

# **Entwicklung einer Methode für Vorhersagen über die Fortführung von Handbewegungen**

**Studienarbeit (T3\_3200)**

im Rahmen der Prüfung zum

**Bachelor of Science (B.Sc.)**

des Studienganges Informatik

an der Dualen Hochschule Baden-Württemberg Karlsruhe

von

**Nicolas Bender und Philipp Jakob Rall**

Mai 2021

Abgabedatum: 17. Mai 2021

Bearbeitungszeitraum: 28.09.2020 - 17.05.2021

Kurs: TINF18B1 & TINF18B2

Gutachter der Dualen Hochschule: Dr. Oliver Rettig

# Eidesstattliche Erklärung

Ich versichere hiermit, dass ich meine Studienarbeit (T3\_3200) mit dem Thema:

*Entwicklung einer Methode für Vorhersagen über die Fortführung von Handbewegungen*

gemäß § 5 der „Studien- und Prüfungsordnung DHBW Technik“ vom 29. September 2017 selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Die Arbeit wurde bisher keiner anderen Prüfungsbehörde vorgelegt und auch nicht veröffentlicht.

Ich versichere zudem, dass die eingereichte elektronische Fassung mit der gedruckten Fassung übereinstimmt.

Bietigheim-Bissingen, den 18. Mai 2021

---

Bender, Nicolas und Rall, Philipp Jakob

## **Abstract**

- *English* -

This thesis deals with the development of a method for real-time prediction of trajectories of lateral grasping movements for collision avoidance in collaborative robotics.

A neural network predicts the final position and duration of the entire grasping process based on the course of the initial movement in a regression approach. Using the minimum jerk model for curved point-to-point movements, the further trajectory can then be precisely calculated.

The thesis focuses on the development of an automated data pre-processing pipeline that transforms recorded raw data from natural grasping movements into high-quality and unified training data in several modular processing phases.

Furthermore, it sorts out erroneous measurement data.

## **Abstract**

*- Deutsch -*

Die vorliegende Arbeit befasst sich mit der Entwicklung einer Methode zur Echtzeit-Vorhersage von Trajektorien seitlicher Greifbewegungen zur Kollisionsvermeidung in der kollaborativen Robotik.

Ein Neuronales Netz sagt hierfür anhand des Verlaufs der Anfangsbewegung in einem Regressionsansatz die Endposition und Dauer des gesamten Greifvorgangs voraus. Durch das Minimum Jerk Model für gekrümmte Punkt-zu-Punkt-Bewegungen lässt sich daraufhin der weitere Verlauf der Trajektorie präzise berechnen.

Die Arbeit legt besonderen Fokus auf die Entwicklung einer automatisierten Pipeline zur Datenvorverarbeitung, die aufgenommene Rohdaten von natürlichen Greifbewegungen in mehreren modularen Verarbeitungsphasen zur qualitativ hochwertigen und vereinheitlichten Trainingsdaten transformiert sowie fehlerbehaftete Messdaten aussortiert.

# Inhaltsverzeichnis

<b>Symbol- und Abkürzungsverzeichnis</b>	<b>VI</b>
<b>Abbildungsverzeichnis</b>	<b>VIII</b>
<b>Tabellenverzeichnis</b>	<b>IX</b>
<b>1. Einleitung</b>	<b>1</b>
1.1. Motivation . . . . .	1
1.2. Zielsetzung . . . . .	2
<b>2. Grundlagen</b>	<b>3</b>
2.1. Datenaufnahme . . . . .	3
2.2. Projektaufbau . . . . .	5
<b>3. Stand der Forschung</b>	<b>8</b>
3.1. Überblick . . . . .	8
3.2. Ansatz . . . . .	15
3.3. Einordnung . . . . .	15
<b>4. Datenvorverarbeitung</b>	<b>17</b>
4.1. Vorüberlegungen . . . . .	17
4.2. Interpolation . . . . .	20
4.3. Glättung . . . . .	21
4.4. Beschneidung . . . . .	25
4.5. Präfilterung . . . . .	31
4.6. Spiegelung . . . . .	33
4.7. Relokation . . . . .	41
4.8. Filterung . . . . .	42
4.9. Rotation . . . . .	53
<b>5. Vorhersagen über die Trajektorie</b>	<b>57</b>
5.1. Ansätze für Machine Learning . . . . .	57
5.2. Umsetzung . . . . .	58
5.3. Pfadberechnung . . . . .	64
<b>6. Ergebnisse</b>	<b>65</b>
6.1. Datenvorverarbeitung . . . . .	65
6.2. Machine Learning . . . . .	70

6.3. Pfadberechnung . . . . .	72
<b>7. Diskussion</b>	<b>75</b>
7.1. Datenvorverarbeitung . . . . .	75
7.2. Machine Learning . . . . .	78
7.3. Praxistauglichkeit . . . . .	78
<b>8. Fazit &amp; Ausblick</b>	<b>81</b>
8.1. Fazit . . . . .	81
8.2. Ausblick . . . . .	81
<b>Literaturverzeichnis</b>	<b>X</b>
<b>Appendices</b>	<b>XI</b>
<b>A. Weitere Abbildungen</b>	<b>XII</b>

# Symbol- und Abkürzungsverzeichnis

## Symbole

$\kappa_v$	Schwellwert der Geschwindigkeit
$v_x$	Geschwindigkeit an einem bestimmten Punkt der Bahnkurve
$x_{v_{max,x_1}}$	Punkt der betragsmäßig größten Geschwindigkeit in Richtung der $x_1$ -Achse
$P_1$	Startpunkt der Bewegung bzw. der Bahnkurve
$P_2$	Endpunkt der Bewegung bzw. der Bahnkurve
$P_3$	Punkt der Bahnkurve mit der größten Entfernung von der Verbindungsline zwischen Start- und Endpunkt
$P_s$	Zu spiegelnder Punkt auf der Bahnkurve
$P'_s$	Gespiegelter Punkt auf der Bahnkurve
$E_s$	Spiegelebene
$M_s$	Punkt auf der Spiegelebene, der senkrecht zum zu spiegelnden Punkt steht
$M_3$	Punkt auf der Spiegelebene, der senkrecht zu $P_3$ steht
$\gamma_m \in \mathbb{R}^d$	Vektor der Metrikwerte aller $d$ Datensätze in Metrik $m$
$\kappa_{a,m,v}^{oben}$	Oberer Schwellwert bei Metrik $m$ und Verteilung $v$ berechnet mittels Ansatz $a$
$\kappa_{a,m,v}^{unten}$	Unterer Schwellwert bei Metrik $m$ und Verteilung $v$ berechnet mittels Ansatz $a$
$\omega_m \in \mathbb{R}^d$	Vektor der Abstände der Metrikwerte aller $d$ Datensätze in Metrik $m$ zu den Mittelwerten derer Verteilungen
$\rho_m \in \mathbb{R}^d$	Vektor der normierten Ausreißerintensitäten aller $d$ Datensätze in Metrik $m$
$\epsilon \in \mathbb{R}^d$	Vektor der Fehlerstärken aller $d$ Datensätze
$\pi_m$	Gewichtungsfaktor der Metrik $m$

## Abkürzungen

<b>API</b>	Application Programming Interface
<b>CNN</b>	Convolutional Neural Network
<b>CSV</b>	Comma-separated values
<b>k-NN</b>	k-nearest neighbors
<b>MAE</b>	Mean Absolute Error
<b>MSE</b>	Mean Squared Error
<b>NaN</b>	Not a Number
<b>ReLU</b>	Rectified Linear Unit
<b>RNN</b>	Recurrent Neural Network
<b>SVM</b>	Support Vector Machine

