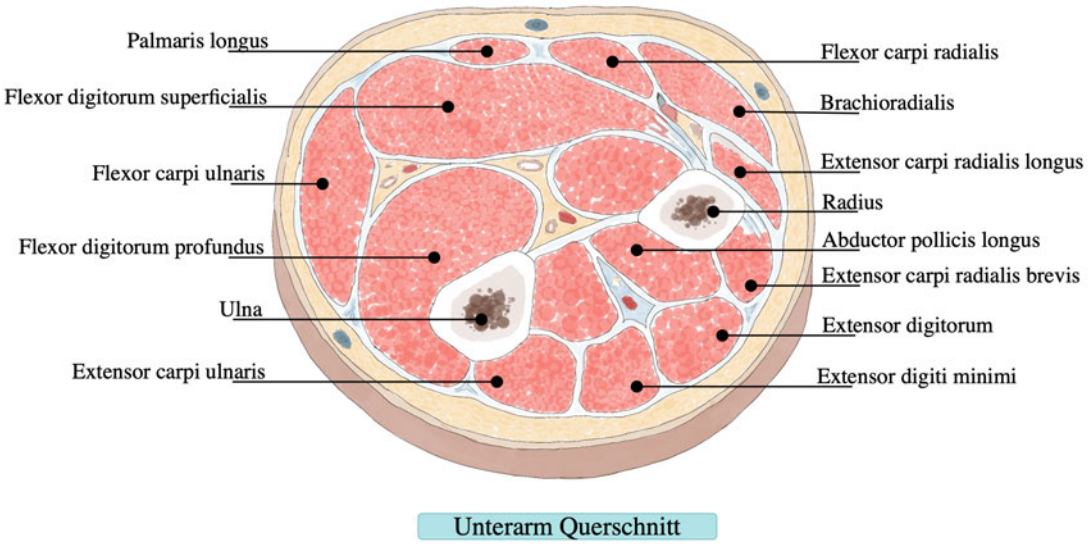


Abbildung 2.3: Querschnitt des rechten Unterarmes. Die Vorderseite des Arms zeigt nach oben. Alle Muskelschichten sind erkennbar, jedoch nur die oberflächlichen Muskeln sind beschriftet, die für diese Arbeit relevant sind. Vorlage entnommen von [36].

se Befunde werden durch Untersuchungen der Armresonanz bestätigt, die eine natürliche Resonanzfrequenz von etwa 0,98 Hz zeigen [55]. Zudem bewegen sich Arme bei funktionalen Aktivitäten hauptsächlich in einem Frequenzbereich von 0,8–1,1 Hz [55], was die charakteristische 1 Hz-Frequenz des Handgelenks unterstützt.

Verzögerungen bei der Steuerung von myoelektrischen Prothesensystemen sind akzeptabel im Bereich von 50 ms bis 400 ms [15]. Dabei sind Verzögerungen unter 300 ms für Benutzer kaum wahrnehmbar [13, 14, 17]. Praktische Implementierungen folgen diesem Ansatz, indem Echtzeit-Controller mit dem spezifischen Ziel entwickelt werden, die Verzögerung unter 300 ms zu halten, um negative Auswirkungen auf die Systemleistung zu vermeiden [10, 22].

2.1.5 Optomyographie

Optomyographie basiert auf der Übertragung von Licht ins Gewebe mittels LEDs und der Erfassung des reflektierten Lichtes durch Photodioden (siehe Abb. 2.4) [39]. In [41, 20, 21] werden Änderungen in den optischen Eigenschaften des Gewebes bei einer Verformung der Hautschichten festgestellt. Wie im Abschnitt 2.1.3 beschrieben ist die Verwendung

von rotem Licht aufgrund seiner hohen Eindringtiefe in menschliches Gewebe vorteilhaft. Obwohl das Licht die Muskelschichten nicht vollständig durchdringt, können durch die richtungsabhängige Lichtstreuung der Muskelfasern bei Kontraktionen reproduzierbare Signalmuster an der Hautoberfläche detektiert werden. Hierbei unterscheiden sich die Signalformen von isotonischen und isometrischen Kontraktionen deutlich voneinander. Bei isotonischen Kontraktionen verkürzt sich der Muskel bei konstanter Kraft, bei isometrischen Kontraktionen bleibt die Muskellänge konstant während die Spannung erhöht wird. Diese unterschiedlichen Signalmuster ermöglichen eine präzise Klassifizierung verschiedener Muskelkontraktionstypen [5].

Die Herausforderungen bei OMG sind die begrenzte Eindringtiefe des Lichts, da nur Wellenlängen von 600 bis 1000 nm ausreichend tief in Gewebe durchdringen können, sowie die Absorption durch verschiedene Gewebeschichten. Zusätzlich erschweren Veränderungen der Sensorposition und räumliche Beziehungen zwischen Muskeln und Haut eine zuverlässige Messung [39]. Eine größere Entfernung zwischen Lichtquelle und Detektor führt zusätzlich zu einem schlechteren Signal-Rausch-Verhältnis [5].

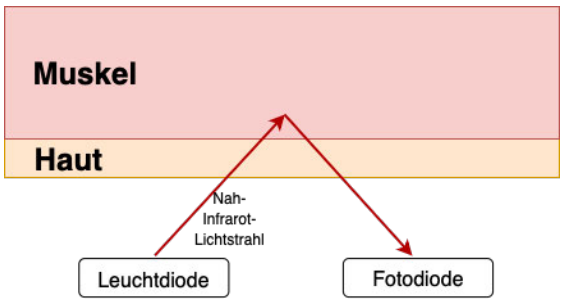


Abbildung 2.4: Einfache Darstellung des OMG-Prozesses. Leuchtdioden erzeugen Lichtstrahlen im Nah-Infrarot Bereich, die durch die Haut dringen. Die Reflexion dieser Strahlen werden von Fotodioden erfasst. Vorlage entnommen von [41].

2.2 Laufzeitumgebung des Mikrokontrollers

In dieser Arbeit wird MicroPython als Laufzeitumgebung verwendet. MicroPython ermöglicht die Implementierung von Python-Code auf höherer Ebene auf ausgewählten Mikrocontrollern. Zudem kommt MicroPython ohne Kompilierung aus und hat umfangreiche Bibliotheken. Im Vergleich zu C und C++, die den Zugriff auf Hardware-Eigenschaften des Mikrocontrollers auf niedriger Ebene ermöglichen, ist bei MicroPython

die Geschwindigkeit nicht die oberste Priorität [60]. Die Laufzeitumgebung kann die primäre UART-Schnittstelle für den asynchronen seriellen Datenaustausch verwenden. Das Verfahren wird als Read-Eval-Print-Loop (REPL) bezeichnet und ermöglicht es dem Programmierer, über die Kommandozeile mit der Hardware zu interagieren [60]. Die Übertragungsgeschwindigkeit per USB-Schnittstelle ist von MicroPython über die Baudrate einstellbar, welche die Anzahl der Symbole pro Sekunde in der UART-Kommunikation definiert. Bei der USB-REPL-Verbindung zwischen Computer und Mikrocontroller entspricht ein Symbol genau einem Bit, wodurch beispielsweise 115200 Baud einer Bitrate von 115200 Bit pro Sekunde entspricht. Die effektive Datenrate ist jedoch niedriger aufgrund des Protokoll-Overheads. Für die am PC eingehenden Daten stellt der virtuelle USB-Serial-Treiber einen Eingangsdatenpuffer bereit [46]. MicroPython-Firmware kann auf Mikrocontrollern im Flash-Speicher installiert werden [60]. Die Schnittstelle zur Hardware-Ebene wird über das Modul `machine` bereitgestellt [46]. Die Klasse `Pin` dient der Verknüpfung des Programms mit den General Purpose Input/Output (GPIO) Pins des Mikrocontrollers. I2C kann als Kommunikationsprotokoll mit Taktleitung, auch bekannt als Serial Clock Line (SCL), und Datenleitung, auch bekannt als Serial Data Line (SDA), für adressbasierten Datenaustausch zwischen Mikrocontroller und Bauteilen verwendet werden [46]. Das System arbeitet in eine Controller-Target-Architektur mit Kollisionsvermeidung und kann als Hardware oder Software implementiert werden [24]. Das Protokoll verwendet Start-Signal, 7-Bit-Adresse, Read/Write-Bits und byteweise Datenübertragung mit Acknowledge. `Software-I2C` kann an den meisten Pins eines Mikrocontrollers verwendet werden, um die erforderlichen Lese- und Schreibvorgänge durchzuführen. Mit der Klasse `Timer` ist es möglich, einen Timer zu starten, der periodisch eine Callback-Funktion aufrufen kann, wenn die Zeit abgelaufen ist [46].

2.3 Datenaufbereitung und CNN-Grundlagen

Die Datenvorverarbeitung ist fundamental für die Leistungsfähigkeit von Mustererkennungsalgorithmen in Anwendungen. Da physiologische Signale typischerweise von Rauschen und Artefakten überlagert sind, können Filterverfahren zur Rauschunterdrückung und Normalisierungsmethoden zur Standardisierung der Signaleigenschaften die Erkennungsgenauigkeit verbessern [51]. Die folgenden Abschnitte behandeln den Moving Median als robustes Glättungsverfahren, Normalisierung zur Skalierung der Eingangsdaten sowie CNN-Grundlagen.

2.3.1 Median-Glättung

Der Moving Median ist ein Glättungsverfahren und stellt eine nichtlineare Methode zur Signalverarbeitung dar. Das Prinzip basiert auf der lokalen Medianberechnung innerhalb eines gleitenden Zeitfensters der Größe N. Für eine diskrete Zeitsequenz $\{X[n]\}_{n\in\mathbb{Z}}$ ist der Moving Median definiert als:

$$Y[n] = \text{MEDIAN}\{X[n+k] : k \in [-N_L, N_R]\} \tag{2.1}$$

wobei N_L und N_R die Anzahl der Samples links und rechts des zentralen Samples bezeichnen und $N = N_L + N_R + 1$ die Fenstergröße darstellt [3, 16]. Bei symmetrischen Fenstern gilt $N_L = N_R$. Als nichtlineares System können lineare Analysemethoden nicht in gewohnter Weise angewendet werden [40, 3]. Trotz seiner nichtlinearen Natur weist das Verfahren Tiefpasscharakteristika auf, da hochfrequente Schwankungen durch die Medianbildung geglättet werden [40]. Verglichen mit linearen Filtern, wie zum Beispiel dem Moving Average, die mit Durchschnittsbildung arbeiten und optimal für die Reduzierung von Gauß'schem Rauschen sind, sortiert der Moving Median die Werte und wählt den mittleren Wert aus [40]. Diese grundlegend verschiedene Arbeitsweise macht ihn gegenüber Ausreißern und Störimpulsen widerstandsfähig und erhält dabei die Eigenschaften des ursprünglichen Signals besser, insbesondere im Falle von sprunghaften Änderungen und Kanten [4].

Der Medianfilter kann in der Analyse biomedizinischer Signale wie Elektroenzephalogramm (EEG), Elektrokardiogramm (ECG) und PPG eingesetzt werden, da diese Signale oft Ausreißer und Störungen enthalten, die das Signal verfälschen können. [16]. PPG-Signale enthalten kardiovaskuläre Informationen, weisen jedoch eine geringe Amplitude und niedrige Frequenz auf, wodurch sie störungsanfällig sind [34].

Bei der praktischen Anwendung stellt die Wahl der optimalen Fenstergröße einen Kompromiss zwischen Rauschunterdrückung und Signalerhaltung dar [4]. Während größere Fenster eine stärkere Glättung und bessere Unterdrückung von Impulsstörungen bewirken, können sie gleichzeitig wichtige Signalmerkmale verwischen und zu stärkerer Signalverzögerung sowie Verzerrung bei harmonischen Signalen führen [40, 3]. Experimentelle Validierung mit realen Daten ist daher für die Parameteroptimierung unerlässlich [4]. Die Ergebnisse der Median-Glättung können mittels der Fast Fourier Transformation (FFT) überprüft werden. Mit der FFT-Methode ist es möglich die Frequenzen der Signale zu untersuchen und die Auswirkungen der gewählten Fensterbreite [58, 8, 1].

In der medizinischen Bildverarbeitung findet die Median Glättung vielfältige Anwendung. Sie dient insbesondere der Rauschunterdrückung in CT-, MRT- und Röntgenaufnahmen sowie als essentieller Vorverarbeitungsschritt für die Segmentierung anatomischer Strukturen [16, 47].

2.3.2 Normalisierung

Die Skalierung und Transformation von Daten ist ein essentieller Schritt in der Datenverarbeitung. Die Min-Max-Normalisierung skaliert Daten auf einen definierten Bereich, typischerweise zwischen 0 und 1, nach der Gleichung (2.2):

$$\text{Normalisierter Wert} = \frac{X - \text{Min}}{\text{Max} - \text{Min}} \tag{2.2}$$

Wobei X der zu normalisierende Wert, Min der kleinste Wert und Max der größte Wert in den Daten ist. Matlab verwendet dafür die `normalize` Funktion [37, 18]. Diese Methode ermöglicht es, Daten auf einer gemeinsamen Skala zu bringen und verbessert die Klassifikationsleistung von Machine-Learning-Modellen, ist jedoch empfindlich gegenüber Ausreißern, die den Normalisierungsbereich verzerren können [51, 18].

Laut [51], der sich mit der Leistung von Machine-Learning-Modellen für Echtzeitbewegungsvorhersagen mittels EMG befasst, kann eine Kombination aus Normalisierung mit dem Sliding-Window-Verfahren im Bereich der EMG-basierten Prothesensteuerung eingesetzt werden.

2.3.3 Convolutional Neural Networks

Convolutional Neural Networks sind eine spezielle Architektur neuronaler Netzwerke, die ursprünglich für die Bildverarbeitung entwickelt wurden, aber auch erfolgreich für die Analyse von Zeitreihendaten und Sensorsignalen eingesetzt werden können. Während klassische CNNs hauptsächlich auf Faltungsoperationen (Convolutions) basieren, können auch vollständig verbundene Schichten (Dense Layers) als alternative oder ergänzende Architektur verwendet werden. Sie verarbeiten alle Eingabemerkmale durch eine Matrix-Multiplikation, bei der jedes Eingabesignal mit jedem Neuron der Schicht gewichtet verbunden ist. Sie ermöglichen eine Analyse der nichtlinearen Beziehungen zwischen allen Sensorsignalen gleichzeitig. Flatten-Layer aggregieren die zeitlich-räumliche

Information und transformieren die mehrdimensionalen Eingabedaten in einen eindimensionalen Vektor, wobei alle relevanten Merkmale für die nachfolgenden Dense-Schichten verfügbar gemacht werden [18, 39].

Die Verwendung von Rectified Linear Unit (ReLU) als Aktivierungsfunktion in den versteckten Schichten bietet gegenüber anderen Aktivierungsfunkionen den Vorteil einer einfachen Berechnung durch die mathematische Definition [18]:

$$f(x) = \max(0, x) \tag{2.3}$$

Zur Vermeidung von Overfitting können verschiedene Regularisierungsmaßnahmen eingesetzt werden. Early Stopping beendet das Training automatisch, wenn sich die Validierungsverluste über eine definierte Anzahl aufeinanderfolgender Epochen nicht ausreichend verbessern [18]. Zusätzlich ermöglicht die adaptive Lernratenanpassung durch `ReduceLROnPlateau` dem Modell, aus lokalen Minima zu entkommen und feinere Optimierungsschritte zu vollziehen, wenn sich die Verbesserung verlangsamt. `ReduceLROnPlateau` ist ein Keras-Callback für Performance Scheduling, der kontinuierlich den Validierungsverlust überwacht und die Lernrate automatisch reduziert, wenn sich dieser über eine definierte Anzahl aufeinanderfolgender Epochen nicht verbessert, wodurch eine effiziente Konvergenz auch in späteren Trainingsphasen gewährleistet wird [18].

CNN-basierte Systeme in Prothesen- und HCI-Anwendungen müssen hohe Genauigkeitsanforderungen erfüllen. Klassifikationsfehlerquoten unter 5% gelten als praktisch akzeptabel, da bereits geringfügig höhere Fehlerquoten zu unerwünschten Bewegungen und reduzierter Benutzerakzeptanz führen [19].

3 Systementwicklung

In diesem Kapitel wird die Entwicklung des Messsystems beschrieben: beginnend mit den Anforderungen, gefolgt von der verwendeten Hardware sowie der Dokumentation der Firmware und Software. Zudem wird auch die Signalverarbeitung beschrieben und die Umsetzung des Machine-Learning-Methodik zur Gestenerkennung erläutert.

3.1 Anforderungen

Die Anforderungen definieren die funktionalen und qualitativen Eigenschaften des zu entwickelnden optischen Muskelaktivitäts-Erfassungssystems. Sie basieren auf den in Kapitel 2 erarbeiteten theoretischen Grundlagen. Im Rahmen dieser Arbeit dienen folgende Anforderungen als Leitfaden für die Entwicklung des Systems:

Funktionale Anforderungen

R1: Das System soll Muskelaktivität im Unterarm mittels PPG-Sensoren, die rotes Licht (ca. 660 nm) und Infrarotlicht (ca. 880 nm) nach dem Reflexionsprinzip der OMG nutzen erfassen [41, 20, 39] (siehe Abschnitt 2.1.5).

R2: Das System soll mindestens acht Sensoren in fester Platzierung am Unterarm verwenden, um eine ausreichende Abdeckung der oberflächlichen Flexor- und Extensor-Muskelgruppen zu erreichen [43, 36] (siehe Abschnitte 2.1.4, 2.3.3).

R3: Das System muss eine CNN-basierte Erkennung für verschiedene Handgelenksgesten ermöglichen, einschließlich grundlegender Bewegungsrichtungen wie Flexion, Extension und Abduktion, entsprechend den biomechanischen Eigenschaften der Unterarmmuskulatur [43, 41, 20] (siehe Abschnitte 2.3.3, 2.1.4).

R4: Das System muss eine Live-Gestenerkennung mit einer maximalen Latenz von 300 ms nach Bewegungsausführung ermöglichen, da längere Verzögerungen die Hand-System-Koordination beeinträchtigen und die Akzeptanz von HCI-Anwendungen reduzieren [15, 13, 14, 17, 10, 22] (siehe Abschnitt 2.1.4).

R5: Das System soll systematische Messabläufe und eine grafische Oberfläche umsetzen, welche die Probanden zur korrekten Ausführung der vorgegebenen Gesten anleitet, um eine konsistente Datenerfassung sicherzustellen. Die grafische Oberfläche muss mindestens eine Wiederholungsrate von 60 Hz aufweisen, damit der Anwender Veränderungen in der Darstellung ohne Verzögerung wahrnehmen kann [28, 29].

Qualitätsanforderungen

R6: Das System muss die vom Medical Sensor Lab bereitgestellte Hardware verwenden, bestehend aus MAX30102 PPG-Sensoren und Raspberry Pi Pico [38, 45]. Die Konfigurationsparameter der Sensoren sind zu untersuchen und für die spezifischen Anforderungen dieser Arbeit zu optimieren [38].

R7: Das System soll die gemessenen Daten per USB-Schnittstelle mit 115200 Baud über UART-Protokoll übertragen, wobei ein medizinischer USB-Isolator (EN 60601-1 zertifiziert) die galvanische Trennung gewährleisten muss [46, 25] (siehe Abschnitte 2.2 und 2.1.1).

R8: Die Temperaturänderungen des Systems müssen geprüft werden. Die Sensoroberflächentemperatur darf 40°C nicht überschreiten, um die Sicherheit der Probanden während der Messungen zu gewährleisten [42] (siehe Abschnitt 2.1.1).

R9: Das System soll Glättung und Normalisierung auf die erfassten Rohdaten anwenden, um bessere Klassifikationsergebnisse bei der CNN-basierten Gestenerkennung zu erzielen. [3, 16, 42, 18] (siehe Abschnitte 2.3.1, 2.3.2).

R10: Die CNN-basierte Gestenerkennung benötigt eine Klassifikationsgenauigkeit von mindestens 95 %, da Klassifikationsfehlerquoten unter 5 % für praktische HCI-Anwendungen erforderlich sind [19] (siehe Abschnitt 2.3.3).

R11: Das CNN-Klassifikationsmodell soll Sensordaten unter einem Zeitfenster von 200 ms verarbeiten, welches ausreichend ist, um die relevanten Frequenzkomponenten der Handgelenkbewegungen bis 5 Hz für die Gestenerkennung zu analysieren [21, 35, 55] (siehe Abschnitt 2.1.4).

R12: Wenn Veränderungen der Muskulatur auftreten, sollen die Sensoren so zeitnah wie möglich diese erfassen. Deswegen soll die Reaktionszeit der Sensoren keine Verzögerungen über 100 ms bei der Erfassung aufweisen, um die Korrelation der Muskelbewegungen bei der Gestenerkennung zu gewährleisten [21, 35, 55] (siehe Abschnitt 2.1.4).

R13: Das System soll eine Messreihe aller Sensordaten innerhalb eines 10 ms-Zeitfensters bereitstellen, sodass bei einer 100 Hz Abtasfrequenz die notwendigen beschriebenen zeitlichen Anforderungen gewährleistet werden können (siehe Abschnitte 2.1.4, 2.3.3).

3.2 Konzept

Das entwickelte System besteht aus zwei Hauptbestandteilen: einem Armbandsystem und einer Software-Komponente am Rechner für Datenempfang und -auswertung.

Im Rahmen dieser Arbeit wird das Armbandsystem für den rechten Unterarm entwickelt. Dieses enthält hauptsächlich einen Raspberry Pi Pico zur Steuerung von 8 PPG Sensoren des Typs MAX30102. Über eine USB-Schnittstelle werden die Daten an den Rechner übertragen, wo die Analyse mittels Software erfolgt. Die 8 Sensoren müssen an bestimmten Stellen am Unterarm angebracht werden. Ein Armband, das als Träger für diese Sensoren dient, stellt sicher, dass alle Sensoren fest platziert werden können. Da verschiedene Probanden unterschiedlich groß sind, soll das Armband elastisch und gewissermaßen flexibel sein. Deswegen werden die Sensoren in gleichmäßigen Abständen zueinander an einem elastischen Stoff befestigt, wie in Abb. 3.1 dargestellt. Die Befestigung soll den gesamten Umfang des Unterarmes abdecken. Der Raspberry Pi Pico steuert die Sensoren nicht nur an, sondern versorgt sie gleichzeitig mit Strom. Die konsistente Platzierung der Sensoren ist relevant für die Gestenerkennung. Deswegen sind die Sensoren nummeriert und bei den Messungen immer an der gleichen Position angebracht. In Abbildung 3.2 ist die festlegte Platzierung der Sensoren zu sehen.

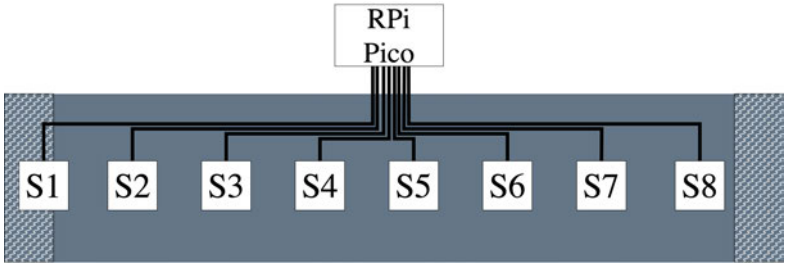


Abbildung 3.1: Skizze des Stoffarmbandes. Alle 8 Sensoren sind in gleichmäßigen Abständen zueinander befestigt und mit dem Raspberry Pi Pico (RPi Pico) verbunden. Die Sensoren sind von S1 bis S8 gekennzeichnet. An beiden Seiten des Armbandes befinden sich ein Klettverschlüsse, um das Armband zu schließen.

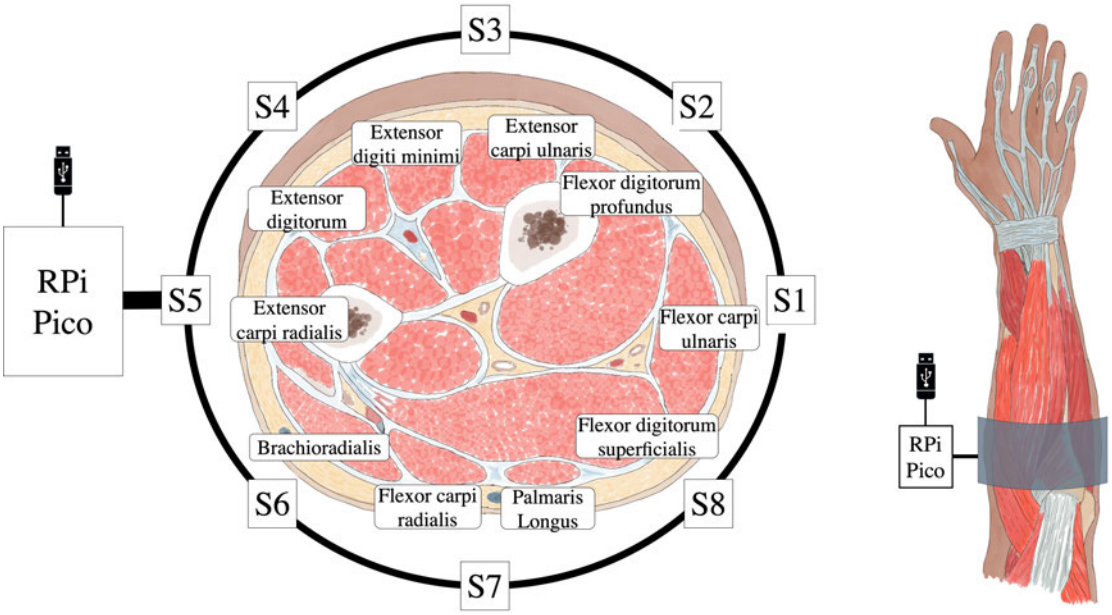


Abbildung 3.2: Platzierung der Sensoren am Unterarm. Links auf dem Bild ist der Querschnitt des rechten Unterarms zu sehen. Die Vorderseite des Unterames zeigt nach unten. Um den Unterarm herum sind die jeweiligen Sensoren des Armbandes erkennbar. Sie sind nummeriert und müssen nach diesem Muster immer angebracht werden. Rechts ist die Vorderseite des rechten Arms dargestellt, um die Position des Arms deutlicher zu machen.

3.3 Verwendete Hardware

Für die Realisierung des Systems stehen die folgende Hardware-Komponenten zur Verfügung. Als zentrale Steuerungseinheit dient der Raspberry Pi Pico (RP2040) aufgrund seiner kompakten Bauweise, ausreichender Rechenkapazität und der Verfügbarkeit von 28 GPIO-Pins, die eine zuverlässige Kommunikation mittels I2C mit den Sensoren ermöglichen. Der MAX30102 dient als photoplethysmographischer Sensor, da seine dualen LEDs mit spezifischen Wellenlängen (660 nm rot und 880 nm infrarot) eine optimale Eindringtiefe im Hautgewebe bieten und somit die Erfassung von Blutvolumenschwankungen ermöglichen (siehe Abb. 3.3b). Die I2C-Kommunikation erreicht ein energieeffiziente und zuverlässige Datenübertragung bei einer Taktrate von bis zu 400 kHz. In den folgenden Abschnitten werden die technischen Spezifikationen und Funktionsweisen der einzelnen Komponenten detailliert erläutert.

(a) Raspberry Pi Pico (b) MAX30102 PPG Sensor

Abbildung 3.3: Obere Ansicht der Hardware-Komponenten für das Optomyographie-Messsystem. (a) Raspberry Pi Pico Board mit gekennzeichneten GPIO-Pins am Rand, dem 3V3(OUT)-Pin für die Stromversorgung der Sensoren auf der oberen Seite und insgesamt 8 Ground-Pins auf beiden Seiten. (b) MAX30102 PPG Sensor mit Stromversorgung, Ground, I2C Takt- und Datenleitung sowie Interrupt-Pins auf der linken Seite. In der Mitte befinden sich die Leucht- und Fotodioden [45].

3.3.1 Raspberry Pi Pico

Der Raspberry Pi Pico, auch als RP2040 oder RPi Pico bekannt, ist ein Mikrocontroller mit 2MByte Flash Speicher und verfügt zudem über ausreichende Dokumentation und Software Beispiele, die seine Verwendung zu erleichtern. Dem Datenblatt zufolge stellt die Hardware 28 multifunktionale 3,3 V General Purpose I/O zur Verfügung, welche sich für die Software I2C verwenden lassen. Die Leiterplatte ist 51 mm x 21 mm groß und hat

eine Dicke von 1 mm (siehe Abb. 3.3a). Der Raspberry Pi Pico lässt sich mittels eines Mikro-USB-Anschlusses mit Strom versorgen [45].

3.3.2 MAX30102 Sensor

Der MAX30102 nutzt ein duales LED-Photodioden-System, welches sich aus einer 1,8 V-Versorgung für die Sensorelektronik sowie einer 5,0 V Versorgung für die LED-Ansteuerung zusammensetzt. Dem Datenblatt zufolge [38] besitzt der MAX30102 einen niedrigen Stromverbrauch (etwa 600 µA). Der Sensor besteht aus einer Infrarot-LED mit einer Peak-Wellenlänge von 880 nm mit einer Abweichung von ±20 nm und einer roten LED mit 660 nm mit einer Abweichung von ±10 nm. Diese Spezifikationen gelten unter Standardbedingungen von 20 mA Betriebsstrom bei +25°C Umgebungstemperatur. Der integrierte Photodetektor weist einen spektralen Empfindlichkeitsbereich von 600 nm bis 900 nm auf [38].