# Abbildungsverzeichnis

1.1. Angestrebtes Gesamtszenario . . . . .	2
2.1. Aufbau zur Datenaufnahme . . . . .	4
2.2. Markercluster am Handgelenk . . . . .	5
4.1. Flowchart für die Pipeline der Datenvorverarbeitung . . . . .	19
4.2. Interpolationslängen mit Anteil betroffener Dateien . . . . .	21
4.3. Vergleich der verschiedenen Glättungsmethoden und -parameter . . . . .	24
4.4. Beispielhafte Trajektorien . . . . .	25
4.5. Histogramm über Dateilängen vor Präfilterung . . . . .	33
4.6. Trajektorie der rechten und linken Körperhälfte mit Sicht von oben . . . . .	34
4.7. Bestimmung der Senkrechten zum zu spiegelnden Punkt . . . . .	36
4.8. Berechnung der Spiegelebene . . . . .	38
4.9. Spiegelebene zwischen den Trajektorien . . . . .	41
4.10. Visualisierung der Endpunkte . . . . .	42
4.11. Gemittelte Ellbogen-Handgelenkdistanzen aller Datensätze . . . . .	46
4.12. Metrikwerte der Ellbogen-Handgelenks-Metrik einer Verteilung mit dynamisch bestimmten Schwellwerten . . . . .	47
4.13. Dreidimensionale Visualisierung von Rotationswinkels und Drehachse . . . . .	54
5.1. Architektur des neuronalen Netzes . . . . .	60
6.1. Vergleich des originalen mit dem gespiegelten Datensatz . . . . .	68
6.2. Gespiegelte linke im Vergleich zur rechten Körperhälfte . . . . .	68
6.3. Visualisierung aller gefilterten Handgelenktrajektorien . . . . .	69
6.4. Visualisierung aller rotierten Handgelenktrajektorien . . . . .	70
6.5. Visualisierung exemplarischer Ergebnisse in Koordinatensystem . . . . .	72
6.6. Vergleich der tatsächlichen zur berechneten Trajektorie . . . . .	73
6.7. Gegenüberstellung von tatsächlicher und berechneter Trajektorie . . . . .	74
7.1. Ablauf zur Anwendung in der Praxis . . . . .	80
A.1. Boxplots für die Länge der Datensätze . . . . .	XIII
A.2. Visualisierung des Lernerfolgs anhand der Trainingsdaten . . . . .	XIV
A.3. Visualisierung des Lernerfolgs anhand der Testdaten . . . . .	XV

# **Tabellenverzeichnis**

4.1. Vergleich der beiden Methoden zur Beschneidung . . . . .	30
6.1. Statistische Analyse der Aussortierungsgründe von Datensätzen . . . . .	66
6.2. Stichprobenartige Prüfung der Trainingsqualität . . . . .	72

# 1. Einleitung

## 1.1. Motivation

Mit dem Fokus auf innovative Produktionsmethoden und technologische Strategien im Zeichen von Industrie 4.0 werden Szenarien zunehmend realistischer, in denen sich Menschen und Roboter einen gemeinsamen Arbeitsplatz teilen. Auch außerhalb der Industrie im medizinischen Bereich ergeben sich Einsatzmöglichkeiten, in denen die Arbeit von Robotern die menschliche ergänzen kann [1].

Unter dem Stichwort Human-Robot-Collaboration sind Situationen denkbar, in denen Menschen Roboter nicht nur aus der Ferne bedienen und warten, sondern auch zeitgleich mit Robotern zusammenarbeiten, um kollaborativ Aufgaben bearbeiten zu können. Um diese geteilten Arbeitsplätze für Menschen sicher zu gestalten, ist es unabdingbar, technische Vorsichtsmaßnahmen zu implementieren. Damit soll verhindert werden, dass abhängig vom Szenario Menschen von leistungsstarken Robotern verletzt oder empfindliche Roboter von Menschen beschädigt werden [2]. Diese Maßnahmen können in Form von Kontrollmechanismen für Roboter eingesetzt werden.

Eine Form des Kontrollmechanismus ergibt sich aus der Beobachtung der menschlichen Bewegung, die in den Kontext der Bewegungen des Roboters gesetzt werden. Bewegungsvorgänge des Roboters haben den Vorteil gegenüber den menschlichen Bewegungen, dass diese präzise gesteuert, lokalisiert und vorhergesagt werden können. Schwieriger gestaltet sich die Betrachtung der Bewegung des Menschen, die nach von außen nicht zwingend ersichtlichen Einflüssen erfolgt und deshalb kaum von außerhalb kontrolliert werden kann. Die Kontrollmechanismen des Roboters müssen sich daher an die Spontaneitäten der menschlichen Bewegungen anpassen und Prognosen für beispielsweise Handbewegungen treffen, um angemessen darauf reagieren zu können.

## 1.2. Zielsetzung

Das Ziel dieser Arbeit ist die Entwicklung und Bewertung einer Methode, mit der sich die Fortführung von Handgelenken vorhersagen lässt. Dabei soll auf Grundlage einer kurzen Bewegungssequenz der weitere Verlauf dieser Bewegung berechnet werden. Unter Zuhilfenahme des Maschinellen Lernens besteht das Ziel darin, ein Modell zu entwickeln, mit dem aus aufgenommenen Anfangsbewegungsdaten einer Armbewegung die Endposition des Handgelenks und Dauer der Bewegung prognostizieren kann.

Das Hauptaugenmerk der vorliegenden Arbeit liegt in der Erarbeitung eines Verfahrens zur automatisierten Verarbeitung aufgenommener Rohdaten von menschlichen Greifbewegungen zu qualitativ hochwertigen Trainingsdaten für das Maschinelle Lernen. Zur Aussortierung fehlerbehafteter Rohdaten sollen Kriterien und Metriken entwickelt werden, die automatisiert die Datenqualität feststellen können. Ein weiteres Ziel der Datenvorverarbeitung besteht in der Definition von Verfahren zur Vereinheitlichung und Steigerung der Vergleichbarkeit der aufgenommenen Daten.

Abbildung 1.1 stellt das angestrebte Gesamtszenario der Kollisionsvermeidung im Bereich der kollaborativen Robotik dar. Die vorliegende Arbeit befasst sich dabei nicht mit der eigentlichen Kollisionsberechnung sowie der Ermittlung von Handlungskonsequenzen für den Roboter, sondern konzentriert sich die Datenvorverarbeitung und die Vorhersage der Endpositionen.

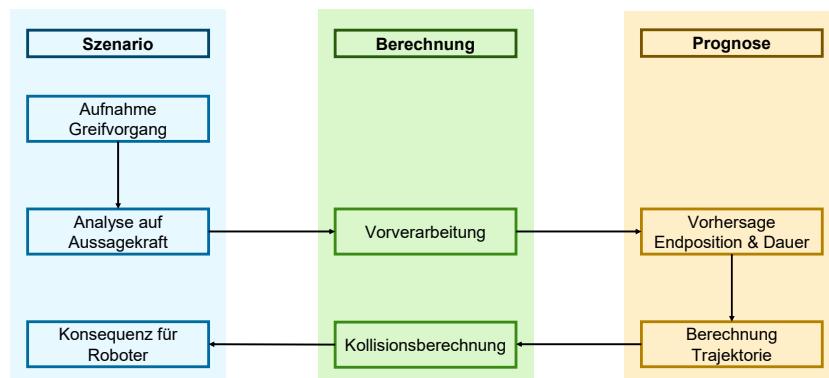


Abbildung 1.1.: Angestrebtes Gesamtszenario

## 2. Grundlagen

In diesem Kapitel sollen die Voraussetzungen für diese Arbeit erläutert und ein Blick auf den Aufbau des Projekts geworfen werden.

### 2.1. Datenaufnahme

Dieser Abschnitt stellt die Art und Weise der Aufnahme von Daten menschlicher Greifbewegungen vor und erläutert die dazu verwendeten Technologien. Dazu soll zunächst das Szenario der Datenaufnahme beschrieben werden, bevor auf die verwendete Markertechnik eingegangen wird. Abbildung 2.1 zeigt den Aufbau zur Datenaufnahme.

#### 2.1.1. Szenario

Zur Datenaufnahme wird ein Szenario der kollaborativen Robotik mit einem Roboter und einer Testperson simuliert. Ziel des Szenarios ist die Übergabe eines Paketes vom Roboter an den Menschen. Als Roboterarm wird das Modell UR5e von Universal Robots verwendet. Das Szenario besteht aus drei wesentlichen Schritten.

1. Roboter nimmt Paket auf
2. Roboter überreicht Testperson Paket
3. Testperson greift Paket und legt dieses ab

Die zum Test benötigten Pakete werden über eine schiefe Ebene dem Roboter kontinuierlich zugeführt und immer an einer festen Position bereitgestellt. Im ersten Schritt identifiziert der Roboter durch eine Kamera zunächst die Position des Pakets, um die Bewegung zum Paket durchzuführen. Anschließend wird das identifizierte und lokalisierte Paket durch ein Unterdrucksystem gegriffen und kann vom Roboter im Raum bewegt werden.