Das Modul nutzt eine I2C-kompatible Schnittstelle mit Taktraten bis 400 kHz zur Kommunikation. Die Standard-I2C-Adressen sind 0xAE (für Schreiboperationen) und 0xAF (für Leseoperationen) und das Kommunikationsprotokoll folgt dem I2C-Standard mit definierten Start-, Stop- und Acknowledge-Bedingungen. Die Datenübertragung erfolgt in 8-Bit-Worten, wobei jedes Wort von einem Acknowledge-Taktimpuls bestätigt wird. Für Leseoperationen vom MAX30102 sendet der Controller die entsprechende Target-Adresse, gefolgt von einer Serie von neun SCL-Impulsen. Diese standardisierte Kommunikationsarchitektur gewährleistet eine zuverlässige Datenübertragung zwischen dem Sensor und dem übergeordneten System [38].

Aus dem Datenblatt [38] kann entnommen werden, dass der MAX30102 unter verschiedenen Einstellungen arbeiten kann. Die Auswahl des Multi-LED-Mode schaltet die Kompensationsfähigkeit für Umgebungslicht aus und verwendet beide LEDs abwechselnd. Der eingebaute Temperatursensor kann zur Kompensation der thermischen Charakteristik der LEDs bei Blutsauerstoffmessungen dienen.

3.3.3 Konfigurationsparameter MAX30102

Die Samplingrate des MAX30102 ist zwischen 50 Hz und 3200 Hz einstellbar. Der integrierte FIFO-Speicher ermöglicht eine Mittelung über 1, 2, 4, 8, 16 oder 32 Samples zur

Rauschreduktion.Die Daten werden in einem FIFO-Puffer mit einer Tiefe von 32 Einträgen zwischengespeichert, wobei jeder Eintrag 6 Bytes (3 für IR, 3 für Rot) umfasst [38]. Die LED-Helligkeit lässt sich in vier Stufen einstellen (0,4 mA, 6,4 mA, 25,4 mA und 50,0 mA).

Der Analog-Digital-Wandler (ADC) kann auf verschiedene Messbereiche eingestellt werden (2048 nA, 4096 nA, 8192 nA, 16384 nA), wobei ein kleinerer Messbereich eine höhere Auflösung für schwache Signale bietet, jedoch bei zu starken Signalen zur Sättigung führen kann. Das bedeutet, dass bei zu intensivem Licht der Sensor seinen maximalen Messwert erreicht und keine weiteren Helligkeitsunterschiede mehr unterscheiden kann. Der Sensor arbeitet mit einem alternierenden Messprinzip, bei dem zunächst die rote LED für die eingestellte Integrationszeit leuchtet, dann folgt nach einer definierten Pause die IR-LED für dieselbe Dauer. Anschließend erfolgt eine weitere Pause, bevor der nächste Messzyklus beginnt. Die Gesamtzykluszeit wird durch die gewählte Abtastfrequenz bestimmt [38].

Die Kombination aus gewähltem Messbereich und Integrationszeit wirkt sich direkt auf die effektive Auflösung in Bit aus, die mit zunehmender Integrationszeit steigt. Eine höhere Auflösung bedeutet feinere Unterscheidung von Signalstärken. Der Auflösungsschritt in pA gibt den tatsächlichen physikalischen Wert an, der einer einzelnen digitalen Stufe entspricht. Beim MAX30102-Sensor können ADC-Messbereich (2048, 4096, 8192, 16384 nA) und Integrationszeit (69, 118, 215, 411 µs) unabhängig voneinander eingestellt werden. Die Integrationszeit bestimmt dabei die Bitauflösung von 15 bis 18 Bit. Bei der höchsten Integrationszeit (411 µs) ist die maximale Abtastfrequenz auf 400 Hz begrenzt [38].

Tabelle 3.1: Auflösungswerte des MAX30102-Sensors in Pikoampere (pA) in Abhängigkeit vom ADC-Bereich und der Integrationszeit. Die Integrationszeiten entsprechen folgenden Bitauflösungen: 69 µs (15 Bit), 118 µs (16 Bit), 215 µs (17 Bit), 411 µs (18 Bit). Werte aus dem Datenblatt [38]

ADC-Bereich	**69 µs**	**118 µs**	**215 µs**	**411 µs**
2048 nA	62,5 pA	31,25 pA	15,63 pA	7,81 pA
4096 nA	125,0 pA	62,5 pA	31,25 pA	15,63 pA
8192 nA	250,0 pA	125,0 pA	62,5 pA	31,25 pA
16384 nA	500,0 pA	250,0 pA	125,0 pA	62,5 pA

3.4 Konstruktion des Armbandes

Die Herstellung des Armbandes ist eigenständig über mehrere Iterationen erfolgt. Als Grundgerüst dient ein elastisches schwarzes Stück Stoff. Die 1. Iteration des Armbandes wird nach dem bereits erklärten Muster im Abschnitt 3.2 erstellt (siehe technische Zeichnung in Abb. 3.4). Die Sensoren werden auf der Innenseite des Armbandes platziert in gleichmäßigen Abständen von 2,3 cm. Ein schützendes Stück Stoff ist angebracht, sodass die Haut vor Kontakt mit den Pins verhindert und der Benutzer des Armbandes keine Kratzer erleidet. An den Seiten sind Klettverschlüsse angenäht für den Verschluss. Diese Iteration bietet erste Erkenntnisse für die Umsetzung. Das Stück Stoff, das die Haut vor Kontakt mit den Pins schützen soll, verringert die Elastizität des Armbandes. Sowohl die zu enge Passform bei einem Unterarmumfang von 30 cm als auch die durch den Klettverschluss entstehenden Lücke zwischen den Sensoren beeinträchtigen die Funktionalität bei der 1. Iteration.

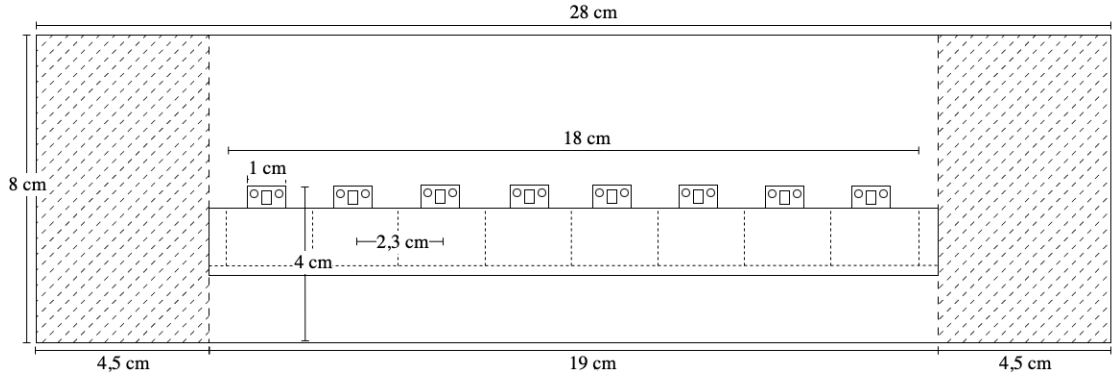
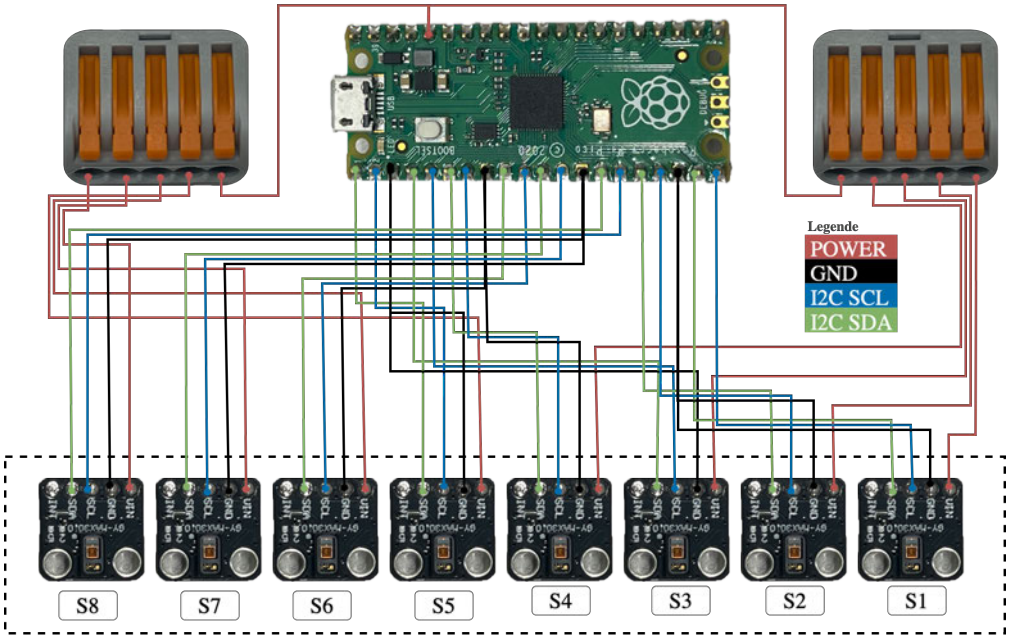


Abbildung 3.4: Technische Zeichnung der 1. Iteration des Armbandes. Das Armband ist von der Innenseite sichtbar. Die Klettverschlüsse befinden sich an beiden Enden der Innenseite. Die Sensoren sind an der Innenseite angebracht.

Die 2. Iteration des Armbandes löst diese Probleme und besitzt einen größeren Umfang. Bei dem neuen Verschluss befindet sich die Hakenseite außen und die Flauschseite innen, was den ersten Sensor näher am Rand platzieren lässt. Das Armband benötigt kein schützendes Stück Stoff zwischen Pins und Haut, weil die Sensoren nicht mehr auf der Innenseite platziert sind. Von der Außenseite können sie durch eine kleine Öffnung im Stoff messen. Das Armband ist flexibler aber es entstehen Probleme mit Brüchen an den Lötstellen. Deswegen verfügt die dritte und letzte Iteration über vieradrige Kabel, die gebogen über das Armband verstrickt sind, sodass keine Bruchstellen entstehen können. Sensoren werden auf der Rückseite verdeckt und geschützt durch Stoffstreifen. Die

Elastizität des Armbands ist dadurch nicht eingeschränkt, weil diese Streifen vertikal angebracht sind, wie in Abb. 3.6a dargestellt.

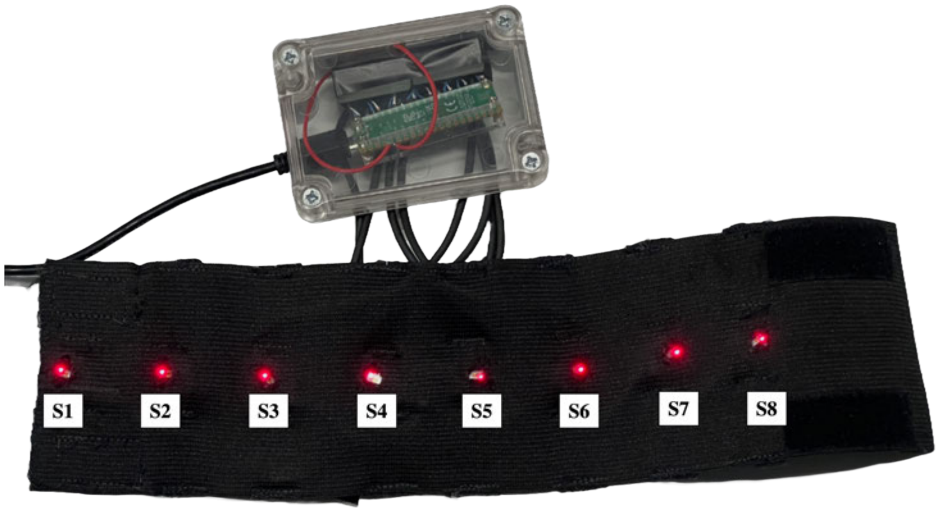
Zwei Wago-Klemmen [56] werden verwendet, um den Stromversorgungspin besser zwischen den Sensoren zu teilen. Diese liegen gemeinsam mit dem Raspberry Pi Pico in einer 6 cm x 8,5 cm x 4 cm großen Plastikbox um sie zu schützen (siehe Abb. 3.6b). Die Sensoren sind mit 3,5 cm Abstand voneinander platziert und decken eine Länge von 25,5 cm ab. Alle Maße des Armbandes sind auf der technischen Zeichnung 3.6c zu sehen. Die Schaltung ist in Abb. 3.5 dargestellt. Die GPIO-Pin Belegung wird in Tabelle in Abb. 3.5 dargestellt. Alle Sensoren teilen sich den 3,3 V Pin und jeweils zwei Sensoren teilen sich die Ground Pins auf der linken Seite des Raspberry Pi Pico (siehe Abb. 3.3a).



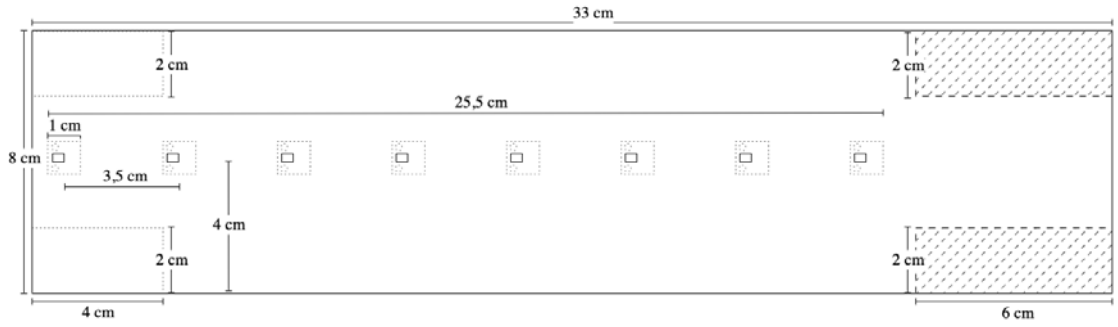
Sensor	8	7	6	5	4	3	2	1
SDA GPIO	10	8	6	0	4	2	12	14
SCL GPIO	11	9	7	1	5	3	13	15

Abbildung 3.5: Schaltung des Armbandsystems. Die Stromverteilung über den 3,3 V Pin erfolgt über zwei Wago-Klemmen, die oben links und rechts zu sehen sind. Die Leitungen sind in der Legende farblich gekennzeichnet. Die Tabelle unterhalb zeigt die GPIO-Pin Zuordnung der 8 PPG Sensoren in der Software-Implementation.

(a) Außenseite



(b) Innenseite



(c) Technische Zeichnung

Abbildung 3.6: Letzte Iteration des Armbandes. Die Abbildungen zeigen das Armband aus verschiedenen Perspektiven ((a) Außenseite, (b) Innenseite) sowie die technische Zeichnung (c) der Innenseite. Der Raspberry Pi Pico sowie zwei Klemmen befinden sich in einem Kasten der Größe 6 cm×8,5 cm×4 cm.

3.5 Systemarchitektur

Dieser Abschnitt beschreibt die Umsetzung der Firmware und Software. Die Firmware wird mittels Micropython auf dem Raspberry Pi Pico realisiert. Sie übernimmt die Steuerung der MAX30102-Sensoren, die zeitgesteuerte Datenerfassung und die Kommunikation über die serielle Schnittstelle. Die Software stellt die übergeordnete Anwendungsebene dar und kümmert sich um die Datenverarbeitung und -speicherung sowie grafische Darstellung der Messwerte. Beide Komponenten arbeiten über eine serielle Verbindung zusammen und ermöglichen die kontinuierliche Erfassung und Darstellung von Daten.

3.6 Firmware

Die Architektur der entwickelten Firmware wird im Klassendiagramm 3.7 gezeigt. Sie wurde in einem modularen Ansatz strukturiert, um dem Prinzip der *Separation of Concerns* zu wahren und schafft eine hierarchische Struktur, die die Hardware Abstraction Layer (HAL) von der Anwendungslogik trennt.

Die Klasse `DataCollector` implementiert die Logik zur Erfassung der Sensordaten. Das Verfahren wird von einer For-Schleife umfasst, die die Anzahl der gewünschten Durchläufe und somit die Gesamtmenge der zu erfassenden Datenpunkte vorgibt. Innerhalb dieser Schleife wird ein Timer gestartet, der in vorgegebenen Abständen zyklisch eine Callback-Funktion aufruft. Die `timer_callback()`-Funktion steuert einen Flag, der freigibt wann die nächste Datenmessung durchgeführt werden darf. Die While-Schleife, die auf den Flag wartet, kann mit minimaler Verzögerung unterbrochen werden. Sobald die Schleife freigegeben ist, werden alle 8 Sensoren nacheinander abgefragt. Dies erspart eine komplizierte Umsetzung von Synchronisationsmechanismen, wie Mutex und Semaphoren, die sich schwer auf dem Raspberry Pi Pico unter Micropython umsetzen lassen. Entstehen Fehler beim Abfragen eines Sensors, so wird -1 als Rückgabewert zurückgegeben. Wie bereits in den Grundlagen 3.3.2 erwähnt sind Software-Timer auf dem Raspberry Pi Pico im Millisekunden-Bereich grundsätzlich zuverlässig. Durch eine Übertaktung des Prozessors auf 250 MHz wird die Reaktionszeit zwischen Timer-Auslösung und Flag-Setzung minimiert.

Die Klasse `SensorConfig` übernimmt die Konfiguration der MAX30102-Sensoren basierend auf den eingegebenen Parametern des Anwenders. Die Anzahl der zu erfassenden

Datensätze wird durch `data_size` übermittelt. Für die zeitliche Steuerung der Datenerfassung dient der Parameter `frequence_in_ms`, der die Abtastrate in Millisekunden definiert. Die Sensor-spezifischen Einstellungen werden durch mehrere Parameter gesteuert. Die Sensor-Abtastrate, oder auch Samplingrate, wird durch `sensor_sample_rate` bestimmt. `sensor_fifo_average` setzt den Parameter für die Durchschnittsbildung im FIFO-Puffer fest. Die LED-Konfiguration erfolgt über `sensor_led_amplitude` für die Helligkeitseinstellung und `sensor_pulse_width` für die Pulsdauer. Der Wertebereich des Analog-Digital-Wandlers wird durch `sensor_adc_range` festgelegt. Optional kann über `sensor_led_toggle` ein Bitmuster zur LED-Steuerung übergeben werden, standardmäßig sind alle LEDs aktiviert `('11111111')`.

Die Klasse `MAX30102` repräsentiert die HAL und somit die Schnittstelle zum Sensor. Sie kapselt die gerätespezifischen Funktionen der Sensoren. Das Grundgerüst der Klasse wurde aus dem Github-Repository [12] entnommen, überarbeitet und optimiert. Wesentliche Modifikationen für diese Arbeit umfassen die vollständige Entfernung des zirkulären Puffers und Unterstützung des MAX30105, sowie eine grundlegende Neuimplementierung der Sensorabfrage. Im Gegensatz zur ursprünglichen Implementierung, die für den SpO2-Modus des Sensors konzipiert war und mehrere Zwischenschritte bei der Datenverarbeitung verwendete, wurde hier ein direkter Zugriff auf die Sensordaten implementiert.

Die Klasse `HardwareManager` ist verantwortlich für die Abstraktion und Verwaltung der spezifischen Konfigurationen der Iteration des Armbands. Sie enthält die richtige Anordnung der Pins, die sich je nach Iteration des Armbandes unterscheiden können, sowie um spezifische Fehlerkorrekturen einzelner Sensoren.

3.6.1 Prozesse der Firmware

Nach der Initialisierung der Klassen wird eine Abfrage gestartet, die zur Identifikation des Armbandes dient, bei anhaltenden ungültigen Eingaben wird eine Fehlerausgabe generiert und das Programm beendet. Bei erfolgreicher Ermittlung werden die gerätespezifischen Hardware-Einstellungen geladen. Eine Tabelle 3.2 mit den verschiedenen Konfigurationsoptionen der Sensoren wird in der Konsole dargestellt, diese sind in Zahlen zwischen 1 und 4 vereinfacht. Die Struktur der Parametereingabe wird in dem Abschnitt 3.7 vorgestellt. Die Abfragen sollen Ähnlichkeit zum Drei-Wege-Handshake aufweisen und finden über die REPL statt.

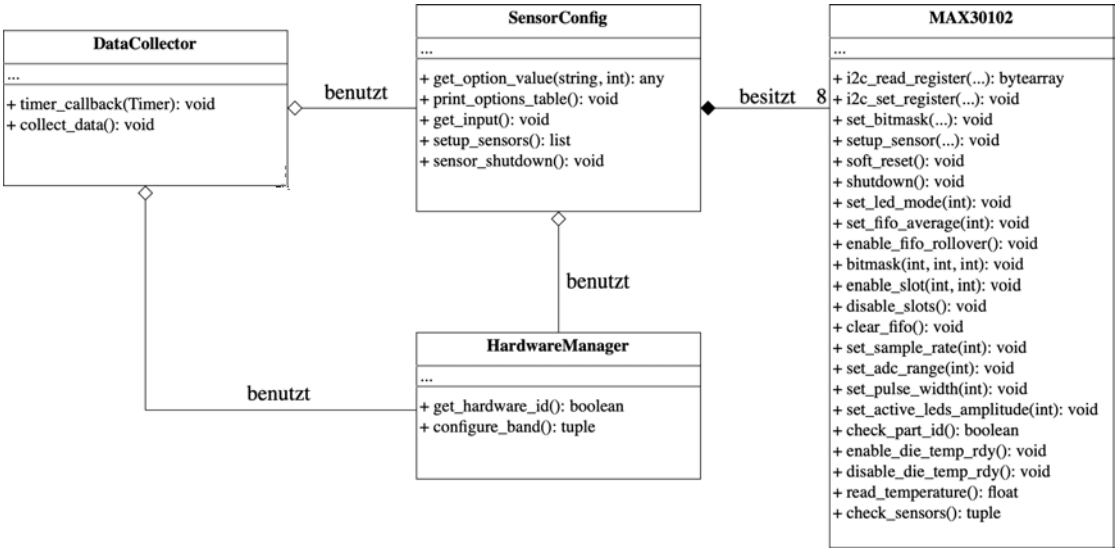


Abbildung 3.7: Klassendiagramm der Firmware mit den wichtigsten Klassen und deren Beziehungen. Die Sensor-Konfiguration erstellt 8 MAX30102-Sensor Objekte, auf die der Datensammler zur Messwerterfassung zugreift. Variablen der Klassen sind nicht dargestellt.

Tabelle 3.2: Konfigurationsoptionen der Sensoren mit 4 verschiedenen Optionen für die entsprechenden Parameterwerte.

Option	**1**	**2**	**3**	**4**
ADC-Bereich [-]	2048	4096	8192	16384
Sensor Samplingrate [Hz]	100	400	800	1600
ADC Integrationszeit [μs]	69	118	215	411
LED Helligkeit [mA]	0,4	6,4	25,4	50
FIFO-Mittelwerte [-]	1	2	4	8

Nach erfolgreicher Konfiguration startet der Timer, der die zeitliche Steuerung der Datenerfassung regelt. Wie bereits beschrieben implementiert die `DataCollector`-Klasse eine Datenerfassungsschleife, die auf die Freigabe dieses Timers wartet. Sollte ein Sensor zum gegebenen Zeitpunkt einen Fehler aufweisen und keine Daten zur Verfügung stellen, wird -1 zurückgegeben, sodass die Software den Fehler behandeln kann. Der Ausgabestring einer Messreihe ist in Abb. 3.8 und das Kontrollflussdiagramm der Firmware in Abb. 3.9 dargestellt. Ist die eingegebene Datenmenge erreicht, endet die For-Schleife und auf der Konsole wird die Gesamtdauer der Messung ausgegeben. Um das Ende der Übertragung deutlich zu signalisieren, wird ein End of Transmission (EOT)-Zeichen übertragen.

Abbildung 3.8: Struktur des Datenausgabestrings. Die ersten 8 Werte enthalten die roten Sensorwerte (Sensoren 1 bis 8), gefolgt von 8 Infrarot-Sensorwerten derselben Sensoren. Am Ende steht ein unformatierter Zeitstempel und ein Datenzähler. Alle Werte sind durch Kommas getrennt und der gesamte String ist von eckigen Klammern umschlossen.

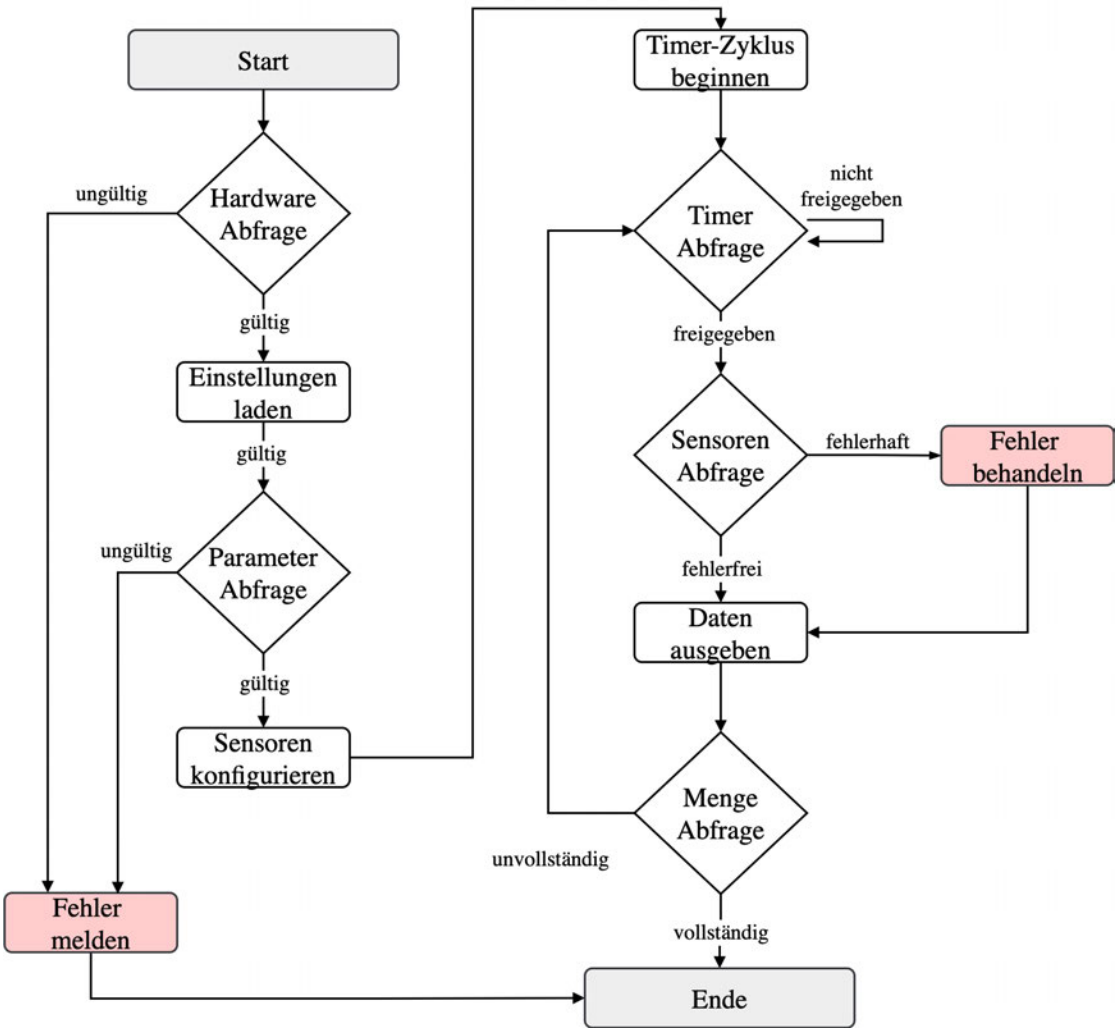


Abbildung 3.9: Kontrollflussdiagramm der Firmware. Auf der linken Seite sind die Prozesse zur Initialisierung und Konfiguration der Hardware. Ungültige Eingaben führen zum Programmabbruch. Auf der rechten Seite ist die Datenerfassungsschleife sichtbar. Die Schleife wird fortgesetzt, bis die erforderliche Datenmenge erreicht ist.

3.7 Software

Die Software ist nach einem modifizierten Model-View-Controller (MVC)-Muster aufgebaut. Die `SensorData`-Klasse stellt als Model-Komponente den Zugriffspunkt der Sensordaten bereit, indem sie alle Sensorwerte sowie Statusflags und Klassifikationslabels speichert, bereitstellt und verwaltet. Die Klasse `SensorPlotter` agiert als View-Komponente und visualisiert diese Daten während der Aufnahme durch verschiedene Darstellungsformen wie Liniendiagramme, Heatmaps und kreisförmige Visualisierungen, die eine intuitive Interpretation der Sensorwerte ermöglichen sollen. Die `Receiver`–Klasse übernimmt durch Erfassung und Verarbeitung der Sensordaten die Controller-Funktion. Sie kommuniziert mit der Hardware über eine Instanz der `Connection`–Klasse, validiert eingehende Daten, korrigiert fehlerhafte Werte und aktualisiert das `SensorData`-Modell. Gleichzeitig organisiert sie die systematische Speicherung der Daten in einer hierarchischen Ordnerstruktur nach Messtyp und Zeitstempel.

Die modulare Architektur unterstützt sowohl die Live-Visualisierung der Sensordaten als auch die Gestenerkennung durch parallele Datenverarbeitung mittels Multithreading.

Die `SystemManager`–Klasse übernimmt die zentrale Steuerung des Systems und startet Skripte der Anwendung basierend auf vordefinierten Parametern aus einer JSON-Konfigurationsdatei. Die Klasse erzeugt und koordiniert die restlichen Komponenten der Software und agiert somit als übergeordneter Prozess, der den gesamten Systemablauf orchestriert.

Eine Abweichung vom klassischen MVC-Muster liegt in der direkten Kommunikation zwischen Model (`SensorData`) und View (`SensorPlotter`). Im traditionellen MVC-Ansatz wird typischerweise das Observer-Pattern verwendet. In dem entwickelten System hingegen greift der `SensorPlotter` direkt auf `SensorData` zu, was den Overhead durch den zusätzlichen Vermittlungsschritt vermeidet. Im Klassendiagramm in Abb. 3.10 werden diese Zusammenhänge dargestellt.