Im zweiten Schritt des Szenarios positioniert der Roboter das Paket an einer von 346

vorgegebenen Positionen im Raum. Dazu ist der Raum wie in [3] beschrieben in Würfel mit einer Kantenlänge von 10cm gleichmäßig aufgeteilt. Einige Würfel können vom Roboter allerdings nicht angefahren werden, wodurch sich die 346 Positionen ergeben.

Nach Abschluss der Bewegung kann die Testperson das Paket greifen und auf einem Tisch ablegen. Diese zielgerichtete seitliche Greifbewegung ist die für die Datenaufnahme relevante Bewegung. Deshalb ist es sehr wichtig, dass die Greifbewegung aus der Ruhe und gleichmäßig beginnt.



Abbildung 2.1.: Aufbau zur Datenaufnahme

### 2.1.2. Markertechnik

Die Aufnahme der Bewegungsdaten erfolgt über am Körper der Testperson angebrachte reflektierende Cluster aus Markern und ein 120-Hz-Vicon-Kamerasystem. Dieses besteht wie in Abbildung 2.1 abgebildet aus acht in einem Quadrat über dem Testszenario angeordneten Infrarotkameras, welche das von den Markern reflektierte Infrarotlicht aus mehreren Perspektiven erkennen können. Sobald mehr als zwei Kameras einen Markercluster erfassen, kann über Triangulation die Position des Markers im Raum berechnet werden.

Ein Cluster an Markern besteht jeweils aus vier Markern, über die sich an jedem Cluster

ein Koordinatensystem berechnen lässt. Dabei ist jedes Cluster von seiner Anordnung unterschiedlich, um eine Unterscheidbarkeit zu gewährleisten. Für diese Arbeit sind vor allem die Cluster am Handgelenk, Ellbogen und Schulter der Testpersonen relevant. Die Cluster werden entsprechend der in der Arbeit von [3] sowie [4] genannten Positionen angebracht. Abbildung 2.2 zeigt ein Markercluster am Handgelenk einer Testperson.



Abbildung 2.2.: Markercluster am Handgelenk einer Testperson

Anhand der Marker lassen sich die Trajektorien der Handgelenke, Ellbögen und Schultern der Testpersonen während des Greifvorgangs ermitteln und in Form von kartesischen Koordinaten im Raum je Kamerabild weiterverwenden.

So entsteht pro Greifbewegung je Körperhälfte eine Comma-separated values (CSV)-Datei mit den Raumkoordinaten für Handgelenk-, Ellbogen- und Schulterzentrum, die als Eingang für die weitere Datenverarbeitung wie in Kapitel 4 beschrieben dient.

## 2.2. Projektaufbau

Dieser Abschnitt soll einen Überblick über verwendeten Technologien im Rahmen des Projekts geben und deren Wahl begründen. Grundsätzlich lassen sich im Projekt für Technologien die drei Anwendungsbereiche Visualisierung, Analyse & Verarbeitung und Machine Learning ausmachen, die diesen Abschnitt gliedern.

## 2.2.1. Visualisierung

Der Anwendungsbereich der Visualisierung lässt sich in zwei Arten der Visualisierung unterteilen. Zum einen ist die Visualisierung der aufgenommenen und verarbeiteten Daten eine wichtige Aufgabe, um deren Qualität nachzuvollziehen. Zum anderen besteht eine Aufgabe darin, die Gesamtszene zum besseren Verständnis geometrischer Operationen während der Datenverarbeitung zu visualisieren.

Zur **Datenvisualisierung** wird die kostenfreie wissenschaftliche Programmiersprache *GNU Octave* [5] eingesetzt werden, da sie einsteigerfreundlich dreidimensionale Visualisierungen von CSV-Dateien mittels Skripten ermöglicht. In folgendem Abschnitt sollen einige für diese Arbeit wesentliche Skripte exemplarisch vorgestellt und deren Anwendungsfälle erläutert werden.

Grundsätzlich ist eine Skript zur Visualisierung eines beliebigen Datensatzes in einem beliebigen Verarbeitungsschritt notwendig, um einen Eindruck des Ergebnisses des Verarbeitungsschrittes und ein besseres Verständnis der Datenqualität zu erlangen. Hierfür wird das Skript `drawData(csvNumber, phaseNumber)` verwendet, das die Trajektorien von Handgelenk, Ellbogen und Schulter beider Körperseiten - falls vorhanden - in einem dreidimensionalen kartesischen Koordinatensystem visualisiert. Sofern eine Trajektorie im Rahmen einer Filterung während der Verarbeitung aussortiert wird, zeigt dieses Skript auch die Ursache der Aussortierung an (siehe 4.5 und 4.8).

Eine weitere Visualisierungsaufgabe besteht darin, die Trajektorien aller Datensätze zu einem bestimmten Verarbeitungszeitpunkt in einem Koordinatensystem zu visualisieren, um die Wirkung der Verarbeitung auf allen Datensätzen gesammelt zu betrachten. Dazu wird das Skript `drawAllFiles(phaseNumber, side, bodyPart)` verwendet. Im Falle der Visualisierung rotierter Datensätze (siehe 4.9) stellt das Skript auch die Positionen der jeweiligen Rotationspunkte dar. Eine Anwendung des Skriptes ist in den Abbildungen 6.3 und 6.4 auf den Seiten 69 und 70 zu sehen.

Um zusätzlich einen visuellen Eindruck über den Geschwindigkeitsverlauf eines Datensatzes zu erhalten, kann das Skript `animateData(csvNumber, phaseNumber, side)` eingesetzt werden. Es stellt die selektierte Trajektorie des Handgelenks animiert dar.

Für komplexe geometrische Transformationen, wie die auf den Seiten Abschnitt 4.6 und Abschnitt 4.9 vorgestellte Spiegelung und Rotation der Trajektorien, ist eine **Visualisierung der gesamten Szene** hilfreich. Dazu wird Geometrie-Software *GeoGebra* [6]

verwendet, in der einsteigerfreundlich konstruierte geometrische Objekte visualisiert und verständlich hergeleitet werden können.

### 2.2.2. Analyse & Verarbeitung

Die Datenanalyse und -verarbeitung soll in der Programmiersprache Python erfolgen, da dort etablierte wissenschaftliche Bibliotheken wie *pandas* [7] oder *NumPy* [8] zum Umgang mit großen Datenmengen existieren. Zudem ermöglicht ein Jupyter Notebook, ein Textdokument mit ausführbarem Code, eine gute Dokumentation verschiedener Datenverarbeitungsschritte. Zur Visualisierung von mathematischen Darstellungen wie beispielsweise Geschwindigkeitsverläufen eignet sich die Python-Bibliothek *Matplotlib* [9] und wird daher in dieser Arbeit verwendet.

### 2.2.3. Machine Learning

Als Machine Learning-Bibliothek verwendet die vorliegende Arbeit die Open Source-Deep-Learning-Bibliothek *Keras* in Python, die als high-level Application Programming Interface (API) für die Machine Learning Plattform *TensorFlow* dient. Keras ermöglicht eine einsteiger- und nutzerfreundliche Verwendung von *TensorFlow* [10].

## 3. Stand der Forschung

Dieses Kapitel gibt zunächst einen Überblick über die wissenschaftliche Grundlage, auf die diese Arbeit aufbaut. Anschließend erfolgt eine Einordnung der Arbeit und die Definition des zu verfolgenden Ansatzes.

### 3.1. Überblick

Die Grundproblematik in der Betrachtung von menschlichen Armbewegungen liegt darin, dass es beliebig viele Möglichkeiten geben kann, einen Pfad und die Geschwindigkeit von einem Startpunkt zu einem Endpunkt zu wählen. Für hindernisfreie Bewegungsvorgänge sieht die Literatur die Gemeinsamkeiten für die physikalische Beschreibung in vier Aspekten [11][12].