Die Klassen sind nach dem Prinzip der Kapselung konzipiert, sodass die Wartung vereinfacht sowie höhere Testbarkeit und Erweiterbarkeit erreicht wird.

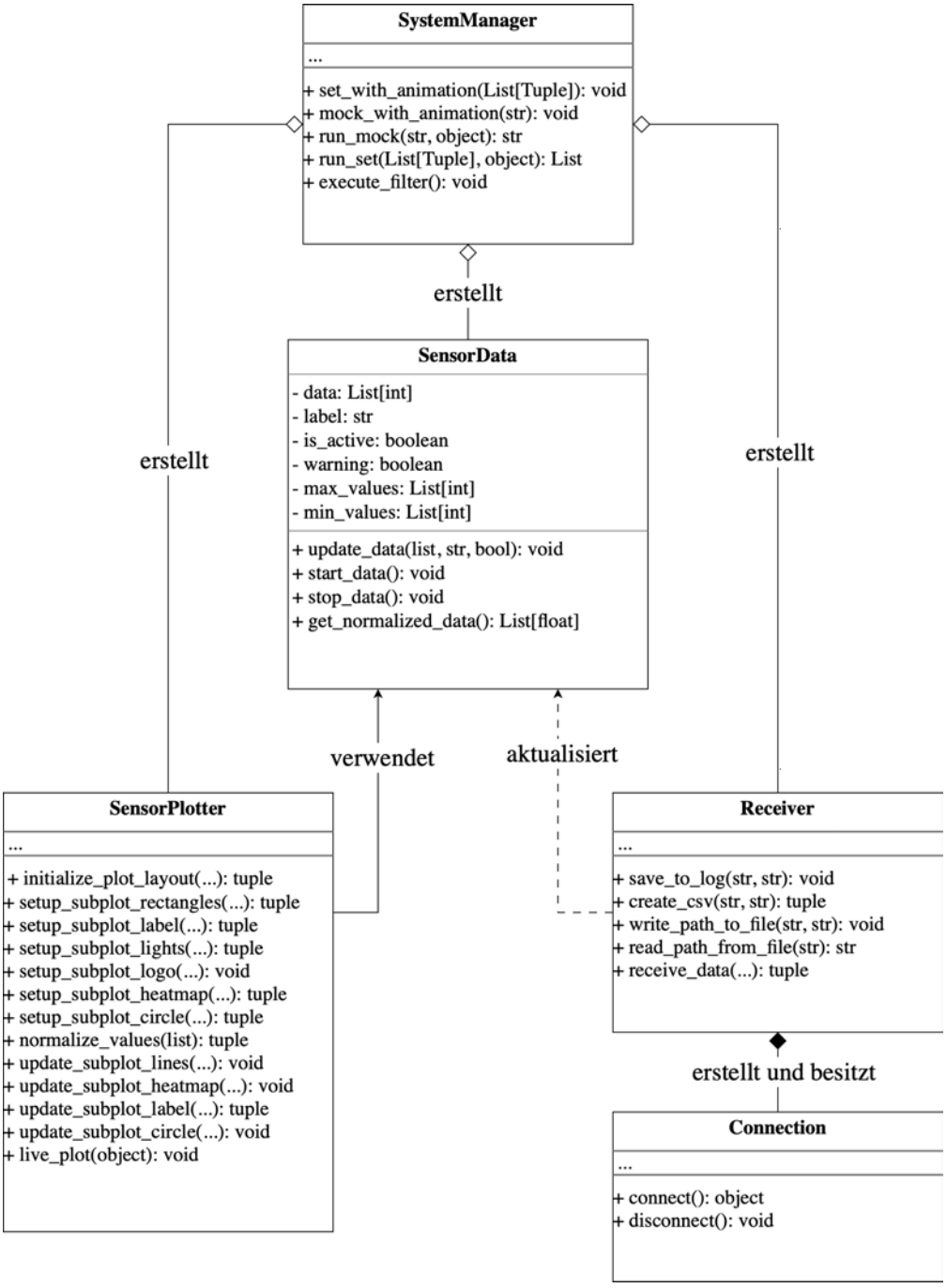


Abbildung 3.10: Klassendiagramm der Software. Die Abbildung zeigt die Klassen und ihre Beziehungen sowie die wichtigsten Methoden. Auschliesslich bei der SensorData-Klasse sind die relevanten Variablen dargestellt.

3.7.1 Prozesse der Software

Zu Beginn initialisiert die `SystemManager`-Klasse alle Komponenten und startet zwei parallele Threads für die Datenerfassung und die Visualisierung. Die `SystemManager`-Klasse ermöglicht über die Methode `set_with_animation()` das Starten von Mess-Skripte, die in eine JSON-Konfigurationsdatei definiert werden. Die `SystemManager`-Klasse interpretiert diese Parameter und initiiert für jede Einstellung einen separaten Messvorgang. Sie erzeugt und konfiguriert die notwendigen Objekte (`SensorData, Receiver, SensorPlotter`).

Im Datenerfassungsthread verbindet sich die `Receiver`-Klasse über die `Connection`-Klasse mit der Hardware und sendet die Konfigurationsparameter. Ihr Lebenszyklus wird von der `Receiver`-Klasse gesteuert und übernimmt das automatische Auffinden und Verbinden mit dem seriellen Port mittels der Nutzung der Python-Bibliothek `serial`.

Konfigurationsparameter der Sensoren werden als leerzeichengetrennter String zusammengefasst. Die Struktur des Strings ist in der Tabelle 3.3 zu sehen.

Tabelle 3.3: Struktur des Konfigurationsstrings. Die Voreinstellungen (1-4) repräsentieren jeweils vordefinierte Konfigurationswerte (siehe Abschnitt 3.6). Alle Werte sind als Ganzzahlen anzugeben.

Position	**Parametername**	**Parameterbeschreibung**	**Wertebereich**
0	data_size	Menge der Datenpunkte pro Sensor	$1 \leq x$
1	frequency_in_ms	Zeitintervall zwischen Messungen in Millisekunden	$10 \leq x$
2	sensor_sample_rate	Samplingrate der Sensoren	$1 \leq x \leq 4$
3	sensor_fifo_average	Durchschnittsberechnung im FIFO	$1 \leq x \leq 4$
4	sensor_led_amplitude	Intensität und Helligkeit der LED	$1 \leq x \leq 4$
5	sensor_adc_range	Messbereich des Analog-Digital-Wandlers	$1 \leq x \leq 4$
6	sensor_pulse_width	Dauer der Integrationszeit (*pulse width*)	$1 \leq x \leq 4$
7	sensor_led_toggle	Aktivierungsmuster für die einzelnen Sensoren	$\{0,1\}^8$
8	\r	Zeilenumbruch zur Termination des Befehls	-

Die Methode `receive_data( )` in der `Receiver`-Klasse ist für das Empfangen der Datenstrings verantwortlich. Ankommende Daten werden in einer CSV-Datei und Log-Datei gesichert. Zur strukturierten Sicherung gehören die Parameter `data_type` und `data_section`. Sie kennzeichnen die Art der Datenerfassung für die Ordnerstruktur. Verzeichnisse werden nach einem hierarchischen Prinzip erstellt, wobei die Methode `create_csv()` automatisch Ordner im Format `Messungen/Typ/Datum/` anlegt. Die Methode `read_path_from_file()` dient zum Zugriff auf diesen Pfad.

Die Daten werden kontinuierlich empfangen und auf Fehler überprüft. Prüfungen umfassen die Überwachung der zeitlichen Abstände zwischen Messungen, den Abgleich der empfangenen Datenmenge mit der Anzahl gemessener Daten und die Plausibilitätsprüfung der Sensorwertebereiche. Bei fehlerhaften oder nicht verfügbaren Sensorwerten werden die letzten gültigen Werte beibehalten. Sollten diese Fehler mehrfach auftreten, liegt es beim Anwender zu entscheiden, ob die Messung gültig ist. Entsteht ein Fehler beim Prüfen des Datenzählers, so fehlt eine Messreihe und das Programm wird beendet. Beim Erkennen eines Fehlers wird dieser im Logfile dokumentiert und ein Warnflag wird gesetzt. Der Datenfluss der Die `Receiver`–Klasse ist im Kontrollflussdiagramm in Abb. 3.11 dargestellt.

Optional können die Daten beim Empfangen mit Labels versehen werden. Die `Receiver`–Klasse implementiert ein Labelling-System, welches die Klassifikation ermöglichen wird. Durch die Parameter `first_label`, `second_label` und `label_interval` wird ein alternierendes Labeling-Schema definiert. Die `Receiver`–Klasse vergibt die Labels basierend auf dem definierten Intervall zwischen verschiedenen Label-Kategorien mit einer fortlaufenden Nummerierung (z.B. A1, B1, A2, B2...).

Die Daten werden dann über die `update_data()`-Methode an das `SensorData`-Objekt übergeben. Wird die Anwendung für das Erkennen von Gesten verwendet, so kann die `update_data()`-Methode verwendet werden, um die Glättung, Normalisierung und Klassifikation durchzuführen. Bei Aktualisierung der Messreihen werden diese Schritte durchgeführt.

Das Empfangen eines EOT-Zeichens führt zum Beenden der Funktion. Nach Beendigung der Messung werden die Ressourcen freigegeben und die Verbindung geschlossen.

Im Visualisierungsthread greift die `SensorPlotter`-Klasse kontinuierlich auf die Daten im `SensorData`-Objekt zu und liest dadurch die aktuellen Messwerte. Die Klasse verwendet `matplotlib` und `numpy`, um mit der Methode `live_plot()` die grafische Oberfläche umzusetzen. Die Daten müssen normalisiert werden, um eine konsistente Darstellung zu gewährleisten (siehe Abschnitt 2.3.2). Die normalisierten Werte werden als Zeitreihen, Heatmap und Kreisdarstellung dargestellt.

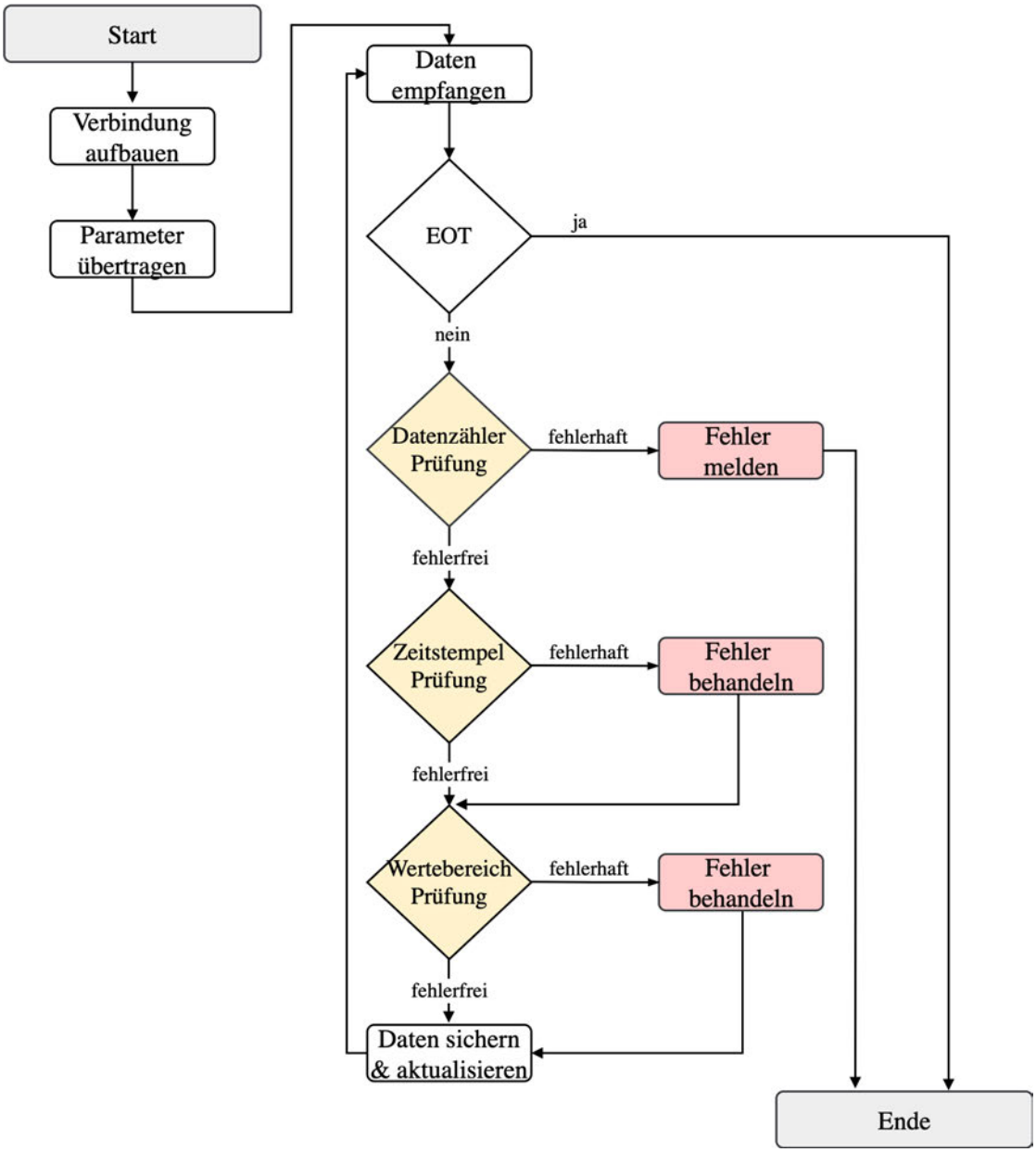


Abbildung 3.11: Kontrollflussdiagramm der Receiver-Klasse. Auf der linken Seite sind die Prozesse der Initialisierung und dem Verbindungsaufbau zu sehen. Auf der rechten Seite befindet sich die Schleife des Datenempfangs. Gelb markiert ist die mehrstufige Validierung der empfangenen Daten. Bei Fehlern werden entsprechende Maßnahmen sowie Fehlerprotokollierung durchgeführt. Diese Prozesse sind rot markiert.

3.7.2 Kommunikationssequenz zwischen Software und Firmware

Die Kommunikation zwischen der Firmware und Software erfolgt über den REPL. Der Austausch beginnt mit der Übertragung und Bestätigung der Konfigurationen und Parameter. Anschließend startet die Firmware mit der Temperaturmessung mittels `read_temperature()` für alle acht Sensoren hintereinander. Für jeden der acht Sensoren werden dabei dieselben Konfigurationen mittels `setup_sensors()` vorgenommen und es beginnt die Datenerfassung.

Die Schleife, die bis zur angegebenen Datenmenge (`DATA_SIZE`) durchlaufen wird und gemeinsam mit einem Timer-gesteuerten Wartezyklus arbeitet, steuert die getaktete Datenerfassung. Der Software-Timer greift auf den internen Hardware-Timer des Raspberry Pi Picos zu, um Zeitstempel mit Millisekunden-Genauigkeit zu erzeugen. Dabei ist der `wait_flag` essentiell, da dieser sicherstellt, dass zwischen den Messungen exakt 10 ms vergehen, bevor die nächste Datenerfassung mittels `check_sensors()` für alle acht Sensoren hintereinander durchgeführt wird.

Die `data.append()`-Funktion sammelt dabei die Werte aller Sensoren in dem Ausgabestring. Die erfassten Daten werden anschließend über `print()` in die REPL ausgegeben, wodurch sie von der Software empfangen werden können. Die `Receiver`–Klasse empfängt in jedem Zyklus einen kompletten Satz von 16 Messwerten (8 für rote und 8 für Infrarot-Sensoren), den Zeitstempel und den Zähler.

Die Durchschnittslänge des ausgegebenen Strings ist ungefähr 100 Zeichen. Bei der konfigurierten Baudrate von 115200 des Raspberry Pi Pico können ungefähr 115 Zeichen pro 10 ms-Intervall übertragen werden, wodurch keine Übertragungsengpässe entstehen sollten. Der Abschluss der Kommunikation erfolgt durch das Senden des EOT-Zeichens. Alle Prozesse werden hiernach beendet und die Sensoren ausgeschaltet.

Auf das Sequenzdiagramm in Abb. 3.12 ist der Datenaustausch zwischen den `Receiver`–, `DataCollector`–, `SensorConfig`– und `MAX30102`–Klassen dargestellt.

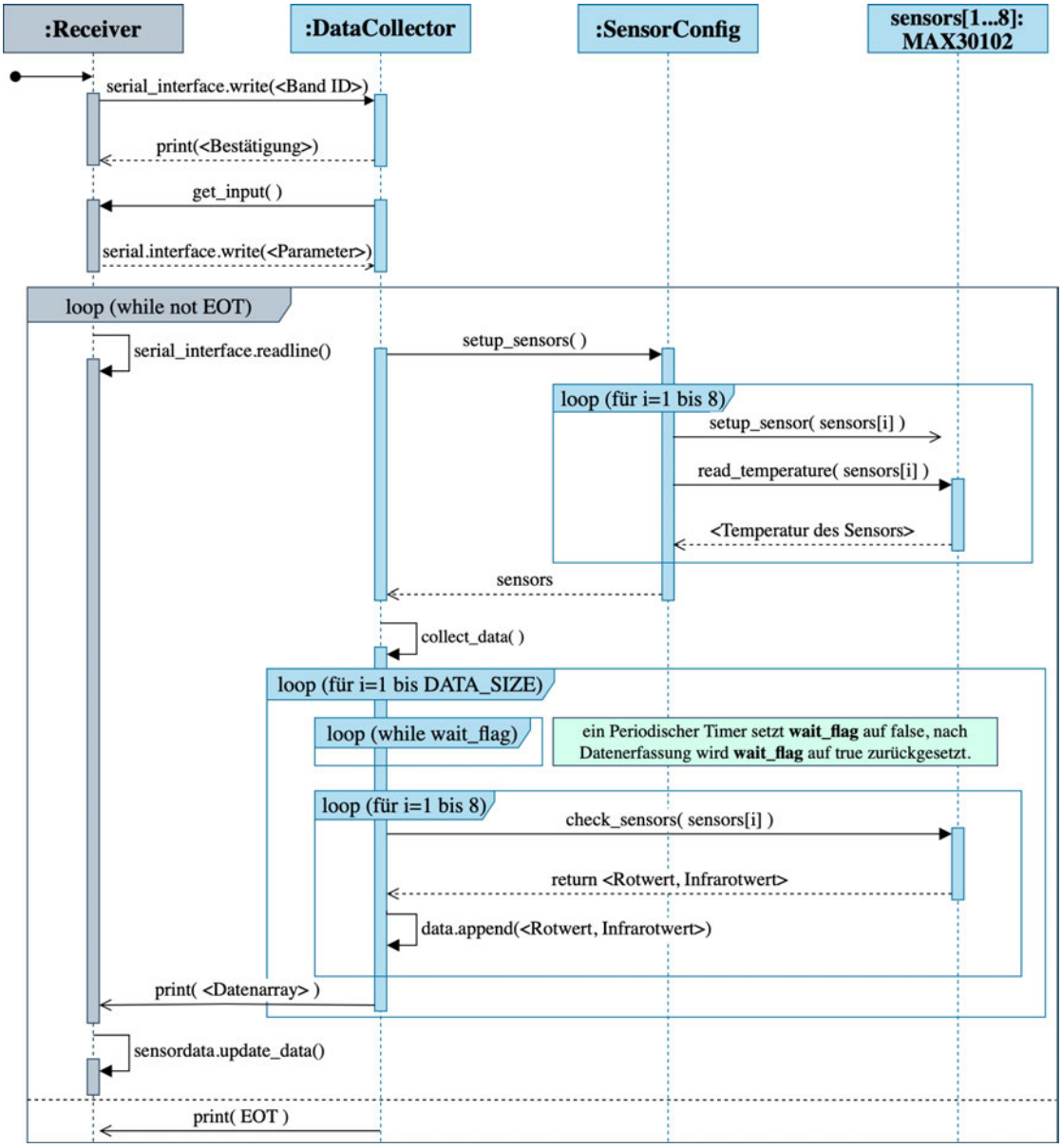


Abbildung 3.12: Sequenzdiagramm der Kommunikation zwischen Software und Firmware. Die Software-Klassen und -Prozesse sind grau markiert, die Firmware-Klassen und -Prozesse blau markiert. Das Diagramm zeigt die Initialisierung, die schleifenbasierte Sensorkonfiguration sowie die timergesteuerte periodische Datenerfassung. Die Timer-Steuerung erfolgt durch eine separate Klasse. Fehlerbehandlung und Debugging-Prozesse sind nicht dargestellt. Die Kommunikation wird durch eine EOT-Nachricht beendet.

3.7.3 Graphische Oberfläche

Die entwickelte grafische Oberfläche ist so gestaltet, dass Probanden die auszuführenden Gesten erkennen können. Links werden 8 Reihen Messwerte der Sensoren visualisiert, wobei rote Werte als durchgezogene rote Linie und Infrarot-Werte als dunkelrote gestrichelte Linie dargestellt werden. Pro Sensor sind 200 Messpunkte sichtbar, was die Beobachtung der Signalveränderungen ermöglicht. Im oberen rechten Bereich befindet sich neben dem Logo sowohl ein Symbol als auch ein Bild der aktuell auszuführenden Geste. Zusätzlich geben drei verschiedene Statussymbole den aktuellen Zustand der Messung an. Das Startsymbol zeigt an, dass Daten aufgezeichnet werden. Das Warnsymbol weist auf mögliche Fehler während der Messung hin. Das Pausensymbol signalisiert, dass momentan keine Daten erfasst werden. Rechts in der Mitte wird eine Heatmap mit 400 Messpunkten dargestellt. Unten rechts werden die Messdaten in einer kreisförmigen Anordnung visualisiert, die die Platzierung der Sensoren am Unterarm entsprechen soll. Die Visualisierung wird mit einer Bildwiederholungsrate von 100 Hz aktualisiert. Der gesamte Ablauf wird durch die `SystemManager`-Klasse automatisiert gesteuert. In Abbildung 3.13 ist die grafische Oberfläche dargestellt

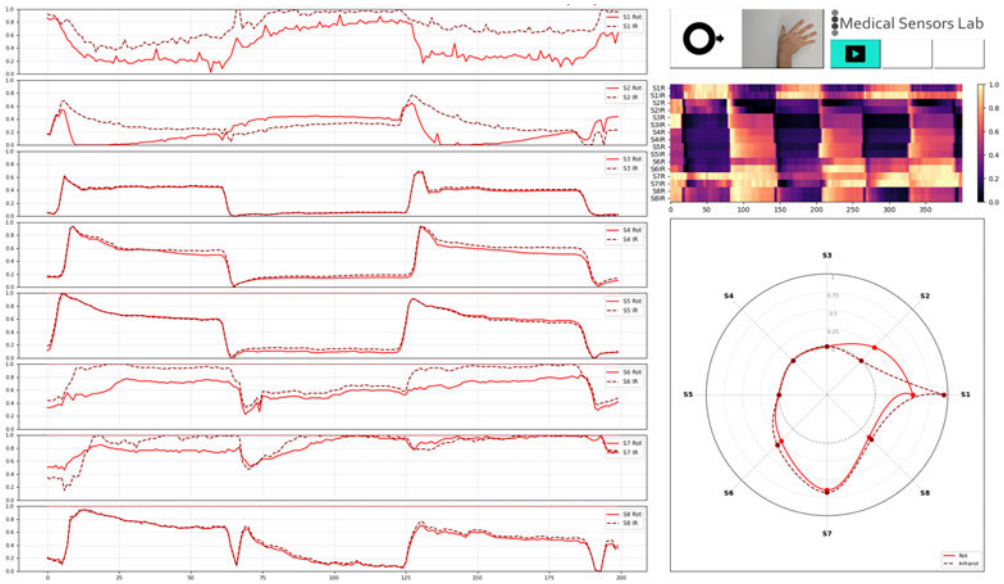


Abbildung 3.13: Grafische Oberfläche. Links befinden sich 8 Reihen, in denen die Rot- und Infrarot-Werte angezeigt werden. Oben rechts befinden sich das aktuelle Gestensymbol und -bild sowie die drei verschiedenen Statussymbole. In der Mitte rechts sich die Heatmap der 16 Kanäle. Unten rechts werden die Messdaten in einer kreisförmigen Anordnung visualisiert.

3.7.4 Testsimulation mit der MockReceiver-Klasse

Die `MockReceiver`–Klasse ermöglicht das Testen des Systems ohne angeschlossene Hardware. Sie simuliert Sensordaten durch die Wiedergabe bereits erfasster Messwerte und erleichtert somit Debugging während der Entwicklungsphase. Die Funktionsweise basiert auf dem Import von CSV-Dateien, die vom System selbst generiert wurden. Bei der Initialisierung wird der Dateipfad zur gewünschten CSV-Datei übergeben. Die Methode `simulate_data()` startet die zeitstempelbasierte Wiedergabe aller Sensordaten und orientiert sich dabei an den in der Datei gespeicherten Zeitintervallen, um realistische Erfassungsbedingungen nachzuahmen. Die Simulation läuft automatisch bis zum Ende der verfügbaren Daten. Die verwendete Datenstruktur, die das Sensorsystem erzeugt, gewährleistet, dass die simulierten Daten echte Messwerten entsprechen.

In der Systemarchitektur verwendet die `MockReceiver`–Klasse dieselbe Schnittstelle wie die `Receiver`–Klasse und kann diese nahtlos ersetzen, ohne Änderungen an anderen Komponenten zu erfordern. Die Datenübertragung an `SensorData`-Objekte entspricht einer Observer-Observable-Beziehung, wodurch die gleichen Verarbeitungspfade wie im einer echten Messung durchlaufen werden. Die `MockReceiver`–Klasse ist im Klassendiagramm in Abb. 3.14 dargestellt.

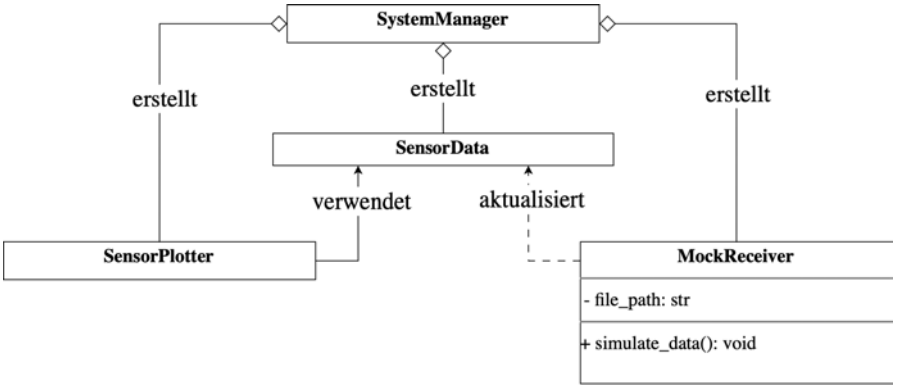


Abbildung 3.14: Klassendiagramm mit Integration der Mock-Klasse. Das Diagram visualisiert die Architektur in seiner Simulationskonfiguration. Die MockReceiver-Klasse simuliert Sensordaten aus gespeicherten Dateien und aktualisiert das zentrale SensorData-Objekt.

3.8 Signalverarbeitung

Die Photodioden in den Sensoren erzeugen kontinuierliche Signale, die durch den integrierten ADC des Sensors in diskrete digitale Werte umgewandelt werden. Wie in den vorherigen Abschnitten beschrieben, unterliegen diese digitalen Rohdaten verschiedenen Störeinflüssen wie Rauschen, Ausreißern und systembedingten Schwankungen, die eine weitere Aufbereitung erforderlich machen. In Abbildung 3.15 ist die implementierte Signalverarbeitungskette dargestellt, die aus drei verschiedenen aufbauenden Schritten besteht. In den folgenden Abschnitten wird auf die Wahl und Implementierung dieser einzelnen Signalverarbeitungsschritte eingegangen.

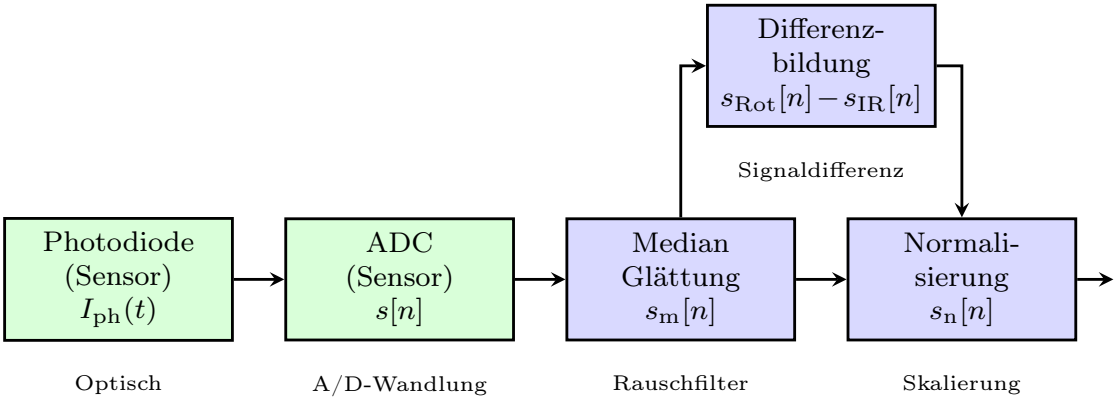


Abbildung 3.15: Signalverarbeitungskette. Das optische Signal wird durch Photodioden in elektrische Ströme umgewandelt und vom ADC digitalisiert. Diese Schritte sind in grün abgebildet. Anschließend werden die Signale geglättet, das Differenzsignal gebildet und alle Signale normalisiert. Diese Schritte auf der Software werden in blau dargestellt.