1. Möglichst glatter Pfad und ruckfreie Bewegung
2. Möglichst geradliniger Pfad
3. Unimodales glockenförmiges Geschwindigkeitsprofil
4. Skalierung der Spitzengeschwindigkeit und -beschleunigung sowie deren Zeitpunkte mit der Bewegungsamplitude

Es wird davon ausgegangen, dass sich die Berechnung eines Bewegungsvorgangs durch das menschliche Gehirn anhand dieser Faktoren annähern lässt. Für den möglichst glatten und geradlinigen Pfad sind Szenarien gemeint, in denen bei einem Greifvorgang ein Endpunkt, eventuell in Form eines Zielobjekts, anvisiert und eine Hand hindernisfrei darauf zu bewegt werden kann. Der Geschwindigkeitsverlauf der Hand ist dabei glockenförmig und enthält in der Spitze eine maximale Geschwindigkeit. Diese Höchstgeschwindigkeit und die maximale Beschleunigung und deren Zeitpunkte sind abhängig von der Entfernung des Endpunkts.

### 3.1.1. Modellkategorien

Für die Vorhersagen von Bewegungsvorgängen werden in der Literatur diverse Herangehensweisen vorgestellt, um einen menschlichen Bewegungsvorgang zu modellieren. Campos und Calado [13] teilen diese in drei verschiedene Kategorien ein, die im Folgenden vorgestellt werden.

1. Motor Execution Models
2. Complete Models
3. Descriptive Models

**Motor Execution Models** gehen davon aus, dass die Bewegungsbahn berechnet werden kann und fokussieren sich auf die Kontrolle der menschlichen kinematischen Strukturen, um die berechnete Bewegungsbahn auszuführen [13]. Dabei handelt es sich um interne Modelle, welche die Dynamik des Körpers und den Einfluss der Umgebung repräsentieren.

**Complete Models** sind Berechnungsmodelle für Trajektorien, die dafür grundlegende biologische Prozesse und muskulären Abläufe, die vom Zentralen Nervensystem ausgeführt werden, miteinbeziehen [13]. Beispiele hierfür aus der Literatur stellen das Minimum Force bzw. Minimum Work/Energy und das Minimum Torque Change Model dar. In diesen Modellen wird die Kostenfunktion des Drehmoments eines zu betrachtenden Gelenks minimiert [14]. Der Vorteil dieser Modelle ist eine sehr präzise Bestimmung des Bewegungsverlaufs. Ein Nachteil ist die Betrachtung der Dynamiken, die sich durch die Anatomie des entsprechenden Gelenks ergeben. In der Gesamtbetrachtung mit mehreren Gelenken und den vielen Parametern wird die Anwendung dieser Modelle sehr aufwendig [15][16].

Die dritte Kategorie der **Descriptive Models** stellen Modelle zur Berechnung von Bewegungsbahnen dar, die gut zu den in der Realität beobachtbaren Trajektorien passen. Im Gegensatz zu den zuvor beschriebenen Complete Models, besteht das Ziel dieser Modelle nicht darin, die muskulären Abläufe miteinzubeziehen, welche die Bewegungsbahn bewirken. Descriptive Models konzentrieren sich auf die Beschreibung der resultierenden Trajektorie, indem sie Parameter wie Positionsdaten, Krümmungen, Distanzen oder die Dauer berechnen bzw. interne Zusammenhänge dieser Parameter quantifizieren.

Ein Beispiel für ein deskriptives Modell ist Fitts' Law, das für schnelle zielgerichtete Bewegungen die Dauer der Bewegung vorhersagt [17]. Ein weiteres Modell ist das

Zweidrittel-Kraft-Gesetz, das den Zusammenhang von Geschwindigkeit und Krümmungsradius einer Bewegung beschreibt [18].

Zudem existieren deskriptive Modelle für die vollständige Berechnung von Bewegungsbahnen, die für vorliegende Arbeit besonders relevant sind und im folgenden Abschnitt thematisiert werden.

### **3.1.2. Vorhersage von Handgelenkbewegungen**

Für die Nutzung für Vorhersagen von Handgelenkbewegungen sind deskriptive Modelle eine viel genutzte Herangehensweise. Dieser Abschnitt stellt dazu zunächst das Minimum Jerk Model und das Modified Minimum Jerk Model vor, die jeweils Trajektorien vollständig berechnen. Anschließend soll ein weiterer Ansatz aus der Forschung thematisiert werden, der keine exakte Berechnung der Trajektorien durchführt, sondern lediglich wahrscheinliche Aufenthaltsbereiche der Hand zu jedem Zeitpunkt liefert.

#### **Minimum Jerk Model**

In der Forschung findet sich als etabliertes Modell der Vorhersage von menschlichen Bewegungen das Minimum Jerk Model von Flash und Hogan [12]. Diese Methode sollte eine mathematische Herangehensweise zur Verfügung stellen, die ohne Einbezug der dahinterliegenden anatomischen Faktoren das Verhalten von willkürlichen menschlichen Bewegungen präzise modellieren kann. Der Grundgedanke dieses Modells ist die Eigenschaft, dass eine menschliche Bewegung einen Bewegungsablauf durchführt, der den kleinsten möglichen Ruck verursacht. Der Ruck ist dabei die dritte Ableitung des Ortes nach der Zeit und physikalisch gesehen damit die Änderung der Beschleunigung. Das Minimum Jerk Model sieht vor, die Trajektorie der Hand bei einem hindernisfreien geradlinigen Greifvorgang zu approximieren und hat als Rahmenbedingung, dass die Bewegung aus dem Stillstand startet und sich nur in einer zweidimensionalen Ebene abspielt. Das Modell lässt sich auch für Bewegungen im dreidimensionalen Raum anwenden [15]. Für die Beschreibung eines Bewegungsvorgangs muss das Integral über die Quadrate des Rucks der einzelnen Dimensionen minimiert werden. Somit lässt sich als zu minimierende

Kostenfunktion C in Gleichung (3.1) definieren:

$$C = \frac{1}{2} \int_0^{t_f} \left( \left( \frac{d^3x}{dt^3} \right)^2 + \left( \frac{d^3y}{dt^3} \right)^2 + \left( \frac{d^3z}{dt^3} \right)^2 \right) dt \quad (3.1)$$

Durch die Minimierung dieser Kostenfunktion, in der  $x$ ,  $y$  und  $z$  Funktionen über der Zeit  $t$  sind, ergeben sich nach Flash und Hogan [12] für verschiedene Arten von Armbewegungen jeweils Funktionen für jede Koordinate, die eine Bewegung mit minimalem Ruck beschreiben. Im Rahmen dieser Arbeit sollen **unbeschränkte Punkt-zu-Punkt Bewegungen** und **gekrümmte Punkt-zu-Punkt Bewegungen** betrachtet werden.

Bei **unbeschränkte Punkt-zu-Punkt Bewegungen** handelt es sich um die klassischen, geradlinigen und hindernisfreie Bewegungsverläufe. Hierfür ergeben sich für die Koordinaten der Trajektorie nach Flash und Hogan [12] die Funktionen 3.2 bis 3.4. Die benötigten Eingabeparameter sind dabei die Startposition  $(x_0, y_0, z_0)$ , die Endposition  $(x_f, y_f, z_f)$  und die Dauer der Bewegung  $t_f$ . Mit  $\tau = \frac{t}{t_f}$  ergibt sich:

$$x(t) = x_0 + (x_0 - x_f)(15\tau^4 - 6\tau^5 - 10\tau^3) \quad (3.2)$$

$$y(t) = y_0 + (y_0 - y_f)(15\tau^4 - 6\tau^5 - 10\tau^3) \quad (3.3)$$

$$z(t) = z_0 + (z_0 - z_f)(15\tau^4 - 6\tau^5 - 10\tau^3) \quad (3.4)$$

**Gekrümmte Punkt-zu-Punkt Bewegungen** stellen Bewegungskurven von einem Startpunkt  $(x_0, y_0, z_0)$  über einen Zwischenpunkt  $(x_1, y_1, z_1)$  zu einem Zielpunkt  $(x_f, y_f, z_f)$  dar und können somit einem Hindernis ausweichen. Der Zwischenpunkt sollte dabei die maximale Distanz zur direkten Verbindung von Start- und Endpunkt aufweisen. Zur Beschreibung der Trajektorie ergeben sich durch die Optimierung je Koordinate zwei Funktionen: Eine Funktion für die Punkte vor dem Zwischenpunkt und eine für die Punkte danach. Am Beispiel der x-Koordinate werden diese mit  $x^-(\tau)$  für  $t \leq t_1$  und  $x^+(\tau)$  für  $t \geq t_1$  bezeichnet, mit  $\tau = \frac{t}{t_f}$  und  $\tau_1 = \frac{t_1}{t_f}$  ergeben sich für die x-Koordinate

folgende Funktionen.