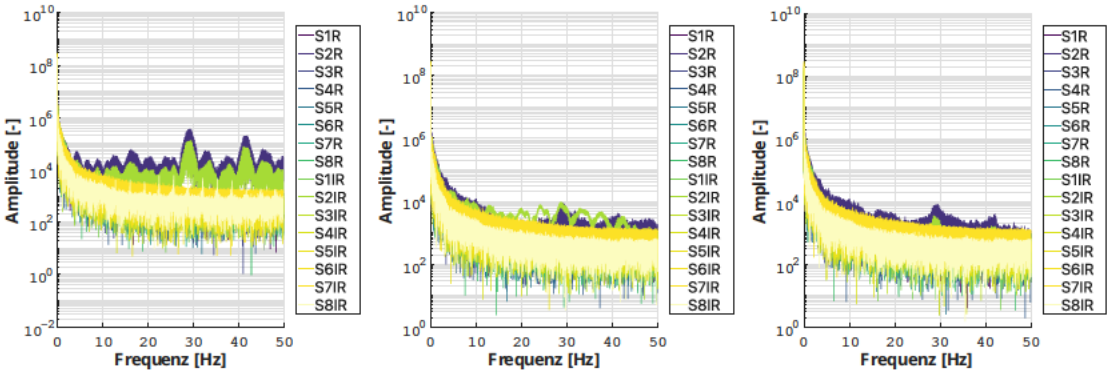
3.8.1 Median-Glättung und optimale Fensterbreite

Wie in 2.3.1 beschrieben, eignet sich die Median-Glättung zur Beseitigung von Ausreißern und zur Dämpfung höherer Frequenzanteile. Da es sich hierbei um einen nichtlinearen Filter handelt, wird der Einfluss der Median-Glättung auf das Frequenzspektrum der Messsignale systematisch untersucht, um die optimale Fensterbreite zu bestimmen.

Für diese Analyse wird eine Messung mit 100 Hz-Abtastrate verwendet, in der alle definierten Gesten jeweils fünfmal ausgeführt werden. Die Untersuchung erfolgte durch grafische Analyse der Frequenzspektren mittels FFT. Beginnend mit einer Fensterbreite von

3 (30 ms) wird diese schrittweise erhöht, bis ein zufriedenstellendes Ergebnis erzielt wird. Ziel ist es bei dem die Gestensignale in den FFT-Spektren das Rauschen ausreichend zu unterdrücken. Die Fensterbreite wird ausschließlich als ungerade Zahl gewählt, um ein symmetrisches Verhältnis zu gewährleisten. Asymmetrische Fensterbreiten führen zu unerwünschten zeitlichen Verschiebungen im gefilterten Signal, da der Medianwert nicht zentriert berechnet werden kann. Bei der Bewertung der Filterergebnisse liegt der Fokus auf der Erhaltung niederfrequenter Signalanteile, während die Fensterbreite so klein wie möglich gehalten wird, da größere Fensterbreiten zu Verzögerungen in der Größenordnung der halben Fensterbreite multipliziert mit der Abtastperiode (10 ms) führen.

Die Ergebnisse der FFT-Analyse sind in Abb. 3.16 dargestellt. Das ungeglättete Spektrum in Abb 3.16a weist bei den Sensorsignalen unterschiedlich starke Rauschanteile im Frequenzbereich bis 50 Hz auf. Durch die Medianfilterung mit Fensterbreite 9 (siehe Abb. 3.16b) und 13 (siehe Abb. 3.16c) werden die Frequenzverläufe zunehmend gleichmäßiger, wobei die wellenförmigen Muster, die bei mindestens zwei Sensorsignalen im ungeglätteten Spektrum noch sichtbar waren, zunehmend geglättet werden.



(a) Ohne Glättung (b) Fensterbreite 9 (c) Fensterbreite 13

Abbildung 3.16: FFT-Analyse zur Bestimmung der optimalen Median-Filterfensterbreite. Vergleich des Frequenzspektrums (a) ohne Glättung, (b) mit Median-Filter der Fensterbreite 9 (90 ms)und (c) der Fensterbreite 13 (130 ms). Alle Sensorsignale zeigen unterschiedliche spektrale Charakteristika mit dominanten niederfrequenten Anteilen unter 10 Hz. Die Filterung reduziert die hohen Rauschanteile, während die niederfrequenten Signalanteile der Gesten erhalten bleiben.

3.8.2 Differenzbildung der Signale

Bevor die Daten normalisiert werden, wird für jeden Sensor die Differenz aus dem roten und dem infraroten Wert gebildet und gespeichert, sodass die relative Beziehung zwischen Rot und Infrarot erhalten bleibt. Obwohl PPG-Sensoren anfällig für Druck sein können, wird das Verhältnis zwischen Rot- und Infrarot-Werten nicht beeinflusst, da auf beide der gleiche Druck ausgeübt wird. Die Merkmale bei diese Signalart sind den Reflexionseigenschaften zuzuschreiben. Im Datensatz werden sie als dritter Signaltyp ergänzt und anschließend genauso wie rote und infrarote Signale normalisiert. Die Differenzbildung erfolgt für jeden Sensor i mittels:

$$\mathrm{SiD}_i = \mathrm{SiR}_i - \mathrm{SiIR}_i \quad \text{für } i = 1, 2, \ldots, n$$

wobei SiR_i das rote Signal, SiIR_i das infrarote Signal und SiD_i das resultierende Differenzsignal des i-ten Sensors darstellt.

3.8.3 Daten-Normalisierung

Wie in den Abschnitt 2.3.2 beschrieben, ist die Min-Max-Normalisierung als einer der einfachste Methode zur Umsetzung, da sie ohne aufwändige Berechnungen auskommt. Durch die Min-Max-Normalisierung werden alle Signale auf eine gemeinsame Skala zwischen 0 und 1 gebracht, was eine gängige Praxis bei Machine Learning und Mustererkennungsverfahren darstellt [18]. Ein wesentlicher Vorteil der Min-Max-Normalisierung liegt in ihrer praktischen Anwendbarkeit in Verbindung mit dem Sliding-Window-Verfahren. Neu auftretende Minimal- und Maximalwerte können dynamisch erfasst und gespeichert werden, wodurch eine kontinuierliche Anpassung der Normalisierung möglich ist. Nach einer kurzen Kalibrierungsphase können alle relevanten Min- und Max-Werte erfasst werden. Die zuvor durchgeführte Median-Glättung trägt zusätzlich dazu bei, dass die normalisierten Werte nicht durch Ausreißer verzerrt werden.

3.9 Gestenerkennung mittels Convolutional Neural Network

Die Implementierung der Gestenerkennung erfolgt in Google Colab unter Verwendung von TensorFlow und Keras. Für die Datenvorbereitung werden alle gemessenen Signale

mit Matlab vorverarbeitet und als CSV-Gruppen aufgeteilt. Jede CSV-Datei enthält eine 10x24 Matrix mit Gestenklassen, die aus dem ersten Buchstaben der Dateinamen extrahiert werden. 50% der Daten werden als Trainingsdaten getrennt, die verbleibenden 50% wurden anschließend zur Hälfte in Test- und Validierungsdaten aufgeteilt.

Die entwickelte neuronale Netzwerk-Architektur wird aus der Vorlage aus dem Buch [18] angepasst. Die Eingabeform von 10x24 entspricht 10 Zeitschritte und 24 unterschiedliche Signale, die sich aus drei Gruppen der Rot-, Infrarot- und Differenzwerte herleiten. In Abbildung 3.17 ist die ausgewählte Architektur dargestellt. Die theoretischen Grundlagen zu den verwendete Schichten, ReLU-Aktivierungsfunktionen und Regularisierungsmaßnahmen werden in Abschnitt 2.3.3 erläutert.

Das Modell wird mit dem Adam-Optimierer kompiliert und als Verlustfunktion wird `sparse_categorical_crossentropy` gewählt, da die Zielklassen als Ganzzahlen vorliegen. Die Accuracy-Metrik wurde zur Überwachung der Trainingsperformance verwendet.

Das Training erfolgt über maximal 300 Epochen mit einer Batch-Größe von 64. Ein Early Stopping Callback wird konfiguriert, der das Training automatisch beendet, wenn sich die Validierungsverluste über 20 aufeinanderfolgende Epochen nicht um mindestens 0,005 verbessern (`min_delta=0.005`). Dabei werden die besten Gewichte automatisch wiederhergestellt (`restore_best_weights=True`).

Zusätzlich wird ein `ReduceLROnPlateau` Callback eingesetzt, welcher die Lernrate um einen Faktor von 0,2 reduziert, wenn sich die Validierungsverluste über 3 aufeinanderfolgende Epochen nicht verbessern. Die minimale Lernrate wird auf 1×10^{-6} gesetzt, um eine zu starke Reduzierung zu vermeiden.

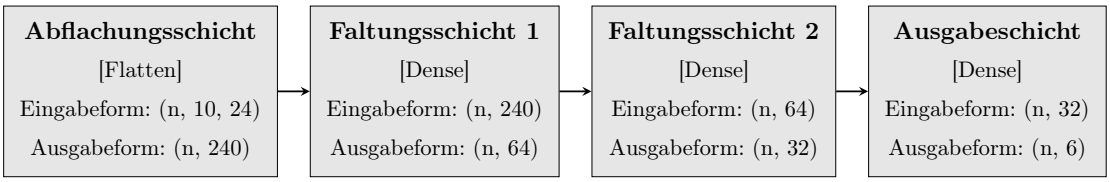


Abbildung 3.17: Neuronale Netzwerk-Architektur. Das Symbol n steht für die variable Anzahl von Eingabesamples pro Verarbeitungsschritt.

4 Charakterisierung

Im Rahmen der Charakterisierung werden die technischen Eigenschaften des entwickelten Sensorsystems untersucht. Ziel ist es, optimale Betriebsparameter zu ermitteln, die Leistungsfähigkeit entsprechend den Anforderungen zu bewerten und mögliche Fehlerquellen zu beseitigen. Der Fokus liegt dabei auf der Anpassung der MAX30102-Sensoren für die Detektion muskulärer Aktivität.

Für die Charakterisierungsmessungen wird ein elektrochromes Glas verwendet, auch bekannt als schaltbares Glas, dessen Lichtdurchlässigkeit elektrisch steuerbar ist. Ein Arbitrary Waveform Generator (SIGLENT SDG2122X [54]) erzeugt dafür ein Rechtecksignal mit einer Frequenz von 1 Hz und einer Amplitude von 10 V, wodurch die Transparenz des Glases periodisch verändert wird. Unter dem Glas wird ein Stück rotes Papier platziert, das als definierte Reflexionsfläche dient. Die Sensoren im Armband werden fest am Glas platziert, um konstante und reproduzierbare Messbedingungen zu gewährleisten. Dieser Messaufbau ist in Abb. 4.1 dargestellt.

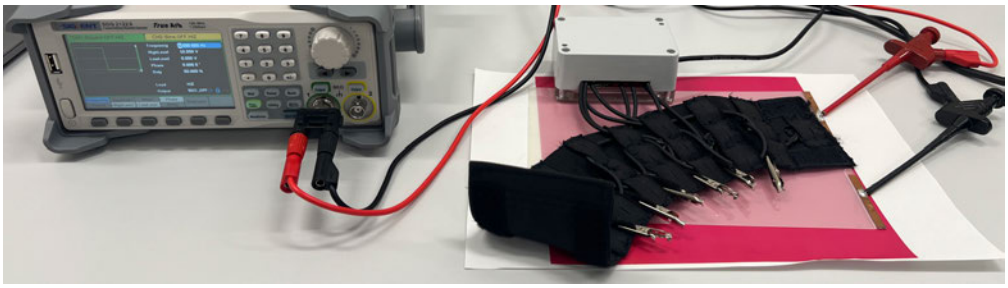


Abbildung 4.1: Messaufbau für die Charakterisierung der MAX30102-Sensoren. Das Armband mit den integrierten Sensoren liegt auf einem elektrochromen Schaltglas, dessen Transparenz über den angeschlossenen Signalgenerator (SIGLENT SDG2122X) gesteuert wird. Unter dem Glas befindet sich ein rotes Papier als definierte Reflexionsfläche.

4.1 Einstellungsparameter der Sensoren

Bei der Einstellung der Sensoren liegt der Fokus auf der Optimierung der Parameter LED-Intensität (oder Helligkeit), ADC-Messbereich und ADC-Integrationszeit. Die Samplingrate der Sensoren wird zunächst auf 400 Hz gesetzt. Die Konfiguration basiert auf dem Zusammenhang zwischen Signalstärke und Messbereich (siehe 3.3.3). Die Erhöhung der LED-Intensität sowie eine Verlängerung der Integrationszeit führen zu einem empfindlicheren Sensorsignal. Der ADC-Messbereich muss entsprechend dimensioniert werden, damit keine Sättigung eintritt. Im Sättigungsfall wird der maximale Wert des ADCs erreicht, wodurch Unterschiede nicht mehr erfasst werden können und relevante Signalinformationen verloren gehen. In Abbildung 4.2 sind Messergebnisse dargestellt, die das Verhalten der Sensoren bei den zwei höchsten LED-Intensitäten zeigen. Bei höchster LED-Intensität (50,0 mA) in Kombination mit einem ADC-Messbereich von 16384 nA und einer Integrationszeit von 411 µs tritt eine Sättigung des Sensors auf und Maximalwerte überschreiten den Messbereich. Eine LED-Intensität von 25,4 mA (Stufe 3), kombiniert mit dem maximalen ADC-Messbereich von 16384 nA (Stufe 4) und der längsten Integrationszeit von 411 µs (Stufe 4) liefert gute Ergebnisse. Die lange ADC-Integrationszeit maximiert die effektive Auflösung auf 18 Bit, was Stromänderungen von 62,5 pA wahrnehmen kann und somit die präziseste Einstellung des Systems darstellt.

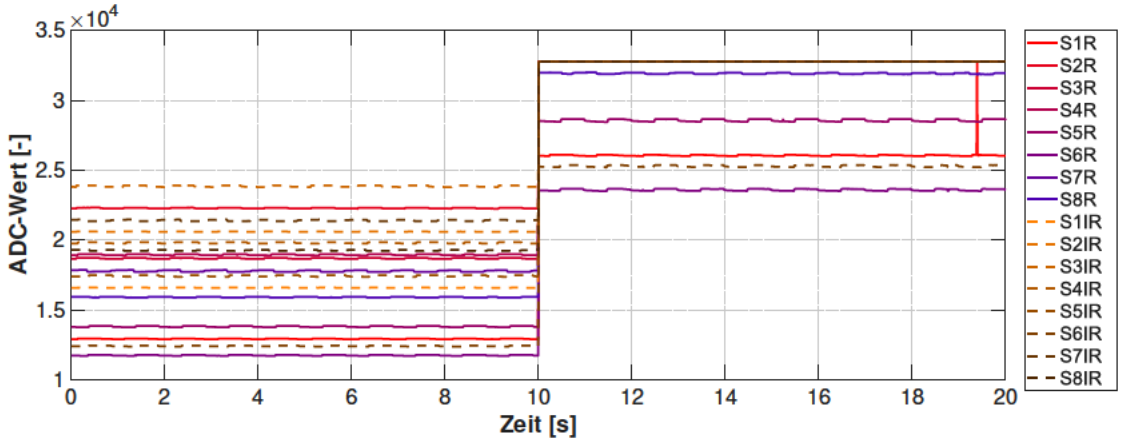


Abbildung 4.2: Vergleich der LED-Intensität. Die ersten 10 Sekunden werden mit LED-Intensität Stufe 3 (25,4 mA) durchgeführt, die nächsten 10 Sekunden mit der LED-Intensität Stufe 4 (50,0 mA). Die durchgezogenen Linien repräsentieren die Rotsignale, die gestrichelten die Infrarotsignale. Bei der höheren Intensität ist eine deutliche Sättigung mehrerer Sensoren erkennbar.

4.2 Stabilitätsmessung

Mit dem gleichen Aufbau wird eine Messung über einen Zeitraum von 20 Minuten durchgeführt, was bei einer Systemabtastrate von 100 Hz insgesamt 120 000 Messreihen ergibt. Dieser Zeitraum deckt die doppelte Zeit, die normalerweise das System bei einer Probandenmessung im Betrieb ist. Diese Messung dient der Überprüfung der Systemstabilität unter kontinuierlicher Belastung und der Identifikation potenzieller Fehlerquellen bei längerer Betriebsdauer. In der Messung kann beobachtet werden, dass gelegentlich Werte aus ihrem typischen Bereich ausbrechen und in den Bereich des Signals der anderen LED desselben Sensors fielen. Die fehlerhaften Werte sind nicht identisch mit dem Wert der anderen LED, sondern im selben Bereich. Die Abweichung kann in einem Bereich zwischen 0 und 30 ADC-Einheiten beobachtet werden, was die Ursache von Fehlern durch Duplizierung ausschließt. Da beide LED-Daten in I2C-Abfrage aus einem Array gelesen werden, deuten die Störungen auf ein internes Sensorproblem hin. Das Debugging der `MAX30102`-Klasse bestätigt, dass der Sensor LED-Werte an falschen Array-Positionen speichert. Da dieser Fehler in der Sensordokumentation nicht beschrieben ist und bei etwa 0,01% aller Abfragen auftritt, wird eine softwarebasierte Fehlerbehandlung in der `Receiver`−Klasse implementiert. Die Fehlerkennung prüft die absolute Differenz zwischen Rot- und Infrarot-Werten gegen einen Toleranzwert. Da die Signalbereiche normalerweise deutlich getrennt sind, deutet eine geringe Differenz einen Sensorfehler an. Ein Schwellwert von 30 erkennt die Mehrheit dieser Fehler zuverlässig. Von einer Erhöhung des Schwellwerts wird bewusst abgesehen, um das Risiko zu vermeiden, valide Datenpunkte fälschlicherweise als Fehler zu klassifizieren. Fehler, die außerhalb dieses Bereichs liegen könnten, werden durch spätere Filtermechanismen behandelt. In solchen Fällen wird der fehlerhafte Infrarot-Wert durch den letzten gültigen Wert ersetzt, der Fehler wird protokolliert und ein Warnflag gesetzt. Diese Art von Fehlern macht die Verwendung der FIFO-Mittelwertbildung des Sensors unbrauchbar. In Abbildung 4.3 ist die Messung vollständig dargestellt. Eine nähere Betrachtung des Fehlers ist in Abb. 4.4 dargestellt.

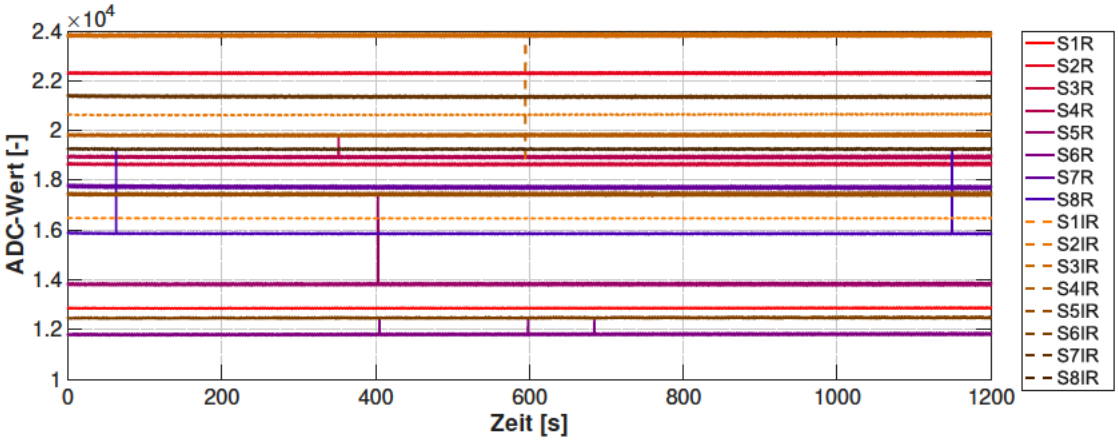


Abbildung 4.3: Stabilitätsmessung aller acht Sensoren über 20 Minuten bei 100 Hz Abtastrate (120.000 Messreihen). Die durchgezogenen Linien repräsentieren die Rotsignale, die gestrichelten die Infrarotsignale. Vereinzelte Ausreißer sind erkennbar, bei denen die Infrarot-Werte kurzfristig aus dem erwarteten Wertebereich springen.

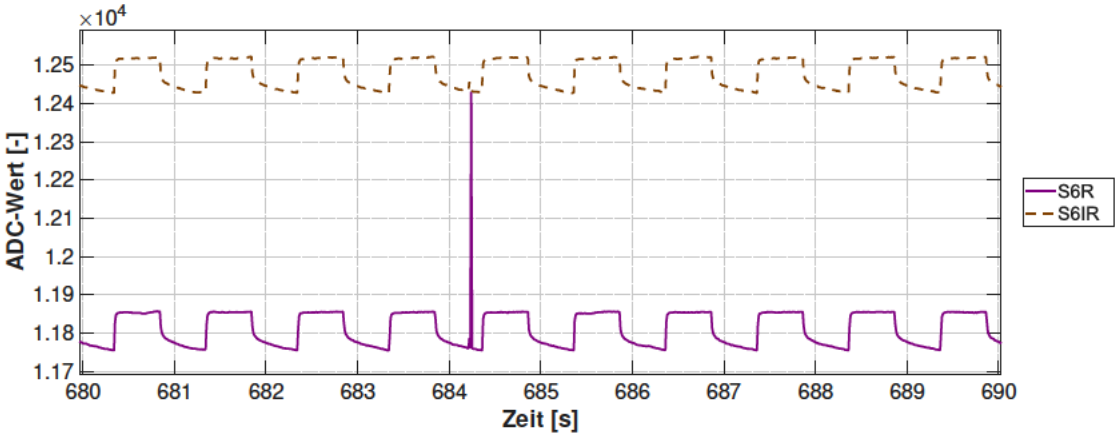


Abbildung 4.4: Detailansicht eines typischen Sensorfehlers am Sensor 6 in Abb. 4.3 zwischen Sekunde 680 und Sekunde 690. Das Rotsignal (durchgezogene Linie) springt kurzzeitig in den Wertebereich des Infrarotensignals (gestrichelte Linie).

4.3 Temperaturbetrachtung

Die MAX30102-Sensoren verfügen über einen integrierten Temperatursensor. Zur Analyse der thermischen Eigenschaften werden die Temperaturwerte aller acht Sensoren während verschiedener Messreihen protokolliert. Die erfassten Temperaturdaten von 10 verschiedenen Messungen werden in Tabelle 4.1 dargestellt und deuten auf eine Fehlfunktion des Sensors 2 hin. Während die restlichen Sensoren im plausiblen Bereich zwischen etwa 28°C und 35°C arbeiten, weicht Sensor 2 mit Werten zwischen 14,5°C und 19,4°C ab. Die konsistente Abweichung von mehr als 10°C gegenüber den anderen Sensoren deutet auf einen Defekt im Temperatursensor. Als Konsequenz dieser Betrachtung wird der Sensor in der letzten Iteration des Armbandes ausgetauscht.

Während der 20-Minuten-Messung (siehe Abschnitt 4.2) ist Sensor 2 bereits ausgetauscht worden. Der Temperaturanstieg während der Messung wird beobachtet. Der durchschnittliche Temperaturanstieg über alle Sensoren beträgt 0,73°C. Diese Erwärmung ist im erwarteten Bereich und belegt, dass keine Probleme bezüglich der erzeugten Wärme bei der Messung an Probanden zu erwarten sind. Diese Messwerte sind in Tabelle 4.2 ersichtlich.

Tabelle 4.1: Statistische Auswertung der Temperaturdaten aller Sensoren, basierend auf 10 verschiedenen Messreihen. Eine Abweichung bei Sensor 2 ist erkennbar und wird in fett markiert.

Eigenschaft	**Sensor 1**	**Sensor 2**	**Sensor 3**	**Sensor 4**	**Sensor 5**	**Sensor 6**	**Sensor 7**	**Sensor 8**
Mittelwert (°C)	30,90	**17,06**	33,71	31,89	31,93	32,66	30,58	31,59
Minimum (°C)	28,81	**14,50**	30,94	29,94	28,63	29,25	28,06	30,19
Maximum (°C)	32,56	**19,38**	34,63	32,81	32,81	33,94	31,81	32,63

Tabelle 4.2: Temperaturwerte der Sensoren zu Beginn und Ende der 20-minütigen Messung, sowie der berechnete Temperaturanstieg.

Messwert	**Sensor 1**	**Sensor 2**	**Sensor 3**	**Sensor 4**	**Sensor 5**	**Sensor 6**	**Sensor 7**	**Sensor 8**
Starttemperatur (°C)	23,75	23,94	26,06	25,69	25,88	26,25	23,31	26,88
Endtemperatur (°C)	24,75	24,94	26,63	27,13	25,81	26,75	24,94	26,69
Temperaturanstieg (°C)	1,00	1,00	0,57	1,44	-0,07	0,50	1,63	-0,19

4.4 Sensor-Pin-Zuordnung und Signalverifikation

Die Pin-Belegung muss mit der Anordnung der Sensoren auf dem Armband übereinstimmen (siehe Schaltplan in Abb. 3.5). Für diese Überprüfung wird mit Hilfe des Parame-

ters `sensor_led_toggle` ein gezieltes Aktivieren und Deaktivieren einzelner Sensoren durchgeführt. Die Messung erfolgt mit den Aufbau aus dem vorherigen Abschnitten über 24 Sekunden. Dabei wird sequenziell immer nur ein einzelner Sensor für 3 Sekunden aktiviert, während alle anderen deaktiviert werden.

Die erkennbaren sequenziellen Ausschläge der einzelnen Sensoren bestätigen, dass sowohl die Verbindungen der Sensoren als auch die Pin-Zuordnung in der Firmware korrekt implementiert werden. Jeder Sensor reagiert ausschließlich dann, wenn er durch das entsprechende Bitmuster im `sensor_led_toggle`-Parameter aktiviert wird. Dies ist wichtig für die spätere räumliche Zuordnung der Sensorsignale am Unterarm. Die Ergebnisse dieser Messung sind in Abb 4.5 dargestellt.

In Abbildung 4.5 lassen sich weitere Erkenntnisse über die Rot- und Infrarot-Wertebereiche gewinnen. Bei der Mehrheit der Sensoren liegen die Infrarot-Werte (gestrichelte Linien) erwartungsgemäß deutlich über den Werten der roten Kanäle (durchgezogene Linien). Dieser Unterschied erklärt sich aus den physikalischen Eigenschaften der verwendeten Sensoren und die unterschiedlichen Reflexions- und Absorptionscharakteristik von rotem und infrarotem Licht. Bei Sensor 2, 3 und 7 ist jedoch eine Umkehrung dieses Musters erkennbar. Die Werte des vermeintlich Rotsignals liegen hier über denen des Infrarotsignals.

Diese Abweichung vom erwarteten Verhalten deutet auf eine Vertauschung der Signalzuordnung hin, vermutlich verursacht durch einen Fertigungsfehler im Sensor. In einem Bug Report [49] wird vermerkt, dass mehrere Anwender des MAX30102 auf diesen Produktionsfehler gestoßen sind.

Wenn den Sensor nur im Rot-LED-Modus betreibt, lässt eine Sichtprüfung erkennen, welche Sensoren fehlerhaft sind, da infrarotes Licht kaum sichtbar ist. In Abbildung 4.6 ist dieser Test dargestellt. Die LEDs, die nicht aufleuchten aber trotzdem Messergebnisse liefern, sind fehlerhaft.

Um diese Inkonsistenz zu beheben, ohne Hardware-Änderungen vornehmen zu müssen, wird eine softwarebasierte Lösung implementiert. In der `HardwareManager`-Klasse wird eine Konfigurationsstruktur hinterlegt, die für jeden Sensor die korrekte Zuordnung definiert. Diese Konfiguration ermöglicht es, bereits während des Auslesevorgangs die Signale zu korrigieren, sodass die nachfolgenden Verarbeitungsschritte mit konsistenten Daten arbeiten können.

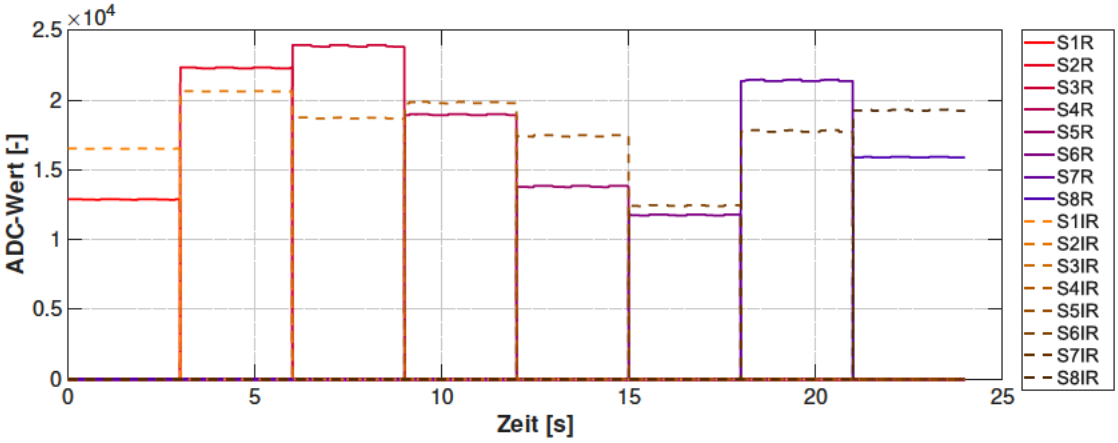


Abbildung 4.5: Sequenzielle Aktivierung der einzelnen Sensoren. Die Messreihe umfasst etwa 24 Sekunden, wobei nur ein Sensor für 3 Sekunden aktiviert wird. Die durchgezogenen Linien zeigen die Werte der Rotsignale, die gestrichelten die der Infrarotsignale.