$$x^-(\tau) = \frac{t_f^5}{720} (\pi_1(\tau_1^4(15\tau^4 - 30\tau^3) + \tau_1^3(80\tau^3 - 30\tau^4) - 60\tau^3\tau_1^2 + 30\tau^4\tau_1 - 6\tau^5) \\ + c_1(15\tau^4 - 10\tau^3 - 6\tau^5)) + x_0 \quad (3.5)$$

$$x^+(\tau) = x^-(\tau) + \pi_1 \frac{t_f^5(\tau - \tau_1)^5}{120} \quad (3.6)$$

Die Konstanten  $c_1$  und  $\pi_1$  sind wie folgt definiert.

$$c_1 = \frac{1}{t_f^5 \tau_1^2 (1 - \tau_1)^5} ((x_f - x_0)(300\tau_1^5 - 1200\tau_1^4 + 1600\tau_1^3) \\ + \tau_1^2(-720x_f + 120x_1 + 600x_0) + (x_0 - x_1)(300\tau_1 - 200)) \quad (3.7)$$

$$\pi_1 = \frac{1}{t_f^5 \tau_1^5 (1 - \tau_1)^5} ((x_f - x_0)(120\tau_1^5 - 300\tau_1^4 + 200\tau_1^3) - 20(x_1 - x_0)) \quad (3.8)$$

Für  $y$  und  $z$  ergeben sich analoge Funktionen.

Zu beachten ist hierbei auch, dass unter realen Bedingungen die Anfangsgeschwindigkeit einer Bewegung ungleich null sein kann. Für diesen Fall schlagen Fligge, McIntyre und van der Smagt [15] eine Anpassung der Kostenfunktion und der Funktion der Trajektorie vor, die durch das Minimum Jerk Model von Flash und Hogan vorgegeben wird. Dafür werden die Koeffizienten des Polynoms so gewählt, dass die erste und zweite Ableitung der Anfangs- und Endgeschwindigkeit beachtet werden.

### Modified Minimum Jerk Model

Als Anpassung sieht es das Modified Minimum Jerk Model vor, das Minimum Jerk Model um einen Term zu ergänzen. Mit diesem soll die Präzision und die Robustheit erhöht werden, da sich in der Praxis zeigt, dass verschiedene Personen unterschiedliche Bewegungsabläufe aufweisen, die nicht ausschließlich über die Minimierung des Rucks verallgemeinert werden können. Für eine realistischere Annäherung schlägt Zhao u. a. [19] vor, einen Modifikationsterm in Form einer Fourierreihe zu ergänzen, der den Fehler zwischen Messung und Berechnung über das Minimum Jerk Model ausgleichen soll. Die Arbeit von Zhao u. a. [19] beschreibt mittels des Modified Minimum Jerk Models eine Prognose für die Fortführung einer Handbewegung anhand der Positionsdaten, der

Geschwindigkeit und der Beschleunigung in der Anfangsphase (early-stage motion) der Bewegung. Hierfür wird das erste Drittel der Bewegung einer Fingerspitze aufgenommen, mittels maschinellem Lernen die Dauer und die Endposition vorhergesagt und anhand dieser Daten der Verlauf der Trajektorie bestimmt.

### **Wahrscheinliche Aufenthaltsbereiche im Raum**

Eine weitere Möglichkeit zur Vorhersage Bewegungsverläufen zur Kollisionsvermeidung in kollaborativen Robotikszenarien ist die Bestimmung von sicherheitskritischen Regionen im Raum, in denen sich ein Roboterarm nicht aufhalten sollte, um nicht mit dem menschlichen Arm zu kollidieren. Dafür wird der Raum in gleichgroße Gebiete unterteilt und Beobachtungsräume definiert, in denen Vektoren zeitdiskrete Positions- und Geschwindigkeitsinformationen zugewiesen werden. Über das Hidden Markov Modell kann so für jeden Zeitpunkt eine Kollisionswahrscheinlichkeit vorhergesagt werden. Für eine Verbesserung der Ergebnisse dieses Vorgehens lassen sich nicht erreichbare Gebiete ausschließen [20].

### **3.1.3. Datenvorverarbeitung**

Dieser Abschnitt gibt einen Überblick über in der Forschung verwendete Verfahren zur Datenvorverarbeitung.

Die Bestimmung der Geschwindigkeit muss bei Messdaten aufgrund von diskreten Werten numerisch erfolgen. Eine einfache Vorwärtsdifferenzierung ist in der Literatur dabei als ausreichend angesehen [20]. Extreme Ausreißer im Geschwindigkeitsprofil können anhand des Ausreißertests nach Grubbs erkannt werden [21].

Für die Rauschentfernung der aufgenommenen Messdaten nutzen Experimente den Savitzky-Golay Filter [19]. Darüber hinaus werden auch aufwendige Verfahren wie der Kalman Filter verwendet, die zufriedenstellende Ergebnisse für sehr verrauschte Daten liefern können [14][22].

Um den Beginn einer Bewegung zu definieren, wird von [19] vorgeschlagen, alles unter 1% der Maximalgeschwindigkeit des Bewegungsvorgangs als Stillstand anzunehmen. Damit lässt sich ausgehend vom Punkt der höchsten Geschwindigkeit in zeitlicher Richtung

nach vorne und nach hinten die Bewegung bis zu den Punkten eingrenzen, an denen der Stillstand erreicht wird.

### 3.1.4. Machine Learning

Für die Prognose der Trajektorien werden in der Literatur verschiedene Ansätze genutzt. Verbreitet ist die Verwendung des Hidden Markov Modells [20][23]. Als probabilistische Methode modelliert dieser Ansatz mithilfe einer Motion Transition Probability Matrix die Übergangswahrscheinlichkeiten von Zuständen in neue Zustände. Dabei wird festgelegten Raumbereichen zeitlich abhängig eine Kollisionswahrscheinlichkeit von menschlichem Handgelenk und Roboter zugewiesen.

Eine weitere Möglichkeit besteht darin, ein Modell zur Erkennung von Features zu entwickeln, die ausschlaggebend für bestimmte Bewegungsmuster sind. Die Einordnung der relevanten Features in einem Bewegungszustand können über den k-nearest neighbors (k-NN) Algorithmus verwirklicht werden [24].

Weiterhin kann zur Prognose von Bewegungsverläufen die Zuordnung von Zuständen mittels Support Vector Machines (SVMs) umgesetzt werden [25][26]. Ein weiterer Ansatz ist die Verwendung von Recurrent Neural Networks (RNNs) mit der Absicht, unbekannte Bewegungsabläufe von den Mustern bekannter Bewegungen abzuleiten [27].

In der Veröffentlichung von Wang u. a. [26] werden verschiedene Modelle verglichen. k-NN bietet den Vorteil, sehr einfach implementierbar zu sein, kann jedoch für große Datensätze sehr rechenintensiv werden. Die SVM als Mittel der Wahl stellt verschiedene Kernel zur Verfügung, die spezifisch für den Anwendungsfall zugeschnitten werden können. Die Komplexität steigt jedoch linear mit der Größe des Trainingsset. Durch die Anzahl der Supportvektoren kann daher bei einem großen Trainingsdatensatz hoher Rechenaufwand entstehen. Künstliche Neuronale Netze bieten die Möglichkeit, nichtlineare Zusammenhänge in den Datensätzen zu erkennen, es lassen sich aber auch bei erfolgreicher Implementierung diese Muster nicht transparent herauslesen. Das in der Arbeit von Wang u. a. [26] präferierte Hidden Markov Modell bietet einen großen Spielraum für Training und Verifikation des Modells und ist durchweg transparent, allerdings müssen viele Parameter an den Anwendungsfall angepasst werden.

Eine simple Methode ist die Nutzung des Minimum Jerk Models in Kombination mit der Schätzung der Endposition und des Endzeitpunkts über eine gewichtete Least-Squares

Gleichung. Für einfache Anwendungsfälle liefert diese Methode gute Ergebnisse. Der Anwendungsfall in der Arbeit von Maeda, Hara und Arai [28] nutzt dieses Verfahren für die Nachführung der menschlichen Armbewegung durch einen Roboter.

Für verallgemeinerte Anwendungen findet sich ein vielversprechender Ansatz für die Vorhersage von Endposition und Endzeitpunkt in der Arbeit von Zhao u. a. [19]. Hier wird die Gaussian Process Regression genutzt. Das Ergebnis dieser Methode erstreckt sich auf einen mittleren absoluten Fehler bei der Endposition von 20mm und beim Endzeitpunkt von 0.05s.

## 3.2. Ansatz

Die vorliegende Arbeit verfolgt den Ansatz, anhand der ersten 85 Datensätze einer Handbewegung über ein neuronales Netz die Endposition und Dauer der Bewegung vorherzusagen. Anhand dieser Informationen soll mittels des Minimum Jerk Models für gekrümmte Punkt-zu-Punkt Bewegungen die restliche Trajektorie berechnet werden. Als Zwischenpunkt dient hierzu der Punkt auf der Anfangstrajektorie mit der größten Distanz zur direkten Verbindung von Start- zum vorhergesagten Endpunkt.

Die Arbeit fokussiert sich auf seitliche Greifbewegungen der rechten Hand, linke Hände können durch Symmetrieeigenschaften auf rechte abgebildet werden.

## 3.3. Einordnung

Diese Arbeit schließt somit an die Ideen von Zhao u. a. [19] an und verwendet ebenfalls die Anfangstrajektorie zur Vorhersage der Endposition und Dauer der Bewegung über Maschinelles Lernen. Im Gegensatz zur Veröffentlichung von Zhao u. a. [19] soll in dieser Arbeit die Vorhersage der Endposition als Regressionsproblem aufgefasst und durch ein einfaches neuronales Netz bestimmt werden. Zudem setzt die vorliegende Arbeit auf das Minimum Jerk Model für gekrümmte Punkt-zu-Punkt Bewegungen statt dem in [19] vorgestelltem Modified Minimum Jerk Modell zur Berechnung der Trajektorie.

Weitere durch bisherige Forschung gelegte Grundlagen der Messtechniken und Datenvorverarbeitung sollen Berücksichtigung finden. So wird beispielsweise das in [3] vorgestellte Szenario zur Datenaufnahme oder der Ausreißertest nach Grubbs [21] verwendet.

# 4. Datenvorverarbeitung

In diesem Kapitel soll die in der Arbeit erarbeitete automatisierte Datenpipeline vorgestellt und erläutert werden, wie diese aus den Rohdaten der Datenaufnahme qualitativ hochwertige Trainingsdaten produziert.

## 4.1. Vorüberlegungen

Aufgrund der großen Datenmenge ist eine automatisierte Verarbeitung der Daten für eine effiziente Verwendung unumgänglich. Dabei soll die Datenverarbeitung als Pipeline mit verschiedenen Phasen strukturiert werden, um die einzelnen Verarbeitungsschritte transparent zu verdeutlichen und klar voneinander abzugrenzen. Zudem besteht so die Möglichkeit, die Verarbeitung um weitere Schritte modular zu erweitern oder die Reihenfolge der Phasen beliebig zu ändern. Die Raumkoordinaten der Trajektorien werden in jeder Phase in jeweils einer CSV-Datei abgelegt. Jede Phase besitzt dazu einen Ordner im Dateisystem und liest als Eingangsdaten die CSV-Dateien der vorherigen Phase ein und schreibt als Ausgangsdaten die verarbeiteten CSV-Dateien in ihren Ordner. Die erarbeitete Verarbeitungspipeline umfasst insgesamt acht Phasen, die in der folgenden Aufzählung kurz beschrieben sind. Abbildung 4.1 auf Seite 19 visualisiert den Gesamtablauf der Datenvorverarbeitung.

0. **Raw:** Umbenennung in einheitliches Namensformat
1. **Interpolated:** Interpolation kurzer Not a Number (NaN)-Strecken
2. **Smoothed:** Glättung der Trajektorien
3. **Truncated:** Einheitliche Beschneidung
4. **Prefiltered:** Filterung anhand statischer Kriterien (Präfilterung)
5. **Mirrored:** Spiegelung der linken Seiten auf die rechte Seite
6. **Relocated:** Verschiebung des Startpunktes in Ursprung

7. **Filtered:** Metrikbasierte Filterung mit dynamischen Schwellwerten
8. **Rotated:** Vereinheitlichung der Anfangsrichtung

Die Verarbeitung verfolgt zwei wesentliche Ziele, die im Folgenden dargelegt werden. Das erste Ziel der Datenvorverarbeitung besteht in der **Vereinheitlichung der Messdaten**, um eine Vergleichbarkeit sicherzustellen. Dabei ist es zunächst wichtig, eine einheitliche Benennung der Dateien festzulegen. Hierfür soll folgende Nomenklatur gelten:

<Datum><Nummer>\\_<Phase>\\_<L/R>.csv

Jede Messung beginnt mit dem Datum des Messtermins gefolgt von der dreistelligen Nummer des Messdurchlaufs. Daran schließt die Bezeichnung ihrer höchsten Verarbeitungsphase im Zuge der in diesem Kapitel beschriebenen Datenvorverarbeitung sowie die Angabe zur Körperseite (links oder rechts) an.

Folgende Verarbeitungsphasen dienen dem Ziel der Datenvereinheitlichung: Zunächst benennt die nullte Phase vor der eigentlichen Datenverarbeitung die Rohdaten entsprechend der festgelegten Namensgebung einheitlich um und korrigiert Tippfehler im Datumsformat in der Bezeichnung der Rohdaten. Die dritte Phase der Verarbeitungskette (siehe 4.4) beschneidet die Messdaten einheitlich, da diese teilweise mit einer längeren Ruhephase oder direkt mit der Bewegung starten. Die Phase der Spiegelung (Phase fünf) dient ebenfalls der Datenvereinheitlichung und normalisiert linke und rechte Daten, indem die Trajektorien der linken Körperhälfte auf die rechten gespiegelt werden. Ebenso Phase sechs der Verschiebung des Startpunktes auf den Ursprung sowie Verarbeitungsphase acht der Rotation der Richtung der Anfangstrajektorie in x-Achsenrichtung.

Das zweite Ziel der Datenvorverarbeitung liegt in der **Verbesserung der Qualität** der gemessenen Daten. Dazu dienen die Verarbeitungsphasen der Interpolation kurzer NaN-Strecken und Glättung der Trajektorien sowie die beiden Phasen zur Aussortierung fehlerbehafteter Datensätze. Diese Phasen vier und sechs filtern die Messdaten zunächst anhand statischer Kriterien wie der Anzahl an NaN-Werten, der Dateilänge oder Maximalgeschwindigkeit. Anschließend erfolgt eine gewichtete Aussortierung von Daten mit Ausreißern in Metriken wie der durchschnittlichen Ellbogen-Handgelenk-Distanz.

Die folgenden Abschnitte erläutern die einzelnen Phasen der Datenpipeline detailliert und legen deren Relevanz dar.