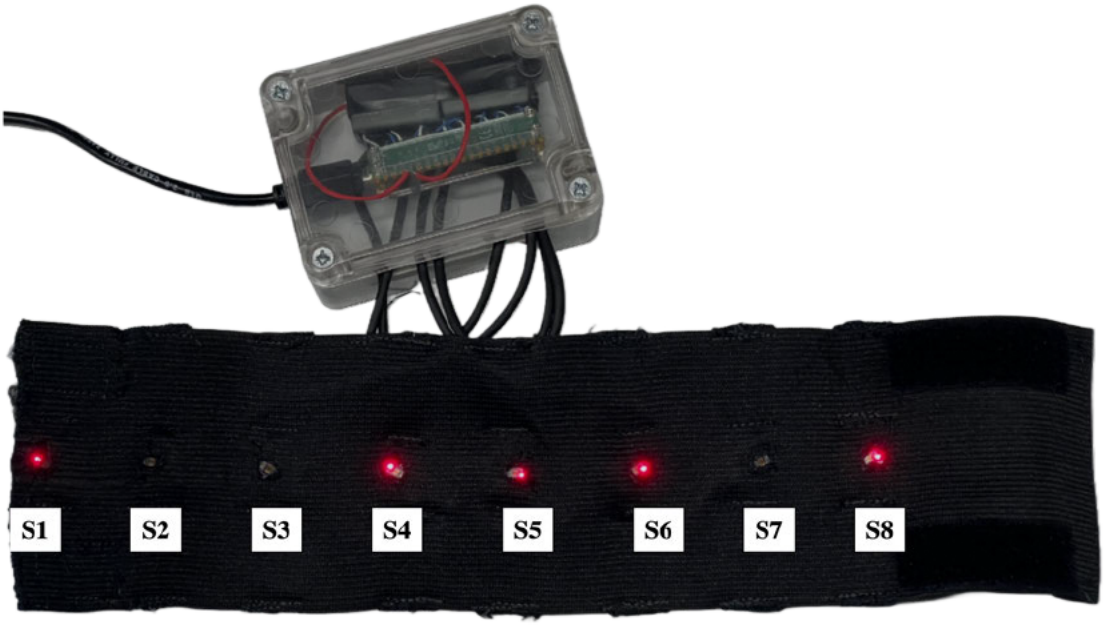


Abbildung 4.6: Visuelle Prüfung der Sensor-LED-Zuordnung im Rot-LED-Modus. Sensoren mit vertauschter Rot- und Infrarot-Zuordnung zeigen keine sichtbare Aktivierung der roten LED (Sensoren 2, 3 und 7).

4.5 Synchronisationsanalyse

Ein wichtiger Aspekt des Systems ist die taktsynchrone Erfassung aller Sensoren, um eine Korrelation der Signale zu ermöglichen. Da die Datenerfassung aus technischen Gründen sequentiell erfolgt, muss analysiert werden, ob die zeitliche Differenz vernachlässigbar ist, um eine akzeptable Erfassung zu gewährleisten.

Der Messaufbau wird für diese Messung angepasst. Die Fläche des elektrochromen Glases ist nicht ausreichend um alle Sensoren im Band gleichmäßig zu platzieren, deswegen werden zwei Messungen durchgeführt. Sensor 1 bis 5 werden in der ersten Messung betrachtet und Sensor 4 bis 8 in der zweiten. Somit dienen Sensor 4 und 5 als gemeinsamer Nenner für die Analysen. Die Rechteck-Welle des Signalerzeugers wird auf 0,5 Hz umgeschaltet.

Die meisten Einstellungen der Sensoren aus Abschnitt 4.1 werden übernommen. Entscheidend ist die Erhöhung der Samplingrate der Sensoren auf 400 Hz. Bei einer Sensorsamplingrate von 100 Hz reagieren die Sensoren nachweislich nicht schnell genug auf Veränderungen. Das System arbeitet jedoch weiterhin im 10 ms Takt, was eine Abtastfrequenz von 100 Hz darstellt. Sensoren mit einer höheren Samplingrate arbeiten im Hintergrund weiter und füllen den FIFO-Buffer. Laut Datenblatt [38] werden ältere Daten im FIFO der Sensoren überschrieben, wenn die `fifo_rollover` Einstellung einschaltet. Die Signale werden bei der Betrachtung zuvor geglättet und normalisiert.

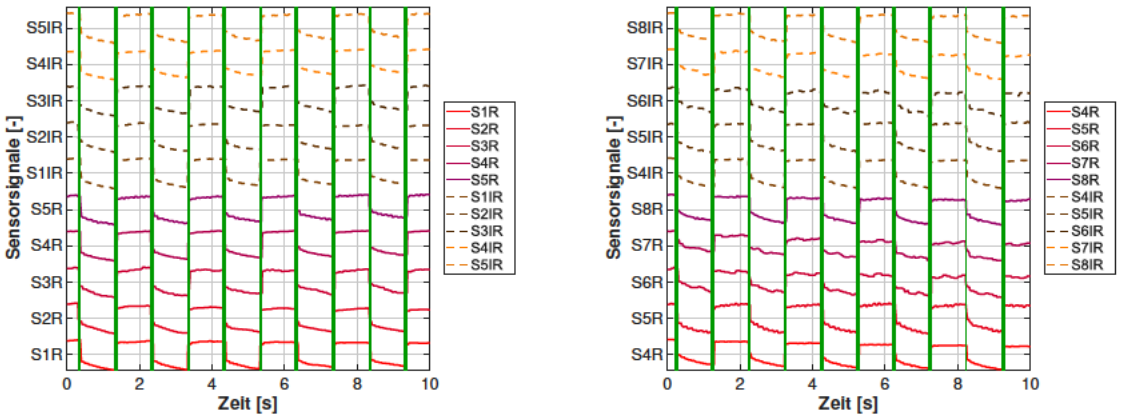
Die Methode zur Erkennung der Signaländerungen basiert auf der Differenz zwischen zeitlich versetzten Abtastpunkten. Für ein Signal s wird eine Änderung erkannt, wenn die Differenz über ein Zeitfenster einen Schwellwert überschreitet, wobei σ der Standardabweichung des Signals entspricht. Dieser Zusammenhang wird in der Gleichung 4.1 dargestellt:

$$|s(t + Fenster) - s(t)| > Schwellenwert \cdot \sigma_s \tag{4.1}$$

In Abbildung 4.7 ist erkennbar, dass diese Signaländerungen oder Übergänge nicht gleichzeitig bei allen Sensoren stattfinden. Es finden Verzögerungen aufgrund der Trägheit der Sensoren oder des elektrochromen Glases. Um weitere Erkenntnisse zu gewinnen, wird die Samplingrate der Sensoren auf 800 Hz erhöht. Bei einer Samplingrate von 800 Hz wird im Datenblatt bemerkt, dass eine ADC Integrationszeit von 411 µs nicht mehr möglich ist und deswegen auf 215 µs verringert werden muss.

Die Auswertung der Sensoren ergibt bei einer 400 Hz-Sensorsamplingrate Verzögerungen bei den Signalveränderungen bis zu 70 ms zwischen den Sensoren S1 bis S5 mit einer

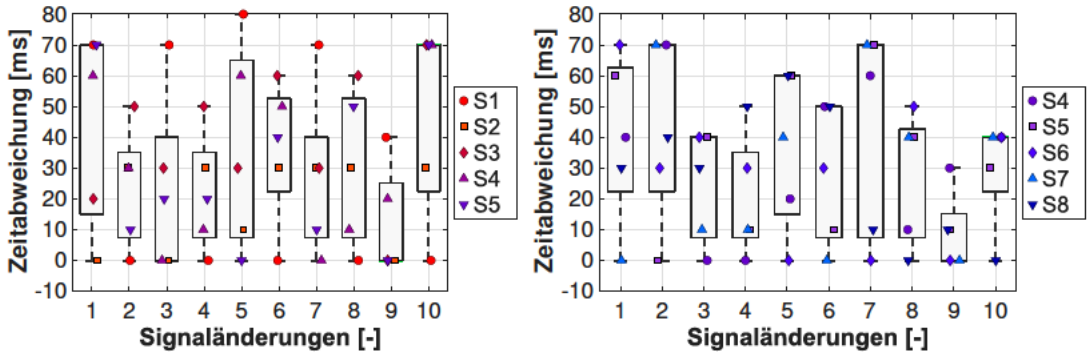
Standardabweichung von 2,003. Während die Sensoren S4 bis S8 Verzögerungen bis 60 ms mit einer Standardabweichung von 1,506 aufweisen. Bei den 800 Hz-Messungen liegen die Verzögerungen zwischen S1 und S5 bis 50 ms mit einer Standardabweichung von 1,101 und zwischen S4 bis S8 ebenfalls 50 ms mit einer Standardabweichung von 1,700. Somit haben die Verzögerungen bei einer 400 Hz-Samplingrate einen Mittelwert von 33,50 ms, während bei 800 Hz ein Mittelwert von 29,50 ms ermittelt wird. In Abbildung 4.8a und 4.8b werden die Verzögerungen dargestellt. Anhand von der Verteilung der Verzögerungen is erkennbar, dass die Ergebnisse unabhängig von der Sensorposition auf dem elektrochromen Glas. Daher liegt de Ursache der Verzögerungen beim Sensor. Die Samplingrate der Sensoren wird auf 400 Hz für die Probandenmessungen gelegt, da eine Erhöhung auf Kosten der ADC Integrationszeit geht. Durchschnittliche Verzögerungen zwischen den Sensoren von 30 ms bis maximal 80 ms sind im akzeptablen Bereich und werden in Abschnitt 3.8 zur Signalverarbeitung durch Glättungsverfahren ausgeglichen.



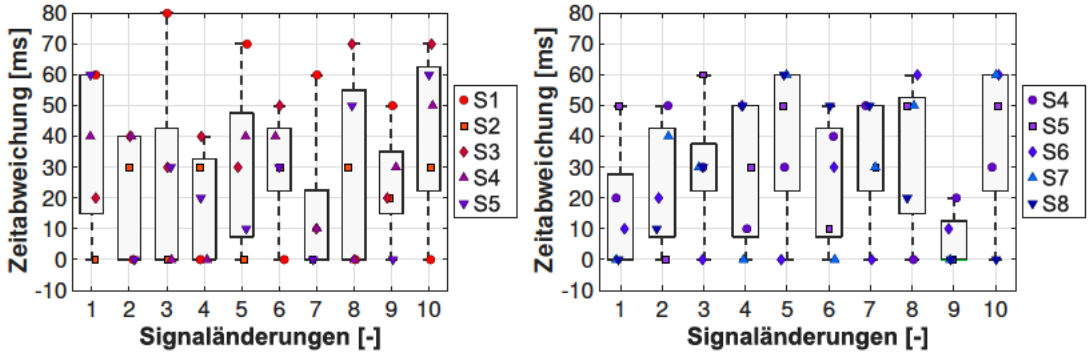
(a) Sensoren S1-S5

(b) Sensoren S4-S8

Abbildung 4.7: Synchronisationsanalyse der Sensordatenerfassung bei 400 Hz Samplingrate mit 0,5 Hz Rechteckwellen-Ansteuerung des elektrochromen Glases. Die geglätteten und normalisierten Signale zeigen (a) Sensoren S1 bis S5 und (b) Sensoren S4 bis S8. Vertikale grüne Linien markieren erkannte Signalübergänge, die leichte zeitliche Verzögerungen zwischen den Sensoren aufweisen.



(a) 400 Hz Samplingrate



(b) 800Hz Samplingrate

Abbildung 4.8: Statistische Analyse der Sensor-Synchronisationsverzögerungen bei (a) 400 Hz und (b) 800 Hz Samplingrate über 10 Signalübergänge. Boxplot-Darstellungen für Sensoren S1-S5 (links) und S4-S8 (rechts) zeigt die Zeitabweichungen. Die höhere Samplingrate reduziert die leicht Verzögerungen. Die Verteilung der Verzögerungen ist unabhängig von der Sensorposition.

4.6 Überprüfung der Sensorkonsistenz und Positionierung

Um das gleichmäßige Verhalten der Sensoren an den Muskelgruppen zu überprüfen, wird eine Rotation des Armbands bei verschiedenen Messungen durchgeführt. Der *Extensor carpi ulnaris* ist ein leicht zu identifizierender Muskel und wird als Referenzpunkt genommen. Sensor 3 wird an diese Stelle gelegt, sodass die restlichen Sensoren auf ihre Position fallen, wie in Abb. 3.2 dargestellt. Jede weitere Messung erfolgt nach einer Rotation des Armbandes um 45° gegen den Uhrzeigersinn, wodurch nach acht Messungen eine vollständige 360°-Rotation erreicht wird. Sensoren, die während verschiedener Rotationspositionen über derselben anatomischen Struktur positioniert sind, sollten ähnliche

Signalmuster aufweisen. Diese Charakteristik ist wichtig für die Robustheit des Systems gegenüber unterschiedlichen Positionierungen des Armbands am Unterarm. Pro Messung werden standardmäßig zwei Gesten innerhalb von 10 Sekunden durchgeführt. Bei einer Abtastrate von 100 Hz entspricht dies 1000 Messreihen pro Messung. Die Signale werden mittels der Median-Glättung von Rauschen befreit und anschließend normalisiert, um eine vergleichbare Darstellung zu gewährleisten.

In Abbildung 4.9 werden die Signalverläufe aller acht Sensoren über die gesamte Rotationssequenz dargestellt. Die Darstellung ist so strukturiert, dass jeder Sensor in einer separaten Zeile dargestellt wird und jedes 1000-Messpunkte-Segment (entsprechend 10 Sekunden Messung) einer Muskelgruppe zugeordnet ist. Die Segmente sind mit dem Namen des wahrscheinlich nächstliegenden Muskels gekennzeichnet. In Abbildung 4.9 sind geringe Abweichungen zwischen den Sensoren erkennbar. Die Sensoren zeigen deutliche Unterschiede zwischen den 2 ausgeführten Gesten. Bei manchen Messungen sind aber Ausreißer vorhanden, die Test aus praktischen Gründen entstehen.

Im Rahmen dieser Arbeit ist keine genaue Methode zur Sensorpositionierung beabsichtigt, weil die anatomischen Gegebenheiten der Unterarme zwischen Probanden stark variieren. Umfang, Muskelmasse und Hautbeschaffenheit unterscheiden sich erheblich. Zusätzlich ist selbst bei einem definierten Positionierungsprotokoll die exakte Reproduktion praktisch schwierig. Unterschiedliche Armumfänge, Hautspannung und die Flexibilität des Armbandes erschweren eine millimetergenaue Platzierung zwischen verschiedenen Messsitzungen.

Werden sowohl die roten als auch die infraroten Signale in Abb. 4.10 betrachtet, dann zeigen beide Kanäle niedrige Standardabweichungen. Die Werte liegen größtenteils zwischen von 0,05 und 0,20. Der Variabilitätsabfall bei den Gestenübergängen tritt in beiden Kanälen auf. Dies ist aber eher auf Bewegungsartefakte zurückzuführen, die bei den Messungen beobachtet werden können während eines Übergangs. Insgesamt zeigen beide Signaltypen ähnliche Abweichungen, wobei die Rot-Signale eine etwas höhere Stabilität aufweisen.

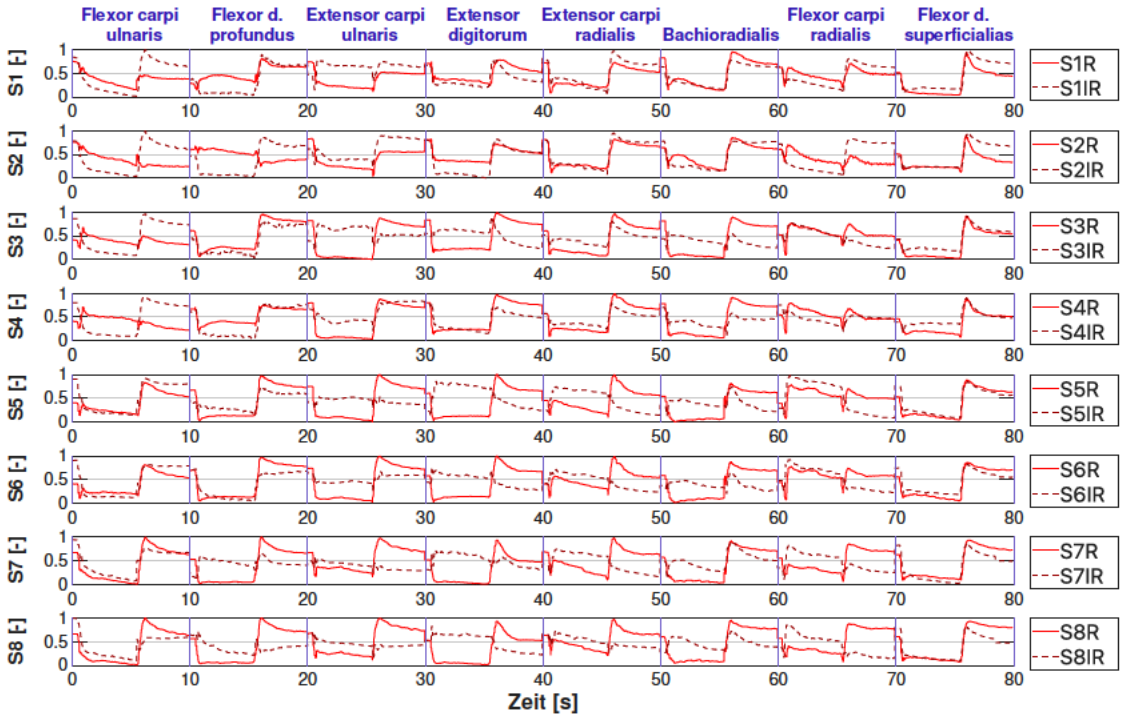


Abbildung 4.9: Systematische Rotationsmessung bei unterschiedlichen Armband-Positionierungen. Sensoren S1 bis S8 mit Roten- (durchgezogen Linien) und Infrarot-Signalen (gestrichelte Linien) werden über acht 45°-Rotationsschritte dargestellt. Jedes 10-Sekunden-Segment zeigt die Sensorreaktion auf zwei unterschiedlichen Gesten. Die anatomischen Bezeichnungen die jeweilige Muskelgruppen bei der Startposition sind oben angeben.

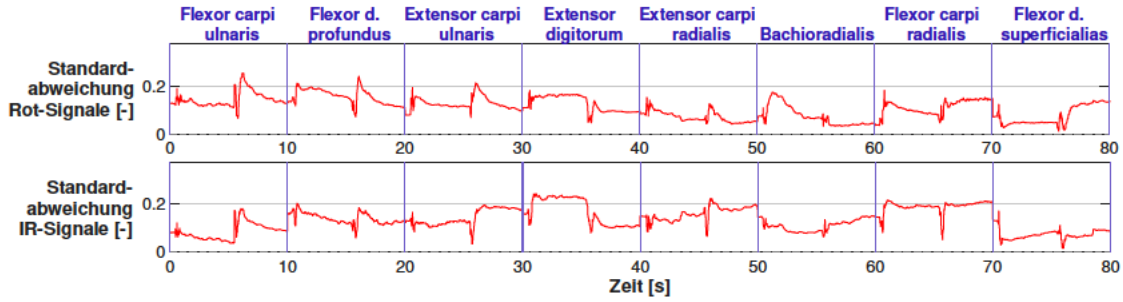


Abbildung 4.10: Standardabweichungsanalyse der Sensorsignale über alle acht Rotationspositionen. Der Vergleich der Standardweichung zwischen Rotsignalen (oben) und Infrarotsignalen (unten) zeigt größtenteils niedrige Standardabweichungen, wobei Variabilitätsanstiege bei Gestenübergängen erkennbar sind. Die Rot-Signale weisen eine geringfügig höhere Stabilität auf als die Infrarot-Signale.

5 Probandenmessungen

Das folgende Kapitel stellt die Ergebnisse der Probandenmessungen vor. Zunächst werden der Messaufbau und die Gesten beschrieben. Anschließend folgen die Trainingsergebnisse des entwickelten CNN-Modells sowie dessen praktische Anwendung in der Gestenerkennung. Abschließend wird die Generalisierungsfähigkeit des Systems durch probandenspezifische Tests vorgestellt.

5.1 Messsaufbau

Der Messaufbau ist so konzipiert, dass es für die Probanden verständlich ist, welche Gesten sie durchführen, während sie das Armband tragen. Dafür werden auf dem Bildschirm des Laptops Handgesten angezeigt, die sie nachmachen müssen. Die Gesten müssen verständlich dargestellt werden und zum richtigen Zeitpunkt durchgeführt werden können. Um die Probanden nicht zu ermüden, soll die Messung nicht länger als 15 Minuten dauern. Die Gesamtdauer der Messungen soll, inklusive Vorbesprechung und Armbandplatzierung, nicht länger als 30 Minuten dauern. Zunächst wird das Vorhaben mit den Probanden besprochen. Es werden das Ziel und das Vorgehen bei der Datenerhebung vermittelt und die Einwilligungserklärung unterschrieben. Selbstverständlich können die Probanden jederzeit die Teilnahme an dem Vorhaben ohne Begründung unterbrechen.

Für die Durchführung des Versuchs stehen acht Probanden zur Verfügung. Voraussetzungen für die Teilnahme sind, dass die Probanden zwischen 18 und 50 Jahre alt sind und keine relevanten Vorerkrankungen bekannt sind. Die erhobenen Daten werden unmittelbar nach der Messung anonymisiert. Die Messungen an Probanden sind von der Ethikkommission der HAW Hamburg bewilligt.

Die Messungen umfasst sechs Gesten. Probanden sollen insgesamt diese Gesten fünf Mal wiederholen, immer in Abwechslung zwischen zwei sich gegensätzlichen Paaren. Es entstehen dadurch drei Durchläufe, die zwei Mal wiederholt werden. Die Abwechslung

zwischen Gesten und Durchläufen soll helfen der Muskelermüdung entgegenzuwirken. Vor Beginn jedes Durchlaufs gibt es fünf Sekunden, die zu keiner Geste gehören. So können die Probanden mit ausreichender Zeit sich auf den Messstart vorbereiten. Die Messung beginnt, sobald das grüne Startsymbol aufleuchtet. Eine grafische Oberfläche erscheint während der gesamten Messung und schließt sich danach. Die Probanden haben über die grafische Oberfläche nicht nur ein Symbol, woran sie sich orientieren können, sondern auch ein Bild von der Geste, die sie durchführen sollen. Der exemplarische Aufbau ist in Abb. 5.1 dargestellt.

Die Messungen aller Durchläufe werden am Ende in einer CSV-Datei zusammengefasst. Durch die eingebaute Tagging-Funktionalität müssen diese Daten nicht mehr später gelabelt werden.

Es wird protokolliert, dass die richtige Geste zum richtigen Zeitpunkt ausgeführt wird. Das Protokoll enthält Angaben zur korrekten (OK) und nicht korrekten (NOK) Ausführung der Gesten, und ob Fehler dabei entstanden sind.

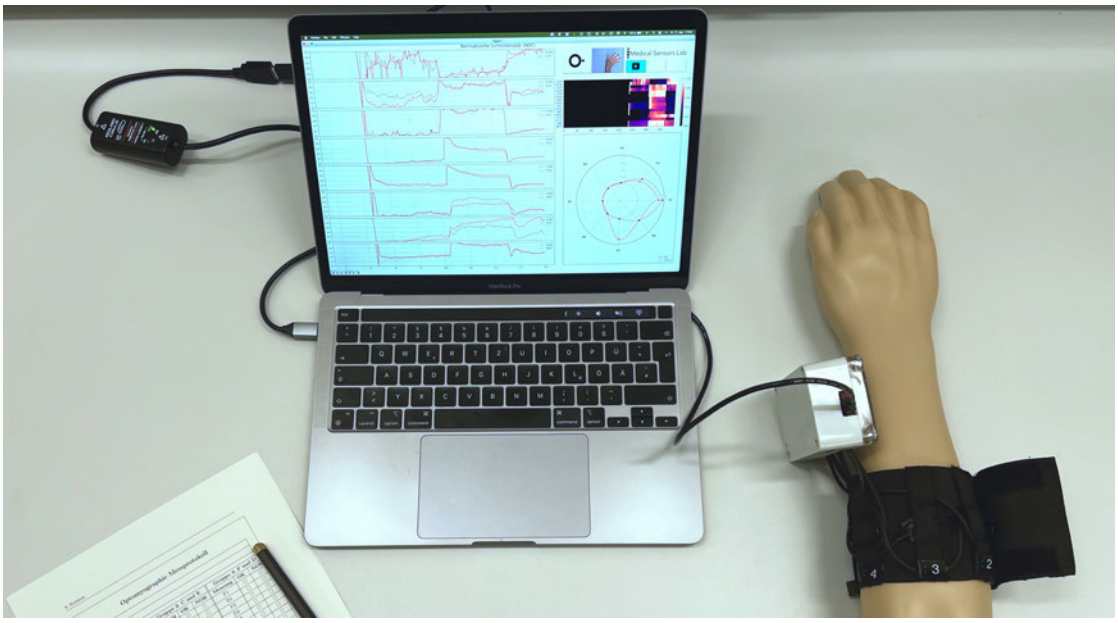


Abbildung 5.1: Versuchsaufbau aus Laptop, grafischer Oberfläche, Übertragungskanal sowie Sensorarmband.

Die Probandenmessungen enthalten eine Gesamtanzahl von 480 ausgeführten Gesten. Die Datenerfassung erfolgte im 10 ms Takt über alle 16 Signalkanäle der acht Sensoren. Die daraus resultierenden Matrizen für das CNN enthalten die Dimensionen 10x24 und

haben eine zeitliche Auflösung von 0,1 Sekunden. Daraus bilden die resultierenden Daten ein dreidimensionales Array der Form 19200x10x24, wobei 19200 die Anzahl der Matrizen repräsentiert.

Die resultierende Klassenverteilung beinhaltet eine ausgeglichene Verteilung der Daten, sodass alle Klassen gleich repräsentiert sind mit 3200 Matrizen. Für die Aufteilung in Trainings-, Validierungs- und Testdaten wird jede Datenmatrix einzeln einem zufälligen Wahrscheinlichkeitsverfahren unterzogen. Dabei werden mit einer Wahrscheinlichkeit von 50% die Daten dem Training, mit 25% der Validierung und mit 25% dem Test zugewiesen. Die Aufteilung dieser Matrizen ist somit zufällig, jedoch ausgeglichen.

5.1.1 Gesten

In diesem Abschnitt werden sechs grundlegende Handgelenksgesten vorgestellt und detailliert beschrieben, die als Vorgabe für die Klassifikationsmethode dieser Arbeit dienen.

Jede Geste wird durch eine charakteristische Bewegung definiert und in der oberen und seitlichen Perspektive visualisiert (siehe Abb. 5.2).

Geste A - Neutrale Position: Diese Geste repräsentiert die natürliche Ruhestellung des Handgelenks. Die Hand ist entspannt und flach ausgestreckt, die Finger sind leicht gespreizt und gestreckt. Das Handgelenk ist weder gebeugt noch gestreckt, sondern befindet sich in einer geraden Linie mit dem Unterarm. Diese Position dient als Referenzstellung für alle anderen Bewegungen.

Geste B - Faustschluss: Bei dieser Geste werden alle Finger eingerollt und zur Handinnenfläche hin gebeugt, bis sie die Handfläche berühren. Der Daumen kann über die Finger gelegt oder seitlich positioniert werden, wodurch eine geschlossene Faust entsteht. Diese Bewegung aktiviert primär die Flexormuskeln der Finger [43, 36].

Geste C - Dorsalflexion: Bei dieser Geste, auch als Extension bekannt, ist die Hand geöffnet und wird am Handgelenk nach oben (in Richtung Handrücken) gebeugt, sodass die Fingerspitzen zum Körper zeigen. Die Handfläche zeigt nach unten. Es entsteht ein charakteristischer Winkel zwischen Hand und Unterarm, wobei der Handrücken die höchste Position einnimmt [43, 36].

Geste D - Ulnarabduktion: Bei dieser Geste, auch als Adduktion bekannt, ist die Hand geöffnet und wird am Handgelenk seitlich in Richtung kleinen Finger gebeugt.

Die Handfläche zeigt nach unten. Der kleine Finger bildet den höchsten Punkt dieser Bewegung, während die Hand eine leichte C-Form in Richtung Kleinfingerseite bildet. Diese Bewegung ist das Gegenteil zur Radialabduktion [43, 36].

Geste E - Palmarflexion: Die Hand ist geöffnet und wird am Handgelenk nach unten (in Richtung Handfläche) gebeugt, sodass die Fingerspitzen vom Körper weg zeigen. Die Handfläche zeigt nach unten. Es entsteht ein Winkel zwischen Hand und Unterarm, wobei die Handfläche die tiefste Position einnimmt. Diese Bewegung ist das Gegenteil zur Dorsalflexion [43, 36].

Geste F - Radialabduktion: Bei dieser Geste, auch als Abduktion bekannt ist die Hand geöffnet und wird am Handgelenk seitlich in Richtung Daumen gebeugt. Die Handfläche zeigt nach unten. Der Daumen bildet den höchsten Punkt dieser Bewegung, während die Hand eine leichte C-Form in Richtung Daumengrundgelenk bildet [43, 36].

Es ist anzumerken, dass Bewegungen des Oberarms in dieser Arbeit bewusst nicht mit einbezogen wurden, obwohl diese Einfluss auf die Muskelaktivität im Unterarm haben.

Um eine möglichst gleichmäßige Ausführung der Gesten zu gewährleisten, werden die Probanden gebeten, den Ellenbogen um 90° anzuwinkeln während sie sitzen, sodass der Oberarm während des gesamten Versuchs in der gleichen Position verbleibt. Dabei wird darauf geachtet, dass das Armband keinen Kontakt mit der Armlehne oder sonstige Störungsfaktoren herstellt. Die Probanden werden angewiesen, die Bewegungen ohne besondere Anstrengung durchzuführen und jede Geste für fünf Sekunden zu halten, bis die Anweisung zur nächsten Geste erfolgt.

Die Auswahl der Gesten orientiert sich an den Arbeiten von [11], in denen Handgesten mittels EMG analysiert werden. Zusätzlich werden die Erkenntnisse aus [41] und [20] im Bereich der Optomyographie berücksichtigt. Alle diese Forschungsarbeiten sind im Kontext der Prothetik entstanden, was die praktische Relevanz der ausgewählten Gesten unterstützt. Das Hauptziel bei der Gestenauswahl bestand darin, den Datensatz bewusst klein zu halten, um die Klassifizierung zu vereinfachen, gleichzeitig aber die grundlegenden Bewegungsrichtungen des Handgelenks abzudecken.