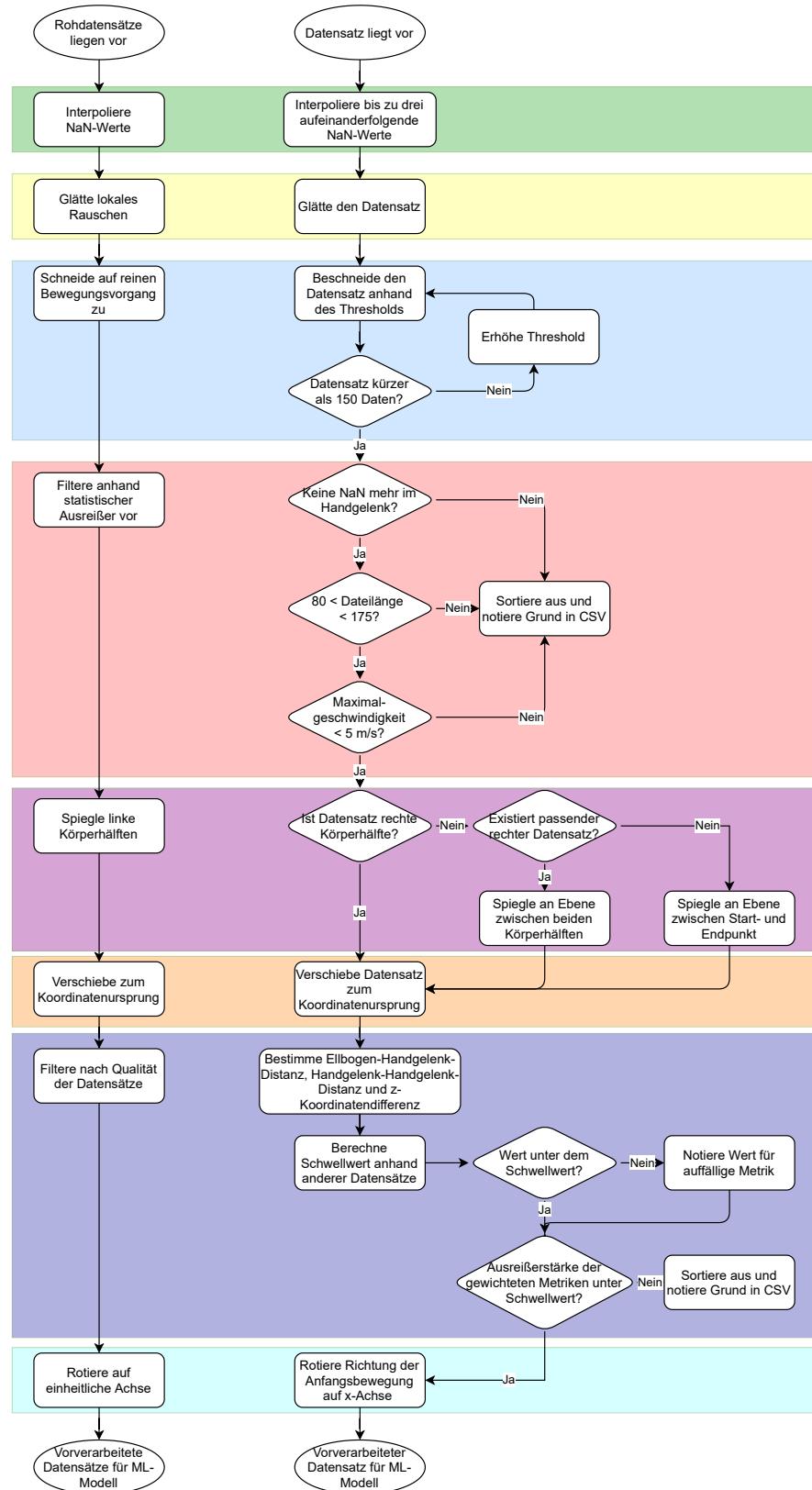


Abbildung 4.1.: Flowchart für die Pipeline der Datenvorverarbeitung, links der grobe Ablauf für den gesamten Datenbestand, rechts detailliert für einen Datensatz

## 4.2. Interpolation

Generell besteht bei Messungen unter realen Bedingungen häufig das Problem von NaN-Werten in den Messergebnissen aufgrund von Fehlern in der Messtechnik. Diese fehlenden Datenwerte würden die weitere Verarbeitung erheblich beeinträchtigen (siehe dazu [29]). Aus diesem Grund sollen bereits im ersten Schritt der Datenpipeline kurze NaN-Strecken in den Handgelenksdaten interpoliert werden, da diese wie in Abschnitt 3.2 auf Seite 15 erläutert für das Maschinelle Lernen die höchste Relevanz aufweisen. Zur Interpolation ist die maximale Länge der zu interpolierenden NaN-Folge sowie die Art der Interpolation festzulegen.

Lange NaN-Folgen über zahlreiche Zeitpunkte hinweg lassen sich nicht sinnvoll interpolieren und sollen in dieser Phase zunächst ignoriert werden. Im weiteren Verlauf der Datenvorverarbeitung verwendet die Phase der Präfilterung die Existenz von NaN-Werten in der Handgelenkstrajektorie als Kriterium zur Aussortierung von Datensätzen. Eine kurze maximale Interpolationslänge ermöglicht eine hohe Qualität der Interpolation, aber berücksichtigt dafür nur längere NaN-Folgen in Datensätzen nicht. Dadurch werden mehr Handgelenksdaten in der Phase der Präfilterung aussortiert. Eine lange maximale Interpolationslänge verringert die Aussortierungswahrscheinlichkeit in der Präfilterung dadurch, dass mehr NaN-Werte interpoliert werden können. Allerdings sinkt bei längeren Interpolationsstrecken auch die Qualität der Interpolation. Um eine Abwägung dieser Faktoren vorzunehmen, ist eine statistische Betrachtung der vorliegenden NaN-Folgen erforderlich.

Hierbei fällt auf, dass ca. 29% der Handgelenksdaten mindestens einen NaN-Wert aufweisen. In Abbildung 4.2 auf Seite 21 ist der Zusammenhang zwischen der Länge der längsten NaN-Folge der Handgelenksdaten eines Datensatzes und dem prozentualen Anteil aller Datensätze, die maximal diese NaN-Folgenlänge aufweisen, als Kurve dargestellt. Ziel ist eine Interpolationslänge zu wählen, die möglichst vielen Datensätzen die weitere Verarbeitung ermöglicht und gleichzeitig für eine hohe Interpolationsqualität möglichst gering ist. Wie im Kurvenverlauf sichtbar, sollte als maximale Interpolationslänge fünf gewählt werden, da ab fünf die Steigung der Kurve deutlich abnimmt. Die minimale Interpolationslänge ist eins, da erst ab einem NaN-Wert überhaupt interpoliert wird. Die ideale Abwägung ergibt sich durch das arithmetische Mittel der beiden Größen und liegt somit bei drei, was 84% der Dateien betrifft.

Für die Interpolationsart muss die allgemeine Gestalt der Kurvenverläufe betrachtet werden. Für eine aussagekräftige Interpolation von drei aufeinanderfolgenden Werten ist die Wahl eines Polynoms dritten Grades sinnvoll, da hier die wenigsten Stützstellen benötigt werden und jede mögliche Verteilung der drei zu interpolierenden Punkte abbildbar ist.

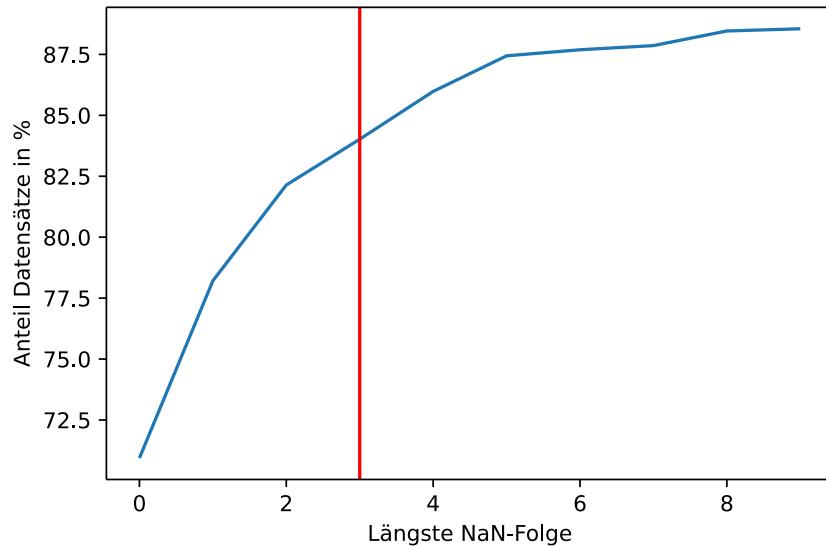


Abbildung 4.2.: Längste NaN-Folge im Handgelenksdatensatz mit betreffender Anzahl an Dateien

### 4.3. Glättung

Bedingt durch Messungenauigkeiten oder falsche Kalibrierungen können in den Datensätzen Ausreißer oder Rauschen auftreten. Für die Berechnung der Geschwindigkeit und der Beschleunigung beruhend auf den Ortspunkten ermöglicht die Glättung verbesserte Ergebnisse, da schon kleine Ausreißer bei mehrfacher Differenzierung große Auswirkungen verursachen können.