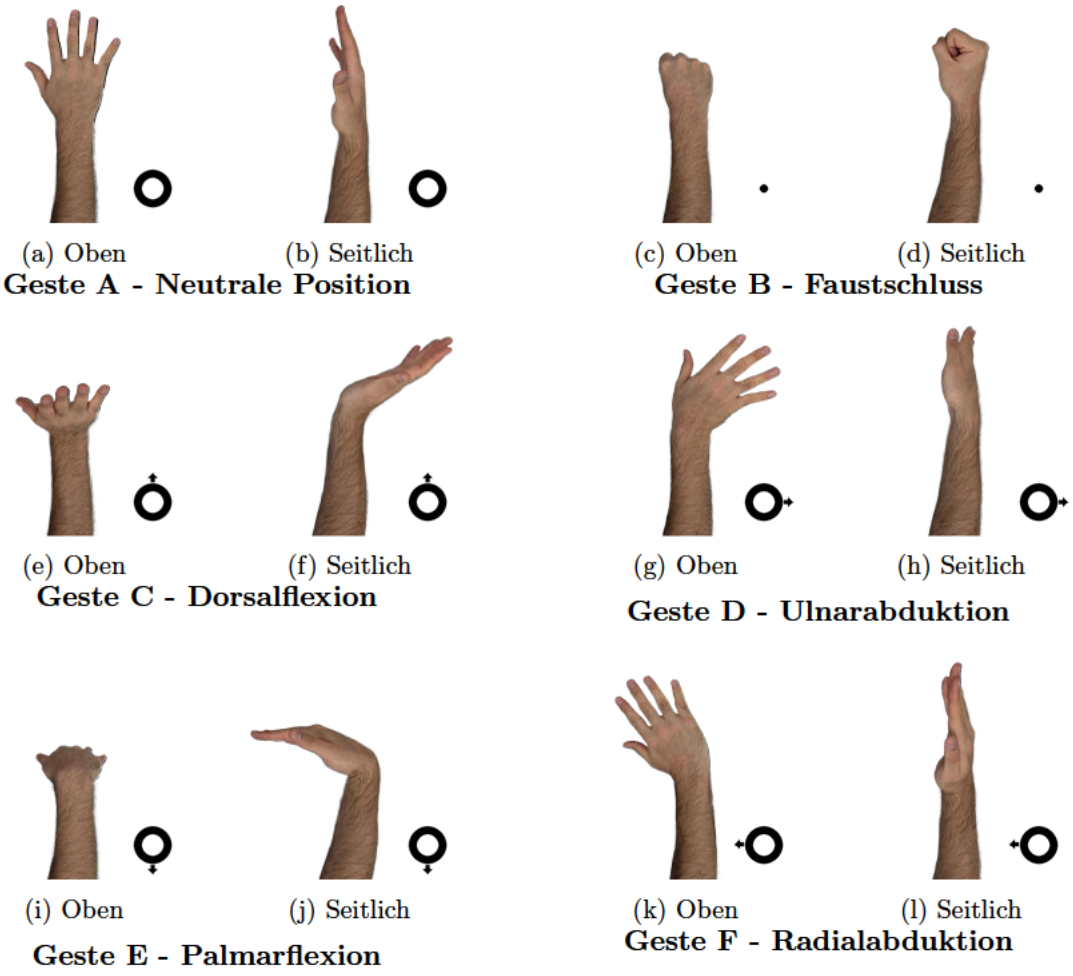


Abbildung 5.2: Übersicht aller sechs Handgelenksgesten mit entsprechenden Symbolen: (a) (b) Neutrale Position, (c) (d) Faustschluss, (e) (f) Dorsalflexion, (g) (h) Ulnarabduktion, (i) (j) Palmarflexion, (k) (l) Radialabduktion. Jede Geste wird von oben und seitlich dargestellt.

5.2 Charakteristische Merkmale der Datensätze

Bei den Probandenmessungen wurden alle 480 Gesten korrekt ausgeführt. Tabelle 5.1 fasst die statistischen Kennwerte aller 16 Sensorsignale vor der Normalisierung zusammen. Dabei werden sowohl minimale und maximale Signalwerte als auch ihre Standardabweichung für jeden Sensor dargestellt. Sensor S4R beispielsweise weist mit einer Spannweite von 7526 und einer Standardabweichung von 2328 für Maximalewerte und 1698 für Minimalwerte die höchste Abweichung auf. Die Infrarot-Werte erreichen höhere absolute

Signalwerte als ihre zugehörigen Rot-Werte, wobei S5IR mit einem Maximum von 20050 den höchsten gemessenen Wert aufweist und S3R das geringste Minimum mit einem Wert von 4371 erreicht.

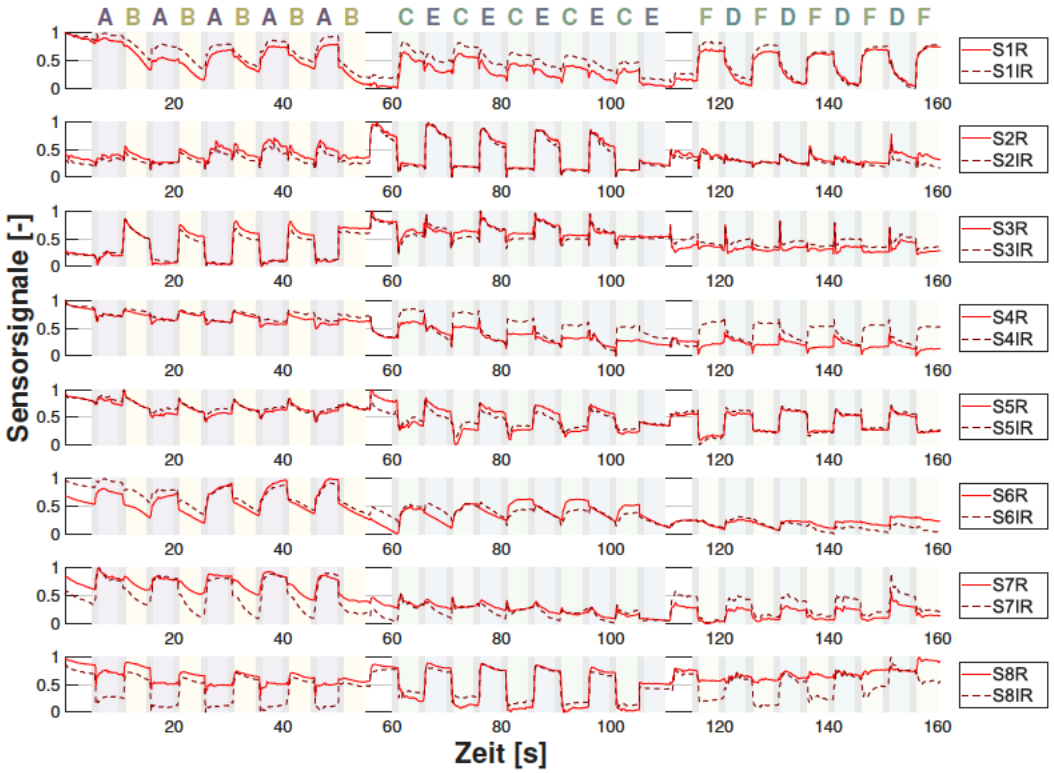
Tabelle 5.1: Messstatistiken aller Probandenmessungen.

Sensor	**Globales Minimum**	**Globales Maximum**	**Signal-spannweite**	**Std. Abweichung Maximalwerte**	**Std. Abweichung Minimalwerte**
S1R	5553	9398	3845	980	512
S1IR	8916	13570	4654	1089	1012
S2R	6191	13169	6978	1295	1192
S2IR	10023	14301	4278	1001	825
S3R	**4371**	10250	5879	1484	1400
S3IR	9548	15686	6138	1403	1428
S4R	6350	13876	**7526**	**2328**	**1698**
S4IR	10997	16618	5621	1450	1175
S5R	8581	14300	5719	1474	1102
S5IR	14430	**20050**	5620	1448	1224
S6R	7970	12885	4915	1289	867
S6IR	10656	15139	4483	525	988
S7R	5618	12137	6519	1635	1615
S7IR	10974	16298	5324	1200	1247
S8R	5119	10028	4909	1469	1316
S8IR	9578	13301	3723	907	746

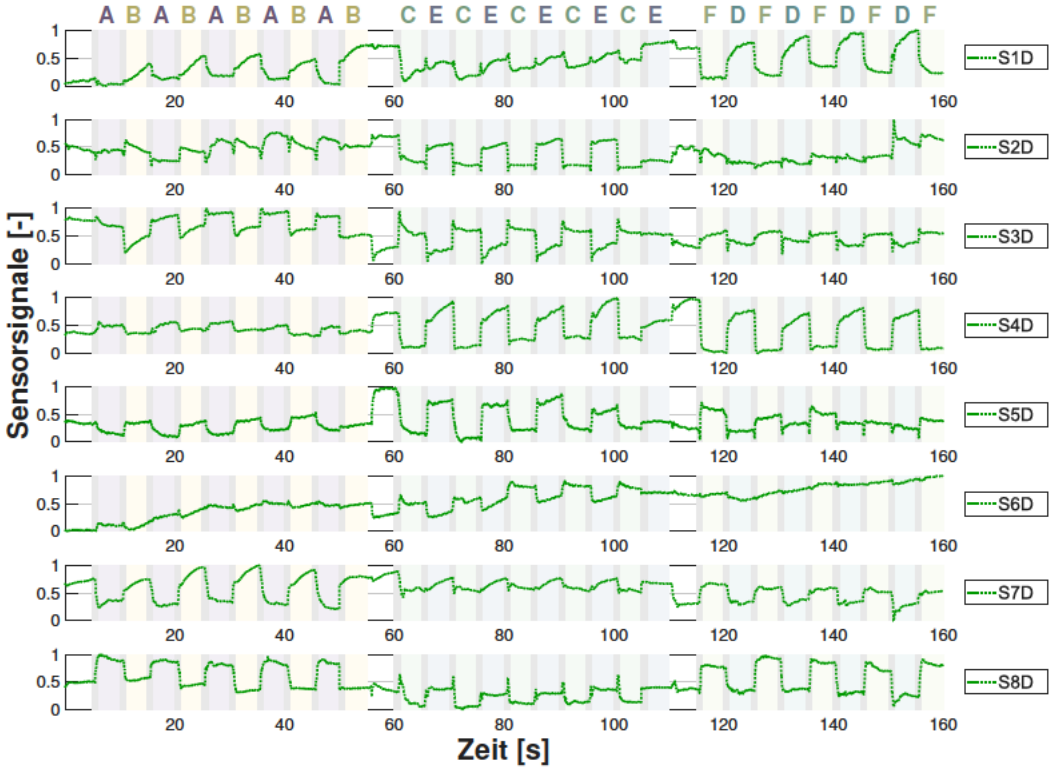
5.2.1 Zeitliche Entwicklung der Messwerte

Bei der zeitlichen Entwicklung der Signale sind die Veränderungen in den Übergangsphasen zwischen den Gesten am deutlichsten. Aufgrund des Aufbaus ensteht abwechselnde Muster, welche sich konsistent über die gesamte Messung zeigen. Diese sind durch stark steigende oder stark fallende Ausschläge erkennbar. In Abbildung 5.3a sind die Signalverläufe des Probanden 1 nach der Signalverarbeitung exemplarisch dargestellt. Die einzelnen Gestensegmente sind oberhalb des Plots markiert. Eine Lücke entsteht zu Beginn eines neuen Gestenpaares, wo keine spezifische Geste ausgeführt wird. Rot- und Infrarotsignale verlaufen meist parallel. Die Veränderungen zwischen den Gestenpaaren bilden ein Wellenmuster. In Abbildung 5.3b sind die Differenzwerte der Rot- und Infrarotsignale, die bei der Signalverarbeitung berechnet werden, dargestellt. In diesen Beispiel handelt es sich um die Differenzwerte des Probanden 1 aus der vorherigen Abb. 5.3a. Bis auf Sensor 6 (S6D) sind hier ebenfalls ähnliche Wellenmuster zu erkennen. Die Implikationen werden im Kapitel 6 erläutert.

Um die Komplexität der Gesten anhand einer dreidimensionalen Merkmalsverteilung zu präsentieren, werden 3D-Scatter-Plots in Abb. 5.4 dargestellt. Die X-Achse zeigt nor-

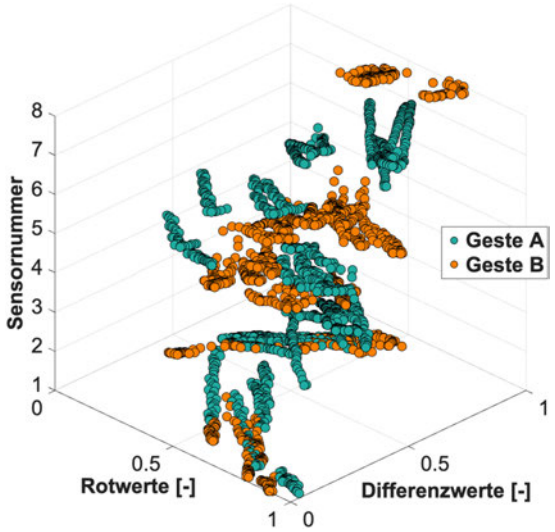


(a) Rot- und Infrarotsignale

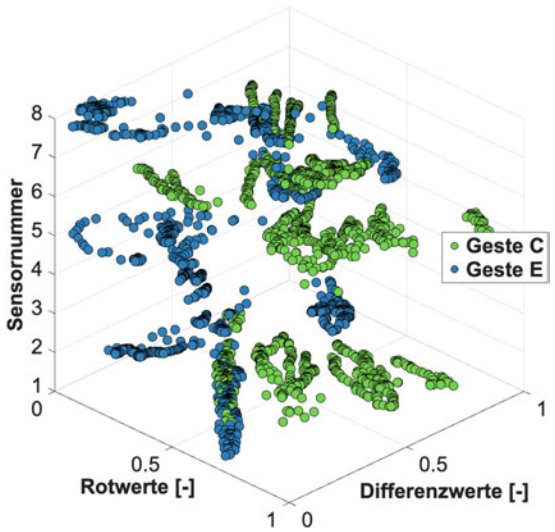


(b) Differenzsignale

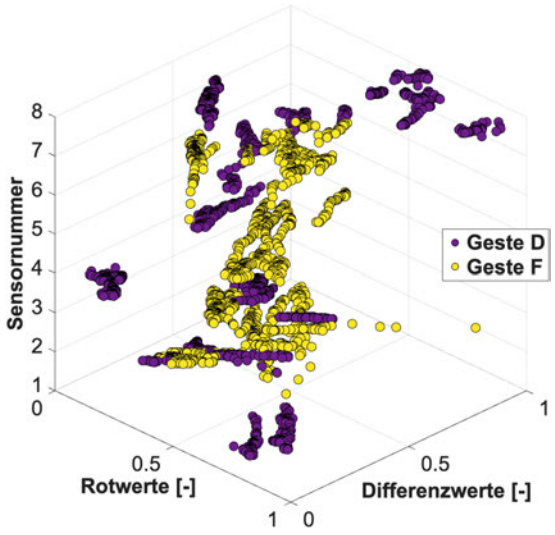
Abbildung 5.3: Zeitlicher Verlauf der Signale bei Proband 1. Gestenbezeichnungen sind oberhalb des Plots vermerkt. (a) zeigt die Rot-(durchgezogene rote Linien) und Infrarotsignale(gestrichelte dunkelrote Linien), (b) zeigt Differenzsignale (gepunktete grüne Linien).



(a) Scatter-Plot der Gesten A & B



(b) Scatter-Plot der Gesten C & E



(c) Scatter-Plot der Gesten D & F

Abbildung 5.4: Scatter-Plot der Sensordaten im 3D-Merkmalsraum - Beispiel anhand einer Probandenmessung. Gefärbt nach Gestentypen (A-F). Die X-Achse zeigt normalisierte Rotwerte, die Y-Achse Differenzwerte und die Z-Achse die Sensornummern (1-8). Die Visualisierung zeigt die räumliche Verteilung und Gruppierung der Messwerte.

malisierte Rotwerte, die Y-Achse die Differenzwerte und die Z-Achse die dazugehörigen Sensoren. Die Kombination mit den Differenzwerten auf der Y-Achse macht die entstehenden Muster besser erkennbar. Ebenso werden gegensätzliche Gesten als Paare gezeigt, sodass die Unterschiede deutlicher zu erkennen sind. In Abbildung 5.4a sind ähnliche Verteilungsmuster der Gesten A (orange) und B (türkis) dargestellt, die stark im 3D-Raum überlappen. Die orangen und türkisen Datenpunkte sind visuell schwer voneinander zu trennen, da beide Gesten komplexe und verzweigte Strukturen im Merkmalsraum bilden. Im Gegensatz dazu zeigen die Gesten C (grün)und E (blau) in Abb. 5.4b eine bessere Trennung, wobei sich Geste E auf bestimmte Rotwert-Bereiche konzentriert, während Geste C eine breitere Verteilung über den gesamten Merkmalsraum aufweist. Somit sind hierbei räumlich getrennte Cluster-Strukturen zu erkennen. In Abbildung 5.4c ist eine bessere Auseinanderhaltung zwischen den Gesten D (violett) und F(gelb) dargestellt. Geste D bildet einen kompakten isolierten Cluster, während Geste F eine zusammenhängende, bandförmige Struktur formt, wodurch sich beide Gesten minimal überlappen und gut zu unterscheiden sind.

5.2.2 Kreis-Visualisierung und Variabilität zwischen Messreihen

Pro ausgeführte Geste sind es 40 Matrizen in dem gesamten Datensatz vorhanden. Die Matrizen zeigen, wenn man sie stichprobenartig betrachtet, dass die Werte innerhalb einer einzelnen 100-ms-Matrix über die Zeit geringe Schwankungen aufweisen. Die Messwerte bleiben während dieser kurzen Zeitintervalle relativ konstant. Beim Vergleich verschiedener Sensoren zeigen sich hingegen deutliche Unterschiede. Die Messwerte variieren zwischen den einzelnen Sensoren stärker als innerhalb der zeitlichen Dimension einer Matrix. Anhand eines Beispiels von Proband 1 wird eine Matrix der Geste A in Abb. 5.5 in Form einer Heatmap visualisiert. Eine solche Darstellung lässt die räumliche Anordnung am Unterarm schwer erkennen. Zur Darstellung wird deswegen eine kreisförmige Visualisierungsmethode entwickelt. In dieser Visualisierung werden alle Sensoren entsprechend ihrer physischen Position am Armband im Kreis angeordnet, wobei der erste Sensor bei 90° positioniert ist und die weiteren Sensoren entgegen dem Uhrzeigersinn folgen. Der Radius der Kreisdarstellung repräsentiert die normalisierte Signalintensität. Die untere Grenze im Radius, und somit der Wert 0, liegt nicht in der Mittes des Kreises, sondern wurde bei einem Drittel des Radius gezogen. Diese untere Grenze soll helfen, die Kreisform zu wahren und somit die Aktivierungsmuster der Sensoren am Unterarm intuitiver erkennbar werden zu lassen. Alle Punkte einer Signalart sind miteinander ver-

bunden mittels Kurveninterpolation. In Abbildung 5.6 ist diese Visualisierungsmethode anhand eines Beispielsmatrix der Geste A in drei separate Kreisdarstellungen dargestellt. In Abbildung 5.6a sind die roten Signalkanäle als durchgezogene rote Linie, in Abb. 5.6b die infraroten Kanäle als gestrichelte dunkelrote Linie und in Abb. 5.6c die Differenz zwischen beiden Signalarten als gepunktete grüne Linie dargestellt. Die präsentierten Daten sind geglättet und normalisiert. In Abbildung 5.6 ist ersichtlich, dass Sensor 2 im *Ulnaris*-Bereich des Unterarms in gleichem Maße niedrige Ausschläge bei den Rot- und Infrarotwerten zeigt, während Sensor 6 im *Radialis*-Bereich das Gegenteil aufweist. Da sowohl Sensor 2 als auch Sensor 6 niedrige Unterschiede zwischen ihren jeweiligen Rot- und Infrarotwerten zeigen, sind in der Kreisdarstellung der Differenzwerte niedrige Ausschläge erkennbar.

Wenn die Werte mehrerer Probanden auf der Kreisdarstellung abbildet sind, so lassen sich die entstandenen Muster mit dem bloßen Auge schwer erkennen. Es ergeben sich aus dem Messungen auch deutliche Aussreisser. Um die entstandenen Muster aller gemessenen Gesten darzustellen, wird der Median aus den Werten ermittelt. So werden die Auswirkungen von Ausreißern verringert. Die Medianwerte der Rot- , Infrarot- und Differenzwerte aller 8 Probanden werden pro Geste in jeweils einer Kreisdarstellung in Abb. 5.7 zusammengefasst. Innerhalb des Kreises ist das Symbol der jeweiligen Geste dargestellt. Bei jeder Geste lässt sich eine unterschiedliche Form erkennen. Die Rot- und Infrarotwerte bilden ähnliche Muster, während die Differenzwerte eine andere Ebene und Form darstellen. Da es sich von Medianwerten mehrerer Messungen handelt, erreichen die Linien nicht die obersten und untersten Grenzen des Kreises, anders als im Beispiel in Abb. 5.6. Die Verhältnisse der entstandenen Formen sind trotzdem zu erkennen. Die Variabilität zwischen den Messungen wird in der Tabelle 5.2 dargestellt. Die Standardabweichung verschiedener Messungen liegt stets unter dem Wert 0,238.

Tabelle 5.2: Standardabweichungen der Messreihen als Maß für die Reproduzierbarkeit nach Gestenklassen

Geste	**A**	**B**	**C**	**D**	**E**	**F**
Rotwerte	0.172	0.238	0.172	0.171	0.210	0.192
Infrarotwerte	0.164	0.232	0.191	0.193	0.202	0.207
Differenzwerte	0.179	0.226	0.199	0.234	0.212	0.199

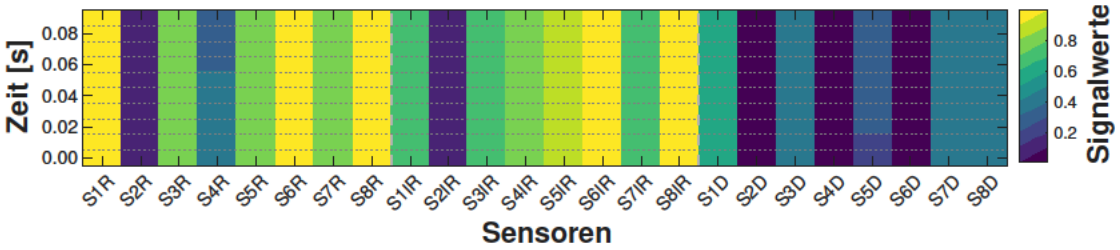
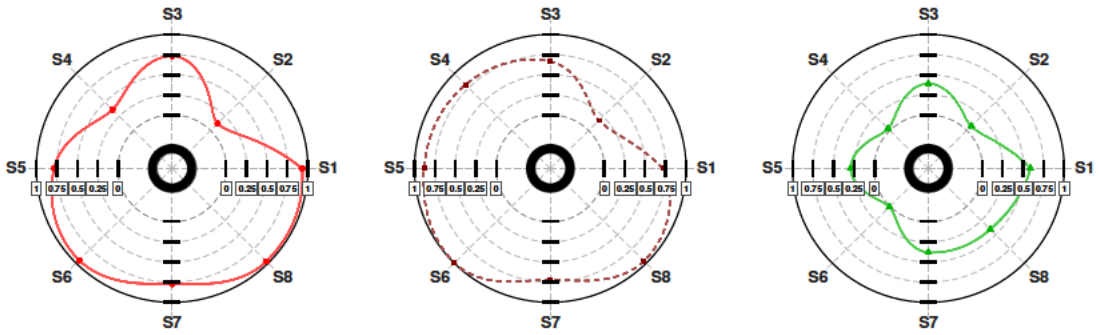
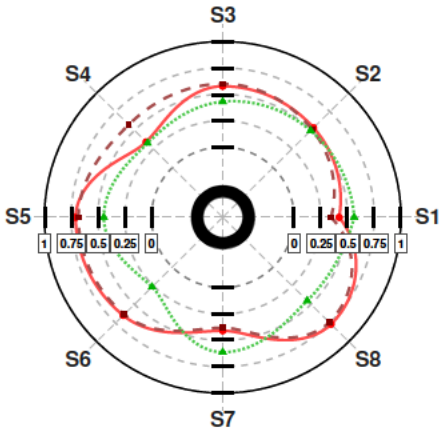


Abbildung 5.5: Heatmap der Signalverteilung über alle 24 Sensoren für ein exemplarisches Segment der Geste A. Die normalisierte Werte zwische 0 und 1 werden in eine Farbskala rechts angegeben. Die farblich dargestellte Messwerte verdeutlichen die charakteristischen Unterschiede zwischen den Sensoren, die zur Gestenerkennung genutzt werden.

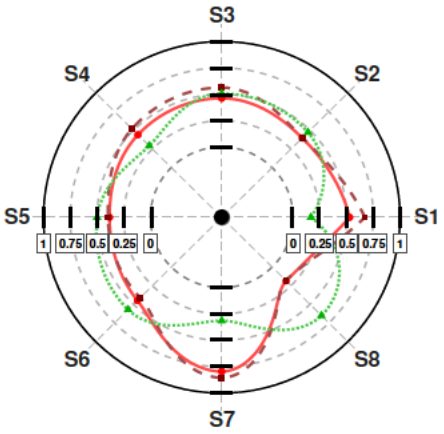


(a) Rotwerte (b) Infrarotwerte (c) Differenzwerte

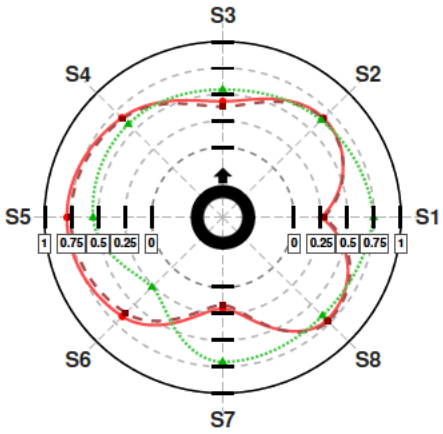
Abbildung 5.6: Kreisvisualisierung der Sensorsignale für Geste A. Die kreisförmige Anordnung entspricht der physischen Position der Sensoren am Armband, wobei der Radius die normalisierte Signalintensität repräsentiert. In der Mitte der Kreise befindet sich das Symbol der Geste A. Die untere Grenze 0 liegt beim ersten inneren Kreis. (a) zeigt die Rotwerte als eine durchgezogene rote Linie, (b) Infrarotwerte als eine gestrichelte dunkelrote Linie und (c) Differenzwerte als eine gepunktete grüne Linie.



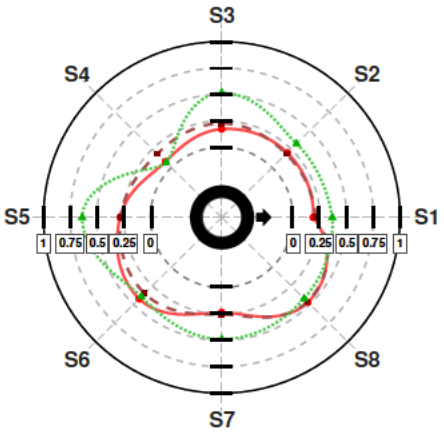
(a) Geste A



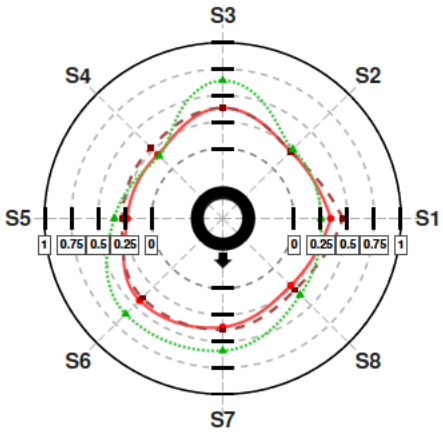
(b) Geste B



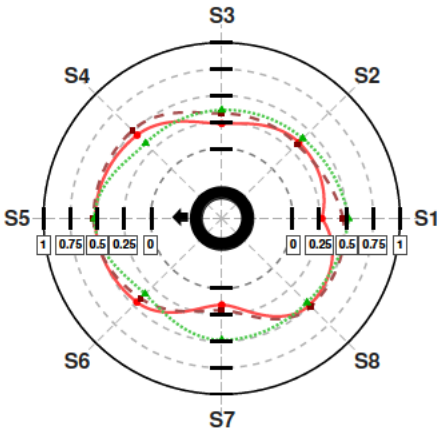
(c) Geste C



(d) Geste D



(e) Geste E



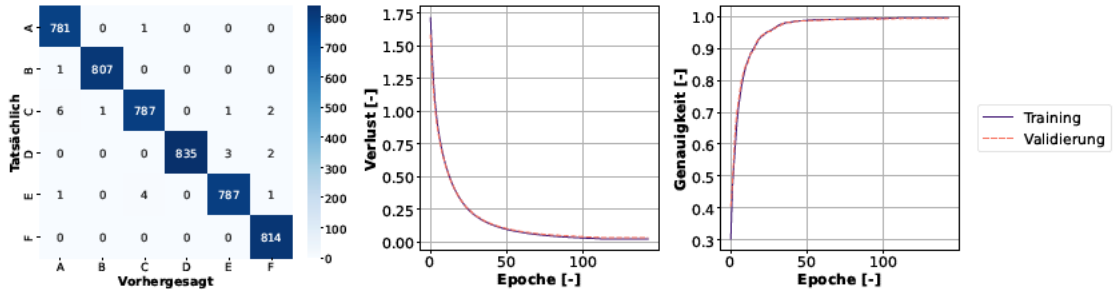
(f) Geste F

Abbildung 5.7: Kreisdarstellung aller sechs Gestenklassen mit gemittelten Signalwerten. In der Mitte der Kreise befinden sich die Symbole der jeweiligen Gesten. Durchgezogene rote Linien repräsentieren die Rot-Signalwerte, gestrichelte dunkelrote Linien die Infrarotwerte und gepunktete grüne Linien die Differenzwerte.

5.3 Trainings-Ergebnisse

Das Modell erreicht auf dem Testdatensatz eine Genauigkeit von 99,52% bei einem Testverlust von 0,0297. Die Modellarchitektur umfasst 17702 Parameter. Die Trainingsgenauigkeit beträgt 99,61% und die Validierungsgenauigkeit 99,44%. Die Konfusionsmatrix zeigt geringe Fehlklassifikationen im Testdatensatz. Die meisten Verwechslungen treten bei Klasse C auf, wobei sechs Instanzen fälschlicherweise als A, eine als B, eine als E und zwei als F klassifiziert wird. Bei Klasse E werden vier Instanzen als C und jeweils eine als A und F fehlklassifiziert. Klasse D weist fünf Fehlklassifikationen auf, davon drei als E und zwei als F. Die Klassen A und B zeigen jeweils eine Fehlklassifikation, wobei A einmal als C und B einmal als A klassifiziert wird. Klasse F erreicht eine vollständig korrekte Klassifikation ohne Fehlzuordnungen. In Abbildung 5.8a ist die Konfusionsmatrix dargestellt, welche die Klassifikationsergebnisse für alle sechs Gestenklassen zeigt.

Die Trainingskurven in Abb. 5.8b zeigen den Verlust und die Genauigkeit während der verschiedenen Epochen. Der Verlust fällt bereits nach wenigen Epochen gegen 0 exponetiell ab, während die Genauigkeit rasant gegen 100% strebt und exponentiell wächst. Der eingebaute Early-Stopping sorgte dafür, dass das Training vorzeitig bei Epoche 149 von 300 abgebrochen wurde.



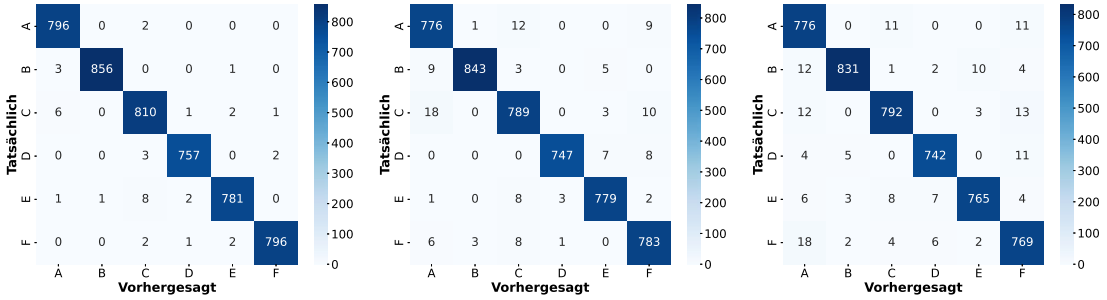
(a) Konfusionmatrix (b) Verlust und Genauigkeit der Trainingskurven

Abbildung 5.8: Metriken des CNNs. Links (a) Konfusionmatrix der Gestenklassifikation mit Darstellung der tatsächlichen und vorhergesagten Klassen. Rechts (b) Verlauf von Verlust und Genauigkeit des Training- und Validierungsdatensatzes in Abhängigkeit von der Epochenanzahl

Um die individuelle Eigenschaften der einzelnen Signalkomponenten zu untersuchen, wurden die Rot-, Infrarot- und Differenzsignale getrennt und für jede Signalart ein separates Training mit der selben Modell-Architektur durchgeführt. Wenn man das Training auf nur eine Signalart beschränkt, lassen sich bei reduzierter Parameteranzahl ebenfalls

hohe Genauigkeiten erreichen. Mit 7462 Parametern erzielten Testgenauigkeiten zwischen 96,71% und 99,21%. Das Rot-Signal erreicht dabei mit 99,21% eine vergleichbare Leistung zum kombinierten Training, während Infrarot-Signal und Differenz-Signal mit 97,58% bzw. 96,71% etwas geringere Genauigkeiten aufweisen. Die Konfusionsmatrizen der Einzelsignale in Abb. 5.9 zeigen unterschiedliche Fehlklassifikationsmuster zwischen den Signalarten.

Die Rot-Sensoren in Abb. 5.9a zeigten hauptsächlich Verwechslungen bei den Klassen C und E mit verschiedenen anderen Gesten. Die Infrarot-Sensoren in Abb. 5.9b wiesen verstärkte Fehlklassifikationen zwischen A und C sowie A und F auf. Die Differenzwerte in Abb. 5.9c zeigen ein breiteres Verwechslungsmuster mit Fehlklassifikationen zwischen A und C, A und F sowie B und verschiedenen anderen Klassen. Trotz der unterschiedlichen Fehlverteilungen erreichten alle drei Signalarten eine vergleichbar hohe Klassifikationsgenauigkeit.



(a) Rotsignale (b) Infrarotsignale (c) Differenzsignale

Abbildung 5.9: Konfusionmatrizen beim Traininig der Modelle mit getrennten Signalarten. Gegenüberstellung der vorhergesagten und tatsächlichen erkannten Gesten mittels (a) Rotsignale, (b) Infrarotsignale und (c) Differenzsignale.

5.4 Ergebnisse der umgesetzten Gestenerkennung

Um die Anwendung der Gestenerkennung zu demonstrieren, wird eine weitere Messung an Proband 1 durchgeführt, die nicht für das Training verwendet wurde. Im Gegensatz zu den Trainingsdaten werden bei dieser Messung in den Darstellungen keine Sekunden entfernt. Im Gegensatz zum Training werden beim Testen keine Übergangssekunden entfernt, um realistische Bedingungen einer kontinuierlichen Anwendung zu simulieren.

Zusätzlich bleiben die jeweils 5-sekündigen Leerphasen zu Beginn jeder Gestenpaarsequenz (ab Sekunden 0, 55 und 110) erhalten, in denen keine Geste ausgeführt wird. Diese Segmente sind daher nicht als fehlerhaft zu bewerten. In Abbildung 5.10 sind die Ergebnisse der Klassifikation über einen Zeitraum von 165 Sekunden dargestellt. Der obere Teil der Abb. 5.10 stellt die zeitliche Abfolge der erkannten Gesten dar, während der untere Plot die entsprechenden Konfidenzwerte der Klassifikation visualisiert. Die Ergebnisse zeigen, dass die festgelegte Gestensequenz korrekt erkannt wird und dem erwarteten Muster entspricht. Während der Übergangsphasen zwischen den Gesten treten jedoch Fehlklassifikationen auf, bei denen das Modell fälschlicherweise andere Gestenklassen erkennt. Die Konfidenzwerte zeigen durchgehend hohe Werte, jedoch ist ein Abfall in den Übergangsphasen auffällig.

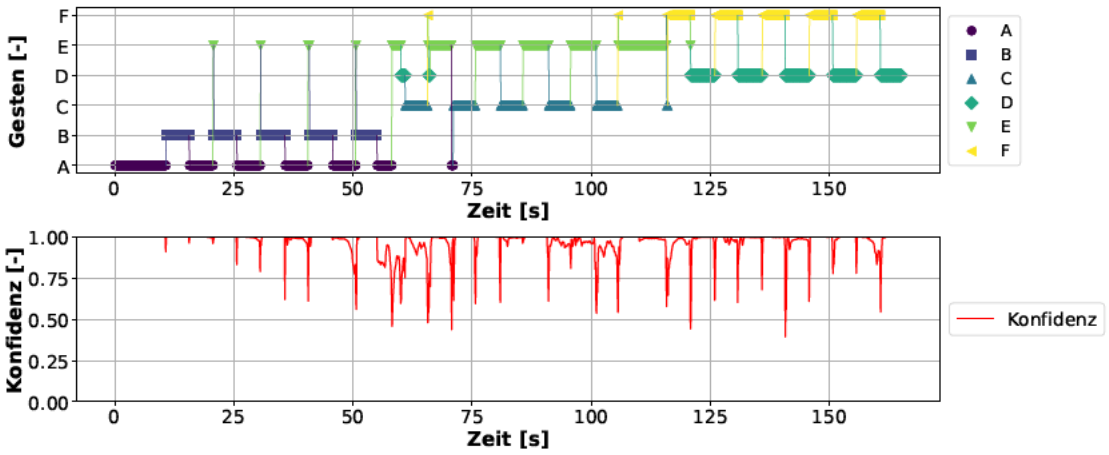


Abbildung 5.10: Ergebnisse der Gestenerkennung anhand eines Beispiels mit Darstellung der erkannten Gesten (oben) und zugehörigen Konfidenzwerten (unten). Die verschiedenen Gesten sind in dem oberen Plot in verschiedenen Formen und Farben in Abhängigkeit der Zeit dargestellt. Pro 100 ms wird eine Geste erkannt.

Um weitere Eigenschaften einer echten Anwendung zu simulieren, wird die vorgestellte Messung anders normalisiert. Bei der Anwendung ist es möglich, dass die Minima und Maxima der Signalwerte am Anfang noch nicht bekannt sind. Eine Sliding-Window-Normalisierung kann durchgeführt werden, um bei jedem neuen Messpunkt diese Werte zu ermitteln und zu aktualisieren. In Abbildung 5.11 ist die Umsetzung diese Art der Normalisierung dargestellt. Im Vergleich zu der Erkennung in Abb. 5.10 wird sichtbar, dass die Ergebnisse stark von der richtigen Umsetzung der Normalisierung abhängig sind.

Erst ab 110 s sind genügend Werte der einzelnen Gesten aufgetreten, sodass die Gesten wieder zuverlässig erkannt werden können.

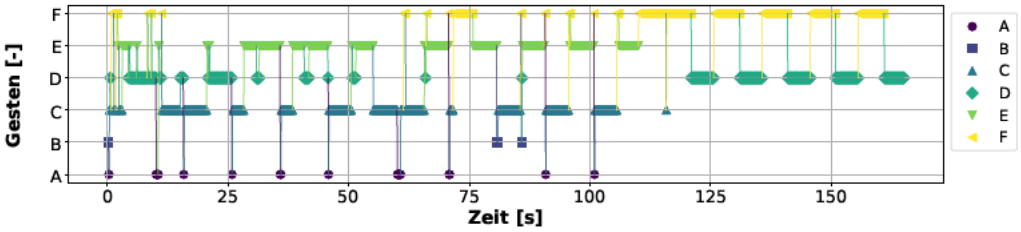


Abbildung 5.11: Ergebnisse der Gestenerkennung mit Sliding-Window-Normalisierung anhand eines Beispiels. Die verschiedenen Gesten sind in dem Plot in verschiedenen Formen und Farben in Abhängigkeit der Zeit dargestellt. Pro 100 ms wird eine Geste erkannt.

5.5 Probandenabhängige Testergebnisse der Modellperformance

Die Modellleistung wird durch verschiedene Aufteilung der Daten beim Training getestet. Dabei wird jede Messung der acht Probanden einmal als Test-Set verwendet und vom Training ausgeschlossen, während die Messwerte der verbleibenden sieben Probanden für Training und Validierung genutzt werden. Die Verteilung der Datenmatrizen bleibt unter Einhaltung dieser Vorgaben zufällig, wie beim zuvor vorgestellten Modell. Ähnlich wie bei der Kreuzvalidierung testet diese Methode die Generalisierungsfähigkeit des Modells auf unbekannte Probanden. Die Ergbenisse sind in Tabelle 5.3 dargestellt. Die Validierungsgenauigkeit zeigt bei allen acht Probanden konstant hohe Werte. Die Testgenauigkeit variiert hingegen deutlich zwischen den Probanden, mit Werten von 20,87 % beim Proband 6 (P6) bis 53,67 % beim Proband 3 (P3). Der Testverlust schwankt zwischen 2,1417 bei P3 und 7,2025 bei P8. Die Anzahl der benötigten Trainingsepochen liegt zwischen 95 bei P2 und 123 bei P7. P3 erreicht sowohl die höchste Testgenauigkeit als auch den niedrigsten Testverlust, während P6 die geringste Testgenauigkeit aufweist.

Im zweiten Testansatz werden die Trainingsdaten anders verteilt. Bei jedem Durchgang werden Messungen von fünf Probanden zum Trainieren, zwei anderen Probanden zur Validierung und einem Proband zum Testen verwendet. Diese Aufteilung wird achtmal durchgeführt, wobei jedes Mal andere Probanden in den Testdaten vorkommen. So wird

Tabelle 5.3: Die höchsten und niedrigsten Werte der Testdurchläufe sind fett markiert.

Test Proband [-]	Validierungs-genauigkeit [%]	Test-genauigkeit [%]	Test-verlust [-]	Epochen [-]
P1	99,05	40,58	4,3674	120
P2	99,35	43,58	4,6024	95
P3	99,32	**53,67**	**2,1417**	105
P4	99,40	46,29	3,6756	113
P5	99,32	31,75	3,7007	111
P6	99,40	**20,87**	5,9836	97
P7	99,49	28,21	2,9808	123
P8	99,61	29,96	**7,2025**	100

sichergestellt, dass das Modell mit verschiedenen Kombinationen getestet wird. In Tabelle 5.4 sind die Ergebnisse dargestellt. Die Validierungsgenauigkeit variiert zwischen 19,08% bei P1 als Testproband und 49,00% bei P3 als Testproband. Die Testgenauigkeit schwankt zwischen 1,92% bei P7 als Testproband und 61,21% bei P2 als Testproband. Die Anzahl der Trainingsepochen liegt zwischen 32 und 35, wobei die meisten Durchläufe 32 Epochen benötigen. Wenn P2 als Testproband verwendet wird, erreicht das Modell die höchste Testgenauigkeit, während P7 als Testproband die niedrigste Testgenauigkeit erzielt. Die Validierungsgenauigkeit zeigt bei P3 als Testproband den höchsten Wert und bei P1 als Testproband den niedrigsten Wert.

Tabelle 5.4: Evaluationsergebnisse der systematischen Datenaufteilung nach Probanden. Die höchsten und niedrigsten Werte der Testdurchläufe sind fett markiert.

Test Proband [-]	Training Probanden [-]	Validierung Probanden [-]	Validierungs-genauigkeit [%]	Test-genauigkeit [%]	Epochen [-]
P1	P2,P3,P4,P5,P6	P7,P8	**19,08**	39,08	32
P2	P3,P4,P5,P6,P7	P8,P1	35,02	**61,21**	32
P3	P4,P5,P6,P7,P8	P1,P2	**49,00**	33,58	32
P4	P5,P6,P7,P8,P1	P2,P3	46,52	32,25	35
P5	P6,P7,P8,P1,P2	P3,P4	40,98	34,25	34
P6	P7,P8,P1,P2,P3	P4,P5	35,46	39,08	34
P7	P8,P1,P2,P3,P4	P5,P6	36,71	**1,92**	32
P8	P1,P2,P3,P4,P5	P6,P7	24,77	30,46	32

6 Diskussion

In diesem Kapitel werden die erzielten Ergebnisse des entwickelten Gestenerkennungssystems diskutiert und ausgewertet. Zunächst erfolgt eine Analyse der Design- und Implementierungsentscheidungen sowie der charakteristischen Merkmale der erfassten Datensätze. Anschließend wird die Performance des CNN-Klassifikationsmodells einschließlich seiner Einschränkungen untersucht, bevor die Erfüllung der definierten Anforderungen überprüft und dabei die zentralen Ergebnisse der Arbeit zusammengefasst werden. Das Kapitel schließt mit einem Ausblick auf mögliche Weiterentwicklungen des Systems.

6.1 Analyse der Design- und Implementierungsentscheidungen

In diesem Abschnitt werden die getroffenen Design- und Implementierungsentscheidungen sowohl für die mechanische Konstruktion als auch für die Software- und Hardware-Architektur kritisch bewertet. Die Konstruktion des Armbandes ist insgesamt über drei Iterationen ausreichend eigenständig entwickelt worden. Viele praktische Probleme werden erst nach der Erstellung der Prototypen deutlich. Iteration 1 ist zu rigide, während Iteration 2 zwar flexibel genug ist, aber bei der Verwendung vermehrt Bruchstellen an den Lötstellen der Sensoren aufweist. Die ungewöhnliche Zuweisung der Pins in Iteration 3, die auf dem Schaltungsdiagramm zu sehen ist, entsteht durch die Verzweigung der Kabel, um die Lötstellen etwas stabiler gegenüber Bruchstellen zu machen. Die Integration der Wago-Klemmen erleichtert die Arbeit bei der Stromverteilung. Der Kasten, der die Komponenten umschließt, bietet trotz seiner Unhandlichkeit einen zusätzlichen Schutz bei Bewegungen.

Im Rahmen der Systemcharakterisierung stellt sich heraus, dass S2 fehlerhafte Temperaturwerte und mehr Rauschen als die restlichen Sensoren aufweist. Deswegen wird die Entscheidung getroffen, diesen Sensor auszutauschen. Ebenfalls müssen für vertauschte

Rot- und Infrarot-Messwerte Fehlerkorrekturen bei drei Sensoren implementiert werden. Die Feststellung, dass manche Sensoren einen Produktionsfehler haben und aus diesem Grund Rot- und Infrarot-LEDs falsch eingebaut sind, war schwer über die Datenanalyse festzustellen. Die sensorindividuellen Unterschiede der Rot- und Infrarot-Messwerte verhindern eine eindeutige Fehlerdefinition, da die variierenden Sensorbereiche keine einheitlichen Sollwerte für Infrarotsignale ermöglichen. Das Ausschalten der Infrarot-LEDs bestätigt erfolgreich die These des Produktionsfehlers. Hier wird die Entscheidung getroffen, die Fehler in der Firmware zu beseitigen, da die Messwerte der Sensoren trotzdem schlüssige Werte zeigen.

Entsteht ein Fehler bei der I2C-Kommunikation oder weist der Sensor interne Probleme auf, so setzt die Firmware den Rückgabewert -1 in den Ausgabestring. Während der Probandenmessungen treten keine kritischen Fehler auf. Messwerte werden vollständig und innerhalb des gegebenen Zeitraums übertragen.

Die softwareseitige Implementierung einer MVC-ähnlichen Architektur erweist sich als geeignete Designentscheidung. Sowohl bei der Messung als auch bei der Auswertung funktioniert die Software im Rahmen des erwarteten Bereiches. Multithreading zur gleichzeitigen Datenaufnahme und Darstellung der grafischen Oberfläche zeigt ein Problem mit der `animate()`-Funktion von `matplotlib`. Die Funktion ist in der Lage, ausschließlich auf dem Hauptthread zu laufen. Somit muss die Software entsprechend angepasst werden. Bei der Integration des Modells in der `SensorData`-Klasse wird im Rahmen dieser Arbeit noch kein Multithreading verwendet. Die Erkennung mit dem Modell läuft sequentiell nach der Erfassung einer vollständigen 10x24-Matrix der Daten alle 100 ms.

Bei der Fehlerbehandlung der Software müssen verschiedene Optionen gegeneinander abgewogen werden. Bei etwa 0,01% aller Sensorabfragen werden Messwerte in dem falschen Bereich der zugehörigen Wellenlängen des Sensors erfasst. Solange diese Art von Fehlern sporadisch auftauchen, ist das Verwenden von den zuletzt gespeicherten Daten bei der Fehlerbehandlung unproblematisch. Werden mehrere Fehler hintereinander registriert, so ist die Messung abzubrechen.

Die Entscheidung für eine getrennte Ausführung von Software und Firmware mit einer REPL-basierten Kommunikation erweist sich als vorteilhaft und leicht umsetzbar. In der initialen Entwicklungsphase, bevor die Software verfügbar ist, ermöglicht die direkte Parametereinstellung die Reaktionen der Sensoren über den Terminal zu testen. Nach Entwicklung der Software dient die `MockReceiver`-Klasse dazu, um das Verhalten des Systems zu analysieren.

6.2 Diskussion der charakteristischen Merkmale der Datensätze

Die Sensorsignale zeigen, dass die Infrarot-Werte meistens höhere absolute Signalwerte als die zugehörigen Rot-Werte erreichen. Die Sensoren zeigen untereinander erwartungsgemäß sehr unterschiedliche Minimal- und Maximalwerte sowie verschiedene Spannweiten (siehe Abschnitt 5.2). Da in dieser Arbeit eine Kalibrierung der Sensoren nicht beabsichtigt ist, werden die Unterschiede der Sensoren durch die durchgeführte Normalisierung verarbeitet. Die gemessenen Maxima und Minima sind von der jeweiligen Position des Sensors am Unterarm abhängig.

Die Analyse der beobachteten Wellenmuster zeigt, dass kurz nach der Ausführung einer Geste das Signal langsam abklingt oder zunimmt. Diese sinkenden und steigenden Tendenzen können auf Muskelermüdung zurückgeführt werden. Die ausgeübte Muskelkraft kann nicht gehalten werden und nimmt langsam ab. Diese Eigenschaft könnte aber auch eine Charakteristik des Sensors darstellen, bei der sich die Lichtreflexion über die Zeit anpasst. Hierbei handelt es sich um eine reproduzierbare Eigenschaft des Sensors (siehe Abschnitt 5.2.1). Ebenfalls anzumerken ist, dass der Hoch- und Tiefpunkt bei den meisten Wiederholungen der selben Geste ebenfalls tendenziell zu- und abnimmt. Zusätzlich ist bei der Betrachtung der Übergänge auffällig, dass hohe Ausschläge bei jedem Übergang entstehen, die nicht zum restlichen Muster der Gesten passen. Es ist anzunehmen, dass dies durch die Übergangsbewegung oder als Reaktion des Sensors auf Veränderungen erzeugt wird.

Der Unterschied zwischen den Rot- und Infrarotsignalen zeigt, dass die Infrarotsignale nicht immer höhere Werte als den Rotsignalen aufweisen. Obwohl Infrarotsignale in der Charakterisierung durchgängig höhere Werte als Rotsignale aufweisen, verlaufen die Signale nicht vollständig parallel. Diese Tatsache deutet darauf hin, dass entweder die beiden Wellenlängen unterschiedliche Aspekte, oder verschiedene Gewebetiefen und unterschiedliche Eigenschaften der Haut erfassen. Die Auswertung der Differenzwerte ist daher relevant, da in Betracht des gleich ausgeübten Drucks auf die roten und infraroten Dioden Artefakte ausgeschlossen werden können. Dies kann ein Indiz sein, dass das System mehr als lediglich mechanische Veränderungen am Sensor erfasst. Die Differenzwerte zeigen eine neue Ebene der Daten, die teilweise durch die Normalisierung verloren gegangen wäre. Bei der Merkmalsdarstellung im 3D-Scatter-Plot (siehe Abb. 5.4) wird die Komplexität besonders deutlich. Die Punkte zwischen den Gesten sind so verteilt,

dass sie komplexe Muster bei der Gruppierung aufweisen. Bei einigen Gesten bilden sich Gruppierungen. Die benachbarten Gruppen ähnlen sich. Da in dieser Darstellung Gesten fünfmal wiederholt werden, können diese Gruppierungen als Varianten der selben Geste identifiziert werden. Zudem ist es zu ergänzen, dass es aufgrund des menschlichen Faktors zu einer Abweichung bei der Gestenausführung kommen kann, was einen Einfluss auf diese Muster hat.

Die Untersuchung der Kreisdarstellungen in Abb. 5.7 zeigt, dass bei jeder Geste verschiedene Muster entstehen (siehe Abschnitt 5.2.2). Wenn angenommen wird, dass Geste A als Basisgeste festgelegt wird, wo alle Muskeln entspannt sind, so können die restlichen Gesten im Vergleich dazu betrachtet werden. Die erwartete Korrelation zwischen den theoretischen Muskelaktivierungen aus [43, 36] und den Sensorausschlägen in der vorliegenden Arbeit lässt sich nicht bestätigen. Bei isotonischen Muskelkontraktionen wird erwartet, dass sich der Muskel unter dem Sensor vergrößert und dadurch erhöhte Sensorwerte entstehen. Eine theoretische Muskelanspannung führt nicht zu den vorhersagbar erhöhten Sensorwerten. Als Beispiel kann die Kreisdarstellung der Geste B (Faustschluss) betrachtet werden, die starke Kontraktionen der Flexormuskeln verursacht und somit Ausschläge in S1, S7 und S8 erwarten lässt. Jedoch zeigen nur die Sensoren S1 und S7 höhere Ausschläge, aber S8 sinkt in die entgegengesetzte Richtung. Wird angenommen, dass nur Unterschiede der Werte aussagekräftig sind, so kann troztdem bei der Betrachtung der Geste C (Dorsalflexion) gesehen werden, dass sie nicht den Erwartungen entspricht. Bei Geste C sollen die Extensormuskeln stark kontrahieren, was sich in erhöhten Signalen bei den Sensoren S2, S3, S4 und S5 zeigen müsste. Jedoch zeigt sich, dass die Unterschiede zwischen Geste A und Geste C bei den angegebenen Sensoren nicht groß sind. Nach [39] können Bewegungen des Handgelenks die Sensorpositionierung erschweren, da sie die räumliche Beziehung zwischen Muskeln und Haut verändern. Pronation und Supination können erhebliche Veränderungen in der Muskelanordnung verursachen. Ähnliche Effekte sind bei Dorsalflexion, Palmarflexion und anderen Handgelenksbewegungen zu erwarten. Diese Betrachtung belegt jedoch nicht, dass die Messungen falsch sind, sondern dass sich die Muskulatur bei isometrischen Kontraktionen unter andere Sensorbereiche verschiebt. Dadurch weicht die erwartete Muskel-Sensor-Zuordnung von der Ausgangsposition ab.

Zusätzlich ist noch zu berücksichtigen, dass es sich in den Kreisdarstellungen in Abb. 5.7 um Medianwerte aller Probanden handelt. Die Standardabweichung der Messreihen kann positiv ausgewertet werden. Trotz Aussreisser zeigen sowohl Rot- als auch Infrarotsignale eine Standardabweichung von ca. 20%, die auch mit den beschriebenen Abweichungen im Abschnitt 4.6 vergleichbar ist.

6.3 Performance-Analyse des Klassifikationsmodells

Die Konfusionsmatrix des Modells zeigt gute Ergebnisse, was darauf deutet, dass das Modell die erkennbaren Muster der Gesten gut erschließen kann (siehe Abschnitt 5.3). Verschiebungen bei den Min- und Max-Werten haben große Auswirkungen auf die Effizienz des Modells. Das Problem könnte beseitigt werden, wenn feste Min- und Max-Werte für die Sensoren bestimmt werden. Ebenfalls ist jede Signalart (Rot, Infrarot, Differenz) in der Lage, allein genügend Merkmale zu bieten, um ein eigenständiges Modell herzuleiten. Auch bei den Differenzwerten ist erkennbar, dass die Veränderungen im Unterarm ausreichend sind, um unterschiedliche Muster zu bilden.

Das Modell weist jedoch Probleme auf, da es ungewöhnlich schnell zu 100% konvergiert. Die Performance steigt verdächtig schnell an und sowohl Training- als auch Validierungsmetriken fallen unrealistisch gut aus. Es gibt ebenfalls kaum eine Lücke zwischen Training- und Validierungsperformance. Grund dafür liegt am wahrscheinlichsten in der Aufteilung der Gestensegmente. Die Daten erhalten nicht genügend Variabilität zwischen den Segmenten, wenn sie in 100 ms aufgeteilt sind. Dadurch gelangen ähnliche Daten ins Training und in die Validierung. Das führt dazu, dass das Modell nicht so gut generalisieren kann. Data Leakage ist ein bekanntes Problem bei medizinischen Anwendungen, wie die Publikation [33] belegt. Es kommt dabei vor, dass Aufnahmen vom selben Patient sowohl im Trainings- als auch im Testdatensatz vorkommen. Die kann zu unrealistisch hohen Leistungswerten führen. CNNs können patientenspezifische Merkmale statt generalisierbare medizinische Muster lernen. Als Lösungen wird in [50] vorgeschlagen, dass Messungen desselben Probanden nicht in den Training- und Testdatensatz gelangen.

Während der Testphase wurde festgestellt, dass das Modell nicht mehr in der Lage war, die meisten Gesten richtig zu klassifizieren, wenn diese nicht beim Training verwendet wurden. Manche Probandenmessungen beinhalten zu starke Variationen zwischen den Gesten und wenn sie vom Training entfernt werden, so ist das Modell nicht mehr in der Lage, diese Gesten zu identifizieren.

Die Ergebnisse in Abschnitt 5.4 zeigen eine erfolgreiche Gestenerkennung unter optimalen Bedingungen. Jedoch offenbart die probandenabhängige Analyse in Abschnitt 5.5 deutliche Einschränkungen des Modells. Proband 3 liefert die zuverlässigsten Klassifizierungsergebnisse, wenn er aus dem Trainingsdatensatz ausgeschlossen wird. Die übrigen Probanden zeigen hingegen deutlich schwächere Ergebnisse bei der Validierung mit einer Testgenauigkeiten unter 50 %.

Um diese Beobachtungen zu untermauern, wurde in Abschnitt 5.5 ergänzend ein systematischer Test vorgestellt, bei dem jeweils drei Probanden aus den Trainingsdaten entfernt wurden. Die Ergebnisse zeigen, dass die Erkennungsleistung bei dem Test bei vielen Probanden deutlich abnimmt. Diese Feststellung deutet auf Overfitting hin.

Diese Problematik ist jedoch primär auf die begrenzte Probandenanzahl von acht Teilnehmern zurückzuführen. Bei einer derart kleinen Stichprobe haben individuelle Unterschiede einen überproportional großen Einfluss auf die Modellperformance. Eine Erhöhung der Probandenanzahl würde dem Modell ermöglichen, robustere Muster zu erlernen und eine bessere Generalisierung zu erreichen. Zusätzlich würden mehr Messungen pro Geste und eine optimierte Datenaufteilung die Modellstabilität verbessern.

Das Modell beantwortet erfolgreich die grundlegende Forschungsfrage, ob Gestenerkennung unter dem gegebenen System realisierbar ist. Die Ergebnisse zeigen deutlich das Potenzial des Ansatzes. Die nächsten Entwicklungsschritte liegen in der Skalierung auf eine größere und diversere Probandenbasis sowie in der Optimierung der Modellarchitektur für verbesserte Generalisierung. Mit diesen Anpassungen lässt sich eine robuste und zuverlässige Gestenerkennung für unbekannte Nutzer erreichen.

Letztlich zeigen die Ergebnisse Fehlklassifikationen in den Übergangsbereichen zwischen den Gesten. Da diese Bereiche gezielt aus den Trainingsdaten entfernt wurden (siehe Abschnitt 5.3) ist dies zu erwarten. Die Tatsache, dass der fehlerhafte Bereich kleiner als ursprünglich erwartet ausfällt, deutet auf eine positive Eigenschaft des Modells in untrainierten Situationen hin. Die beobachteten Konfidenzeinbrüche bei Übergängen spiegeln die erwartete Unsicherheit des Systems in diesen kritischen Bereichen. Die durchgeführten Tests offenbaren eine geringe Toleranz bei der Armbandpositionierung, was ein Hinderniss für die praktische Anwendbarkeit ist. Die einmalige Durchführung aller Gesten erweist sich als eine effektive Methode zur Systeminitialisierung, jedoch enstehen Probleme, sollten unerwartete Minimal- und Maximalwerte auftreten. Eine Abhängigkeit der Erkennungsleistung von der Gestenintensität wurde ebenfalls beobachtet. Beispielsweise bei der Faustbildung wurde die Geste manchmal nur erkannt, wenn die Faust stark genug geschlossen wurde. Die erreichten Inferenzzeiten von 15 ms bis 30 ms bei einer Datenerfassung von 100 ms unterstützen die Realisierbarkeit des Systems.

6.4 Erfüllung der Anforderungen

In diesem Abschnitt werden die definierten Anforderungen auf ihre Erfüllung überprüft und dabei die zentralen Ergebnisse der Bachelorarbeit zusammengefasst. Zunächst werden die **funktionalen Anforderungen** aus Abschnitt 3.1 betrachtet:

R1: Das System misst Aktivität am Unterarm nach dem Reflexionsprinzip der OMG erfolgreich. Die Auswirkungen der Muskulatur auf die Haut werden mittels rot- und infrarotem Licht erfasst. Es bilden sich ähnliche Muster jeweils zu den verschieden Gesten, mit einer Abweichung von ca. 20%. Dies ist am besten in der Kreisdarstellungen in Abb. 5.7 ersichtlich. Die Differenzen zwischen Rot- und Infrarotsignalen zeigen ebenfalls ähnliche Muster bei den jeweiligen Gesten.

R2: Die Anzahl der Sensoren deckt einen für die Messung relevanten Bereich des Unterarms gut ab. Die Umsetzung mittels `SoftwareI2C` lässt die acht Sensoren erfolgreich integrieren. Die Erhöhung der Sensorenanzahl kann durch eine höhere Abdeckung des Unterarms und somit einen erweiterten Messbereich zu besseren Ergebnissen führen. Jedoch würde diese Maßnahme auch die Reaktionsverzögerungen zwischen den Sensoren steigern.

R3: Die für den experimentellen Teil festgelegten Gesten sind ausreichend vielfältig gewählt, um solide Ergebnisse zu liefern. Sie beinhalten grundlegende Bewegungsrichtungen und wirken sich auf die Mehrheit der Muskeln im Unterarm aus. Eine Aussage, ob bei der Erweiterung der Gesten, wie zum Beispiel Pronation und Supination, ebenfalls positive Ergebnisse zu erwarten wären, lässt sich nicht treffen.

R4: Die Latenzzeit bei der Live-Gestenerkennung setzt sich aus drei verschiedenen Faktoren zusammen. Die Wartezeit, um eine vollständige Matrix zu bilden (100 ms), die Hälfte des Zeitfensters der Median-Glättung (60 ms) und die Inferenzzeit bei der Klassifizierung (ca. 30 ms). Sie bilden zusammen eine Latenzzeit von ca. 200 ms und liegen somit im akzeptablen Bereich unter 300 ms.

R5: Die grafische Oberfläche wird erfolgreich mit 100 Hz aktualisiert, aber durch das 60-Hz-Display begrenzt. Zusätzlich können durch die Rechenzeit weitere Verzögerungen entstehen. Probanden waren in der Lage, die Bewegungen bei den Messungen zu verfolgen, und der Tester konnte die Ergebnisse auf Plausibilität prüfen. Wegen der Modularität der Architektur der grafischen Oberfläche ließ sich diese für die Darstellung der Erkennungsergebnisse während einer Gestenerkennung erweitern.

Bezüglich der **Qualitätsanforderungen** aus Abschnitt 3.1 konnten folgende Beobachtungen gemacht werden:

R6: Das System wurde mit dem MAX30102 PPG-Sensor und dem Raspberry Pi Pico erfolgreich umgesetzt. Fehler des MAX30102 ließen sich größtenteils ausgleichen und die Ergebnisse sind davon nur minimal beeinflusst. Die Sensoren wurden mit einem ADC-Bereich von 16384 nA, einer ADC-Integrationszeit von 411 µs, einer LED-Helligkeit von 25,4 mA und einer Samplingrate von 400 Hz konfiguriert. Der Raspberry Pi Pico ließ sich problemlos einsetzen. Sollte eine höhere Abtastrate angestrebt werden, so gerät der Raspberry Pi Pico an die Geschwindigkeitsgrenzen unter Verwendung von MicroPython.

R7: Die Übertragung der Messdaten per USB mit 115200 Baud wurde erfolgreich umgesetzt, wobei die Eigenschaften des REPL einbezogen werden müssen. Entstehen Blockaden beim Ausgeben in der Konsole, so hat das eine Auswirkung auf die Datenübertragung. Die Durchschnittslänge eines übertragenen Strings ist ungefähr 100 Zeichen. Bei der Baudrate können ungefähr 115 Zeichen pro 100 Hz übertragen werden, wodurch keine Übertragungsengpässe entstehen. Sollten Veränderungen in der Stringlänge vorgenommen werden, so müssen Verzögerungen bei der Übertragung berücksichtigt werden. Der USB-Isolator hatte keine Auswirkung auf die Messungen.

R8: Die Sensoren erreichen nach den Messungen maximale Temperaturen von 35°C und bleiben damit unter dem Grenzwert von 40°C. Der durchschnittliche Temperaturanstieg bei der 20-Minuten-Messung beträgt 0,73°C.

R9: Das System entfernt erfolgreich Rauschen und Ausreißer durch Median-Glättung mit einem symmetrischen Zeitfenster von 130 ms. Die Min-Max-Normalisierung wurde erfolgreich implementiert und erfordert bei der Live-Erkennung eine Kalibrierungsphase, in der alle notwendigen Referenzwerte erfasst werden. Dadurch werden optimierte Eingangsdaten für die CNN-basierte Gestenerkennung bereitgestellt.

R10: Die Gestenerkennung erreicht eine Klassifikationsgenauigkeit von 99,56% bei den Probandenmessungen und liegt damit über der geforderten Mindestleistung von 95%, jedoch mit Einschränkungen bezüglich der Position des Armbandes und Ungenauigkeiten bei der Normalisierung der Daten. Die Erkennung ist lediglich auf das Set der Probanden beschränkt und nicht auf Probanden außerhalb des Trainings generalisiert. Um die Erkennungsfähigkeit des Modells bei Anwendern ausserhalb der Probandengruppe zu steigern, ist eine Erweiterung der Probandenanzahl notwendig.

R11: Das CNN-Klassifikationsmodell nutzt erfolgreich Matrizen mit einem Zeitfenster von 100 ms für die Gestenerkennung und unterschreitet damit die geforderten 200 ms. Die Eingabedaten werden als 10×24-Matrizen strukturiert, die aus 3 verschiedenen Signalarten von acht Sensoren bei einer Abtastrate von 100 Hz generiert werden. Der resultierende Datensatz mit 19200 Matrizen aller acht Probanden basiert auf 480 ausgeführten Gesten aller Probanden.

R12: Die Synchronisationsmessungen ergeben sensorspezifische Verzögerungen von durchschnittlich 33,5 ms mit maximalen Werten bis 80 ms bei 400 Hz Samplingrate. Eine Erhöhung auf 800 Hz reduziert die Verzögerungen auf durchschnittlich 29,5 ms mit maximalen Werten bis 50 ms. Dies erfolgt jedoch auf Kosten der ADC-Integrationszeit, die von 411 µs auf 205 µs reduziert wird, was die Messempfindlichkeit der Photodioden beeinträchtigt. Daher wurden 400 Hz als optimaler Kompromiss zwischen zeitlicher Synchronisation und Signalqualität gewählt, da die Verzögerungen unter der geforderten 100 ms Grenze liegen und durch Glättungsverfahren kompensiert werden können.

R13: Die Sensoren erfassen alle Signale erfolgreich innerhalb eines Zeitintervalls von 10 ms ohne Verzögerungen. Die Implementierung nutzt einen Software-Timer auf dem Raspberry Pi Pico, der zyklisch eine Callback-Funktion aufruft und über ein Flag die Datenerfassung freigibt und koordiniert. Durch die Verwendung des Flags wird ein Zeitintervall von 10 ms ohne komplexe Synchronisationsmechanismen erreicht.

6.5 Ausblick

Basierend auf den gewonnenen Erkenntnissen dieser Arbeit ergeben sich verschiedene Ansätze für die Weiterentwicklung des Systems.

Die Erhöhung der Sensoranzahl um fünf weitere Sensoren würde mehr Bereiche des Armes abdecken und ein präziseres Messbild schaffen. Ziel ist es, dass benachbarte Sensoren näher an denselben Bereich des Unterarmes messen. Die Messgenauigkeit lässt sich so durch gewichtete Kombination der Sensordaten steigern.

Um die Verzögerungen zwischen den Sensoren zu verringern, wäre es zielführend, die Auswirkungen einer Erhöhung der Samplingrate auf 1600 Hz zu testen. Zusätzlich könnte getestet werden, ob eine andere Laufzeitumgebung wie C oder C++ höhere Abtastraten des Systems erlauben würde.

Eine weitere Möglichkeit wäre der Umbau des entwickelten Messarmbands zu einem vollständig eingebetteten System. Dafür ist zusätzliche Hardware notwendig, die fähig ist, die Erkennung am Armband auszuführen und anzuzeigen. Das System müsste batteriebetrieben werden und genügend Rechenleistung besitzen. Dies würde den Anwendern mehr Möglichkeiten geben, sich freier zu bewegen, und das Training realistischer gestalten.

Die Position des Oberarmes wurde in dieser Arbeit statisch festgelegt. Jedoch könnten sich die Ergebnisse verändern, wenn man neue Positionen testet. Darüber hinaus wäre es zielführend, die Bewegungen des Oberarmes zu messen und im Vergleich zu denen des Unterarms zu stellen. Dadurch könnten sich neue Einblicke in die Auswirkungen des Oberarmes auf den Unterarm ergeben. Das Armband besitzt eine ausreichende Größe, um Messungen am Oberarm auszuführen.

Die Kombination des Systems mit EMG- oder EIM-Methoden wäre theoretisch möglich, solange die PPG-Sensoren nicht von den elektrischen Signalen gestört werden. Eine solche Umsetzung kann mehr Merkmale der Muskulatur gleichzeitig erfassen und somit umfassendere Datensätze bieten.

Schliesslich ist es nennenswert, dass die Versuchsdurchführung durch mehr Gesten erweitert werden kann, die beispielsweise auch Fingerbewegungen miteinbeziehen können. Eine größere Anzahl an Probanden könnte ebenfalls helfen, die Probleme mit Overfitting und Data Leakage zu beseitigen. Interessant wäre außerdem die Analyse der Übergänge zwischen den Klassen. Starke Anstiege oder Abfälle des Signals könnten dabei als Übergänge interpretiert und entsprechend behandelt werden.

Literaturverzeichnis

[1] Anand, R.: *Digital Signal Processing An Introduction*. Dulles, VA : Mercury Learning and Information, 2022. – 652 S. – URL https://doi.org/10.1515/9781683928010

[2] Anderson, R. R. ; Parrish, John A.: The optics of human skin. In: *Journal of Investigative Dermatology* 77 (1981), Nr. 1, S. 13–19. – ISSN 0022-202X

[3] Arce, Gonzalo R.: *Median and Weighted Median Smoothers*. Wiley, 2004. – 80–138 S. – ISBN 9780471691853

[4] Azami, Hamed ; Mohammadi, Karim ; Bozorgtabar, Behzad: An Improved Signal Segmentation Using Moving Average and Savitzky-Golay Filter. In: *Journal of Signal and Information Processing* 3 (2012), February, S. 39–44. – URL http://www.scirp.org/journal/jsip

[5] Bansal, A. ; Hou, S. ; Kulyk, O. ; Bowman, E. ; Samuel, I.: Wearable Organic Optoelectronic Sensors for Medicine. In: *Advanced Materials* 27 (2014), December

[6] Barlian, A A. u. a.: Review: Semiconductor Piezoresistance for Microsystems. In: *Proceedings of the IEEE* 97 (2009), Nr. 3, S. 513–552

[7] Bashkatov, A. N. ; Genina, E. A. ; Kochubey, V. I. ; Tuchin, V. V.: Optical properties of human skin, subcutaneous and mucous tissues in the wavelength range from 400 to 2000 nm. In: *J. Phys. D. Appl. Phys.* 38 (2005), Nr. 15, S. 2543–2555

[8] Beucher, Ottmar 1.: *Signale und Systeme: Theorie, Simulation, Anwendung Eine beispielorientierte Einführung mit MATLAB*. Berlin, Heidelberg : Springer-Verlag Berlin Heidelberg, 2011 (SpringerLink Bücher). – URL https://doi.org/10.1007/978-3-642-20294-0. – Includes bibliographical references and index

[9] Chaiken, Joseph ; Deng, Bin ; Goodisman, Jerry ; Shaheen, George ; Bussjager, Rebecca J.: Analyzing near-infrared scattering from human skin to monitor changes in hematocrit. In: *Journal of Biomedical Optics* 16 (2011), September, Nr. 9

[10] Chu, J. K. ; Moon, I. ; Mun, M.: A real-time EMG pattern recognition based on linear-nonlinear feature projection for multifunction myoelectric hand. In: *IEEE 9th International Conference on Rehabilitation Robotics (ICORR)*. Chicago, IL, 2005

[11] Daochai, S. ; Sueaseenak, D. ; Chanwimalueang, T. ; Laoopugsin, N. ; Pintavirooj, C.: *Independent component analysis: An application for muscular contraction classification*. 2007

[12] Elia, Nicola: *MAX30102 MicroPython driver*. https://github.com/n-elia/MAX30102-MicroPython-driver. 2023

[13] Englehart, K. ; Hudgins, B.: A robust, real-time control scheme for multifunction myoelectric control. In: *IEEE Transactions on Biomedical Engineering* 50 (2003), Nr. 7, S. 848–854

[14] Englehart, K. ; Hudgins, B. ; Chan, A.: Continuous multifunction myoelectric control using pattern recognition. In: *Technology and Disability* 15 (2003), S. 95–103

[15] Farrell, Todd R. ; Weir, Richard F.: The Optimal Controller Delay for Myoelectric Prostheses. In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 15 (2007), Nr. 1, S. 111–118

[16] Goshtasby, A. A.: *Preprocessing*. John Wiley & Sons, Ltd, 2004. – 7–41 S. – ISBN 9780471724278

[17] Graupe, D. ; Salahi, J. ; Zhang, D. S.: Stochastic analysis of myoelectric temporal signatures for multifunctional single-site activation of prostheses and orthoses. In: *Journal of Biomedical Engineering* 7 (1985), S. 18–29

[18] Géron, Aurélien: *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*. 3. O'Reilly Media, 2023. – 193–423 S. – ISBN 978-1098125975

[19] Hahne, Janne M. ; Markovic, Marko ; Pardo, Luis A. ; Kusche, Roman ; Ryschka, Martin ; Schilling, Arndt F.: On the Utility of Bioimpedance in the Context of Myoelectric Control. In: *IEEE Sensors Journal* 21 (2021), September, Nr. 17

[20] Hamid Muhammed, H. ; Raghavendra, J.: Optomyography (OMG): A Novel Technique for the Detection of Muscle Surface Displacement Using Photoelectric Sensors. In: *Measurements* 10 (2015), S. 13

[21] Hauschild, Sebastian ; Hellbrück, Horst ; Kusche, Roman: Optical Muscle Contraction Detection via Frequency Multiplexed LEDs. In: *IEEE Conference Proceedings* IEEE (Veranst.), IEEE, 2023, S. 1–5

[22] Hefftner, G. ; Zucchini, W. ; Jaros, G. G.: The electromyogram (EMG) as a control signal for functional neuromuscular stimulation—Part I: Autoregressive modeling as a means of EMG signature discrimination. In: *IEEE Transactions on Biomedical Engineering* 35 (1988), Nr. 4, S. 230–237

[23] Hudgins, Bernard ; Parker, P.A. ; Scott, R.N.: A New Strategy for Multifunction Myoelectric Control. In: *IEEE Transactions on Biomedical Engineering* 40 (1993), Nr. 1, S. 82–94

[24] Hüning, Felix 1.: *Sensoren und Sensorschnittstellen*. De Gruyter Oldenbourg, 2016 (De Gruyter eBook-Paket Technik, Informatik). – 185 S. – URL https://doi.org/10.1515/9783110438550

[25] IFTOOLS GmbH: *Zertifiziertes USB Isolator-Kabel ISOUSB-Cable-A: Technisches Datenblatt*. Matterhornstrasse 17, 65199 Wiesbaden, Germany: IFTOOLS GmbH (Veranst.), Mai 2023. – 2 S. – URL https://www.iftools.com/shop/index.de.php. – Zertifiziert nach EN 60601-1, 3. Edition; Hohe verstärkte Isolierung von 4kVrms

[26] Jacques, Steven: *Optical penetration depth of light into skin tissue*. June 2008. – URL https://omlc.org/news/jun08/penetration/index.html. – Zugriffsdatum: 2025-06-15

[27] Jacques, Steven L.: Optical properties of biological tissues: a review. In: *Physics in Medicine and Biology* 58 (2013), Nr. 11, S. R37–R61

[28] Khoei, Mina A. ; Galluppi, Francesco ; Sabatier, Quentin ; Pouget, Pierre ; Cottereau, Benoit R. ; Benosman, Ryad: Faster is better: Visual responses to motion are stronger for higher refresh rates. In: *bioRxiv* (2018). – URL https://doi.org/10.1101/505354. – Preprint posted December 29, 2018

[29] Kuroki, Yoshihiko ; Nishi, Tomohiro ; Kobayashi, Seiji ; Oyaizu, Hideki ; Yoshimura, Shinichi: A psychophysical study of improvements in motion-image quality by using high frame rates. In: *Journal of the Society for Information Display* 15 (2007), Nr. 1, S. 61–68

[30] Kusche, Roman: *Mehrkanal-Bioimpedanz-Instrumentierung zur zeitaufgelösten Messung physiologischer Ereignisse*. Lübeck, Universität zu Lübeck, Inauguraldissertation (Dr.-Ing.), 2019. – Aus dem Institut für Medizintechnik, Sektion Informatik/Technik

[31] Kusche, Roman ; Oltmann, Andra ; Rostalski, Philipp: A Wearable Dual-Channel Bioimpedance Spectrometer for Real-Time Muscle Contraction Detection. In: *IEEE Sensors Journal* (2023). – Member, IEEE

[32] Lee, Gae H. ; Kang, Hyunbum ; Chung, Jong W. ; Lee, Yeongjun ; Yoo, Hyunjun ; Jeong, Sujin ; Cho, Hyeon ; Kim, Joo-Young ; Kang, Sung-Gyu ; Jung, Ji Y. ; Hahm, Suk G. ; Lee, Jeahyuck ; Jeong, In-Jo ; Park, Minho ; Park, Gunkuk ; Yun, In H. ; Kim, Justin Y. ; Hong, Yongtaek ; Yun, Youngjun ; Kim, Sung-Han ; Choi, Byoung K.: Stretchable PPG sensor with light polarization for physical activity–permissible monitoring. In: *Science Advances* 8 (2022), Nr. 15

[33] Lee, Hyung-Tak ; Cheon, Hye-Ran ; Lee, Seung-Hwan ; Shim, Miseon ; Hwang, Han-Jeong: Risk of data leakage in estimating the diagnostic performance of a deep-learning-based computer-aided system for psychiatric disorders. In: *Scientific Reports* 13 (2023), Nr. 1. – URL https://doi.org/10.1038/s41598-023-43542-8. – ISSN 2045-2322

[34] Liang, Yongbo ; Elgendi, Mohamed ; Chen, Zhencheng ; Ward, Rabab: Analysis: An optimal filter for short photoplethysmogram signals. In: *Scientific Data* 5 (2018), May. – URL https://www.nature.com/articles/sdata201876

[35] Mann, Kenneth A. ; Werner, Frederick W. ; Palmer, Andrew K.: Frequency Spectrum Analysis of Wrist Motion for Activities of Daily Living. In: *Journal of Orthopaedic Research* 7 (1989), Nr. 2, S. 304–306

[36] Martini, Frederic H. ; Timmons, Michael J. ; Tallitsch, Robert B.: *Anatomie - Bafög-Ausgabe* . Pearson Deutschland, 2017. – 920 S. – URL https://elibrary.pearson.de/book/99.150005/9783863268282. – ISBN 9783868943399

[37] MathWorks: *normalize - Normalize matrix data*. https://de.mathworks.com/help/matlab/ref/double.normalize.html. n.d.. – Abgerufen am 01.04.2025

[38] Maxim Integrated: *MAX30102 High-Sensitivity Pulse Oximeter and Heart-Rate Sensor for Wearable Health*. – Datenblatt

[39] Miyake, Shota ; Miyake, Tamon: CNN-based Gripping-force Estimation Using Optical Muscle Deformation Sensors. In: *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems* ACM (Veranst.), 2025, S. 1–8

[40] Miček, Juraj ; Kapitulík, Ján: Median Filter. In: *Journal of Information, Control and Management Systems* 1 (2003), January, Nr. 2, S. 51–56

[41] Muhammed, H. H. ; Jammalamadaka, R.: A new approach for rehabilitation and upper-limb prosthesis control using optomyography (OMG). In: *IEEE International Conference on Biomedical Engineering*, 2016

[42] Nikita, Konstantina S. (Hrsg.): *Handbook of biomedical telemetry.* IEEE Press, 2014 (IEEE Press series in biomedical engineering [23]). – 79–451 S. – URL http://www.gbv.de/dms/tib-ub-hannover/775733091.pdf

[43] Phillips, Jacqueline P.: *Hands-on anatomy.* North Broad Press, 2024

[44] Rangayyan, R.M. ; Krishnan, S.: *Biomedical Signal Analysis.* Wiley, 2024 (IEEE Press Series on Biomedical Engineering). – URL https://books.google.de/books?id=PEmtzwEACAAJ. – ISBN 9781119825852

[45] Raspberry Pi (Trading) Ltd.: *Raspberry Pi Pico Datasheet.* 2020. – Build-date: 2021-01-21, Build-version: fcd04ef-clean, Licensed under Creative Commons Attribution-NoDerivatives 4.0 International (CC BY-ND)

[46] Rohnen, Armin: *MATLAB® meets MicroPython mit MATLAB® Mikrocontroller nutzen.* Springer Vieweg, 2022 (essentials). – 17–31 S. – URL https://doi.org/10.1007/978-3-658-39949-8

[47] Salditt, Tim ; Aspelmeier, Timo ; Aeffner, Sebastian: *Biomedical Imaging, Principles of Radiography, Tomography and Medical Physics.* Berlin, Boston : De Gruyter, 2017. – URL https://doi.org/10.1515/9783110426694. – ISBN 9783110426694

[48] Scheme, Erik ; Englehart, Kevin: Electromyogram pattern recognition for control of powered upper-limb prostheses: State of the art and challenges for clinical use. In: *Journal of rehabilitation research and development* 48 (2011), Nr. 6, S. 643

[49] soluna1: *Wrong activation of R and IR in MAX30102.* GitHub Issue. 5 2020. – URL https://github.com/sparkfun/SparkFun_MAX3010x_Sensor_Library/issues/25. – Zugriffsdatum: 2025-07-08. – Issue #25 in SparkFun_-MAX3010x_Sensor_Library repository

[50] Tampu, Iulian E. ; Eklund, Anders ; Haj-Hosseini, Neda: Inflation of test accuracy due to data leakage in deep learning-based classification of OCT images. In: *Scientific Data* 9 (2022), Nr. 1, S. 580. – URL https://doi.org/10.1038/s41597-022-01618-6. – ISSN 2052-4463

[51] Tanaka, Taichi ; Nambu, Isao ; Maruyama, Yoshiko ; Wada, Yasuhiro: Sliding-Window Normalization to Improve the Performance of Machine-Learning Models for Real-Time Motion Prediction Using Electromyography. In: *Sensors* 22 (2022), Nr. 13, S. 5005. – URL https://doi.org/10.3390/s22135005

[52] Tarata, Mihai T.: Mechanomyography versus Electromyography, in monitoring the muscular fatigue. In: *BioMedical Engineering OnLine* 2 (2003), February, Nr. 3. – URL http://www.biomedical-engineering-online.com/content/2/1/3

[53] Taroni, P. ; Pifferi, A. ; Torricelli, A. ; Comelli, D. ; Cubeddu, R.: In vivo absorption and scattering spectroscopy of biological tissues. In: *Photochemical and Photobiological Sciences* 2 (2003), Feb, Nr. 2, S. 124–129

[54] Technologies, Siglent: *SDG2000X Series Function/Arbitrary Waveform Generator Datasheet*. Siglent Technologies (Veranst.), 2025. – URL https://www.welectron.com/mediafiles/datasheets/siglent/Siglent_SDG2000X_Datasheet-Web.pdf. – Abgerufen am 18. Mai 2025

[55] Wagenaar, R. C. ; Emmerik, R. E. A. van: Resonant frequencies of arms and legs identify different walking patterns. In: *Journal of Biomechanics* 33 (2000), Nr. 7, S. 853–861

[56] WAGO Kontakttechnik GmbH & Co. KG: *Verbindungsklemme mit Hebeln - Datenblatt*. WAGO (Veranst.), 2025. – URL https://www.wago.com/de/installationsklemmen/verbindungsklemme-mit-hebeln/p/222-415. – Zugriffsdatum: 2025-08-09. – Art.-Nr. 222-415

[57] Welch, Ashley J. ; Gemert, Martin J. C. van: *Optical-Thermal Response of Laser-Irradiated Tissue*. 2nd. Springer, 2011

[58] Werner, Martin: *Digitale Signalverarbeitung mit MATLAB® Grundkurs mit 16 ausführlichen Versuchen*. 5., durchgesehene und aktualisierte Auflage. Wiesbaden : Vieweg+Teubner Verlag / Springer Fachmedien Wiesbaden GmbH, Wiesbaden, 2012 (Studium). – URL http://dx.doi.org/10.1007/978-3-8348-8621-7

[59] Zecca, Massimiliano ; Micera, Silvestro ; Carrozza, Maria C. ; Dario, Paolo: Control of Multifunctional Prosthetic Hands by Processing the Electromyographic Signal. In: *Critical Reviews in Biomedical Engineering* 30 (2002), Nr. 4-6, S. 459–485

[60] Ünsalan, Cem 1.: *Embedded System Design with ARM Cortex-M Microcontrollers Applications with C, C++ and MicroPython.* 1st ed. 2022. Springer International Publishing, 2022 (Springer eBook Collection). – 41–569 S

A Anhang

A.1 Verwendete Hilfsmittel

In der Tabelle A.1 sind die im Rahmen der Bearbeitung des Themas der Bachelorarbeit verwendeten Werkzeuge und Hilfsmittel aufgelistet.

Tabelle A.1: Verwendete Hilfsmittel und Werkzeuge

Tool	Verwendung
Claude Sonnet 4	Animation der Kreisdarstellung, Debugging, LaTeX-Formatierung, Rechtschreibprüfung, Syntaxprüfung, Erstellung von Vorlagen
draw.io	Erstellung von UML-Diagrammen und Abbildungen
GitLab	Datensicherung und Versionskontrolle
Google Colab	Training des Modells
HAW-Katalog	Recherche und Literaturbeschaffung
IEEE Xplore	Recherche wissenschaftlicher Literatur
LaTeX	Textsatz- und Layout-Werkzeug verwendet zur Erstellung dieses Dokuments
MATLAB	Datenverarbeitung und -analyse
PyCharm	IDE für die Software-Entwicklung
Staatsbibliothek Hamburg	Recherche und Literaturbeschaffung
TeXstudio	Bearbeitung der LaTeX-Dokumente
Thonny	IDE für die Firmware-Entwicklung

A.2 Ordnerstruktur des Anhangs

Die folgende Ordnerstruktur ist auf der beigelegten CD dieser Arbeit zu finden:

```
Anhang/
|-- Firmware
|-- Abbildungen
|   |-- Grundlagen
|   |-- Charakterisierung
|   |-- Gui
|   |-- Methodik
|   |   |-- Hardware
|   |   |-- Band
|   |   |-- Software
|   |-- Ergebnisse
|   |-- Gesten
|-- Software
|   |-- Images
|-- Messungen
|   |-- Charakterisierung
|   |   |-- LED-Einstellungen
|   |   |-- Synchronitaetsanalyse
|   |   |-- Positions-Rotation
|   |   |-- Sensor-Anordnung
|   |   |-- 20-Minuten-Messung
|   |   |-- ADC-Einstellungen
|   |-- Probandenmessungen
|-- Referenzen
```

Erklärung zur selbständigen Bearbeitung

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbständig verfasst und nur die angegebenen Hilfsmittel benutzt habe. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich gemacht.

Ort	Datum	Unterschrift im Original