Eine Glättung der Daten und damit eine Befreiung der Datensätze von Ausreißern ist notwendig, um im Laufe der weiteren Verarbeitung der Daten eine möglichst fehlerfreie Grundlage zur Verfügung zu stellen. Um die Datensätze von Rauschen zu befreien, bieten sich diverse Methoden zur Filterung an: Der Savitzgy-Golay Filter, der Smirnov-Grubbs Filter und der gleitende Durchschnitt, darunter der Simple Moving Average oder der Moving Median.

Bei näherer Betrachtung der Datensätze werden verschiedene Arten von Messungenauigkeiten deutlich:

- Versatz einer Datenfolge
- Ausreißer
- Rauschen

Der **Versatz von Datenfolgen** um einen bestimmten Wert tritt vereinzelt in Datensätzen auf. Dabei verschiebt sich die Trajektorie der Bewegung zu einem bestimmten Zeitpunkt im Raum und führt sich dort weiter fort. Teilweise verschiebt sich die Trajektorie ein paar Datensätze später auf die zu diesem Zeitpunkt erwartete ursprüngliche Position zurück. Eine Reparatur der betroffenen Datensätze gestaltet sich aufwendig, da sich der Versatz und die Länge dieser verschobenen Datenfolge jeweils unterscheidet. Aufgrund der geringen Anzahl von Datensätzen, die diesen Versatz während des Bewegungsvorgangs aufweisen, und der aufwendigen Reparatur dieser Daten, wird auf eine Anpassung dieser Art von Messungenauigkeiten verzichtet und im späteren Verlauf der Pipeline entsprechend behandelt.

Große **Ausreißer** in Form von Datenpunkten weit außerhalb des erwarteten Bewegungsverlaufs können mithilfe des Smirnov-Grubbs Filters erkannt werden. Dieser Filter findet in diesem Schritt der Pipeline keine Verwendung, da große Ausreißer im weiteren Verlauf der Datenvorverarbeitung als Anhaltspunkt dienen, um stark beschädigte Datensätze auszusortieren.

**Rauschen**, das sich durch kleine Ausreißer zeigt, kann durch eine Glättung basierend auf dem Mittelwert beseitigt werden. Dafür stehen mehrere Methoden zur Verfügung, die im Folgenden auf die Qualität ihrer Ergebnisse untersucht werden sollen. Durch das gleitende arithmetische Mittel (Simple Moving Average) werden kleine Ausreißer in den Datensätzen ausgeglichen. Ein Nachteil ist der Verzug der geglätteten Daten in zeitlicher Richtung nach hinten abhängig von der Fenstergröße. Dieser Verzug entsteht durch die Berücksichtigung vergangener Datenpunkte. Ein weiterer Nachteil ist der Verlust von Randdaten zu Beginn und am Ende des Datensatzes in der Größe des verwendeten Fensters.

Eine Möglichkeit zur Vermeidung von Verschiebungen ist der zentrale Durchschnitt mit einer gewissen Fensterbreite. Mit diesem werden Werte vor und nach dem Datenpunkt

für die Berechnung des Mittelwertes herangezogen. Für die aufgenommenen Messdaten stellt sich eine Fensterbreite von sieben Werten als effizient heraus. Bei dieser Methode funktioniert die Glättung sehr gut in Anbetracht der lokal naheliegendsten Werte, ohne dass dabei markante Spitzen wie die Maximalgeschwindigkeit zu stark beeinflusst werden. Durch das Fenster entsteht allerdings ein Verlust von Datenpunkten zu Beginn und am Ende der Datensätze von jeweils drei Werten.

Für die Inklusion der Randwerte eignet sich der gleitende Durchschnitt mit dynamischem Fenster. Dieser ist allerdings für die vorliegenden Daten ungeeignet, da sich durch den Algorithmus einige Durchschnittswerte doppeln können, sobald sich das Fenster anpasst und dieselben Werte für die Durchschnittsberechnung verschiedener Datenpunkte verwendet werden.

Der Ansatz des Moving Median als Durchschnittsberechnung eignet sich vor allem für laplaceverteilte Datensätze [30]. Für die vorliegenden Daten sind die Verbesserungen der Qualität durch diese Methode unerheblich und sie wird damit verworfen.

Eine in der Literatur verwendeten Filtermethoden ist der Savitzky-Golay Filter [19]. Dieser Filter basiert auf polynomialer Regression anhand von Stützstellen. Im Vergleich auf Grundlage der aufgenommenen Daten zeigt sich, dass eine Regression auf Basis eines Polynoms dritten Grades und 13 Stützstellen eine zufriedenstellende Annäherung an eine von lokalen Ausreißern geglättete Trajektorie ergibt. Nachteile wie der Verlust von Datenpunkten zu Beginn und am Ende des Datensatzes oder eine Einebnung von markanten Stellen ergeben sich hier nicht. Dieser Algorithmus stellt damit das geeignetste Mittel zur Glättung der Datensätze dar und findet in diesem Schritt der Pipeline Verwendung. Ein direkter Vergleich der infrage kommenden Methoden zur Glättung sind an einem beispielhaften Datensatz in Abbildung 4.3 abgebildet. In diesem Beispiel werden die Auswirkungen auf den Geschwindigkeitsverlauf der Trajektorie dargestellt. Es ist zu sehen, dass der Moving Median mit dem Rohdatensatz übereinstimmt, der Savitzky-Golay Filter in rot und der zentrale Durchschnitt in grün hingegen eine zufriedenstellende Glättung erzeugt. Im unteren Teil der Abbildung sind die Geschwindigkeitsverläufe nach Bearbeitung mit verschiedenen Fenstergrößen des Savitzky-Golay Filters gezeichnet.

In Abbildung 4.4 ist die Glättung des Datensatzes bei mehreren Trajektorien erkennbar. Ein gutes Beispiel ist der linke Ellbogen (grün): Auf der linken Hälfte ist die ungeglättete Trajektorie abgebildet und auf der rechten Hälfte derselbe Datensatz nach dem Durch-

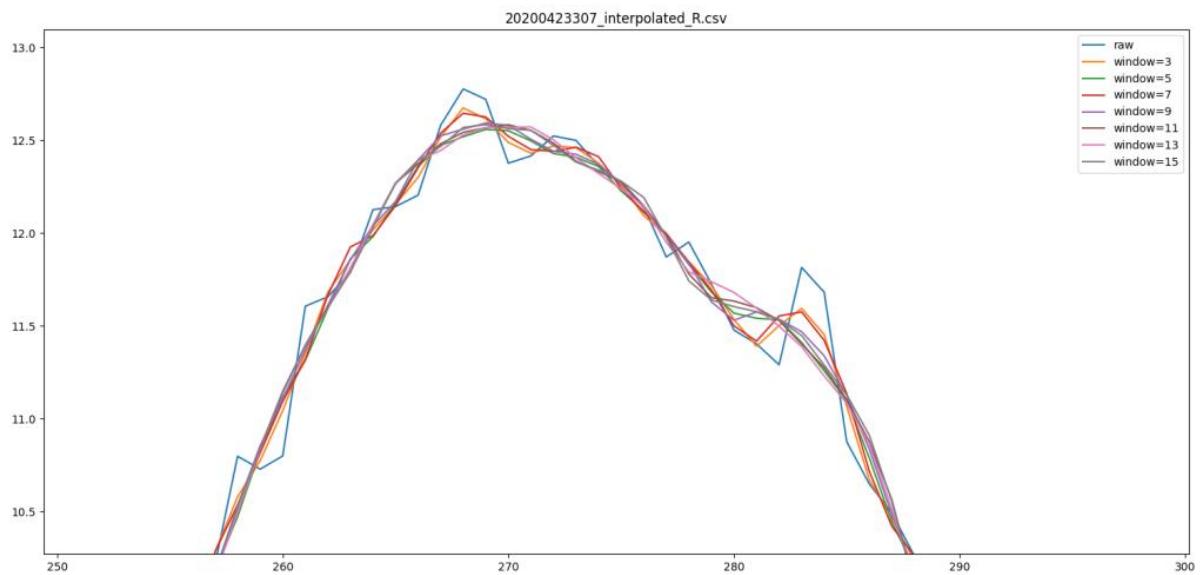
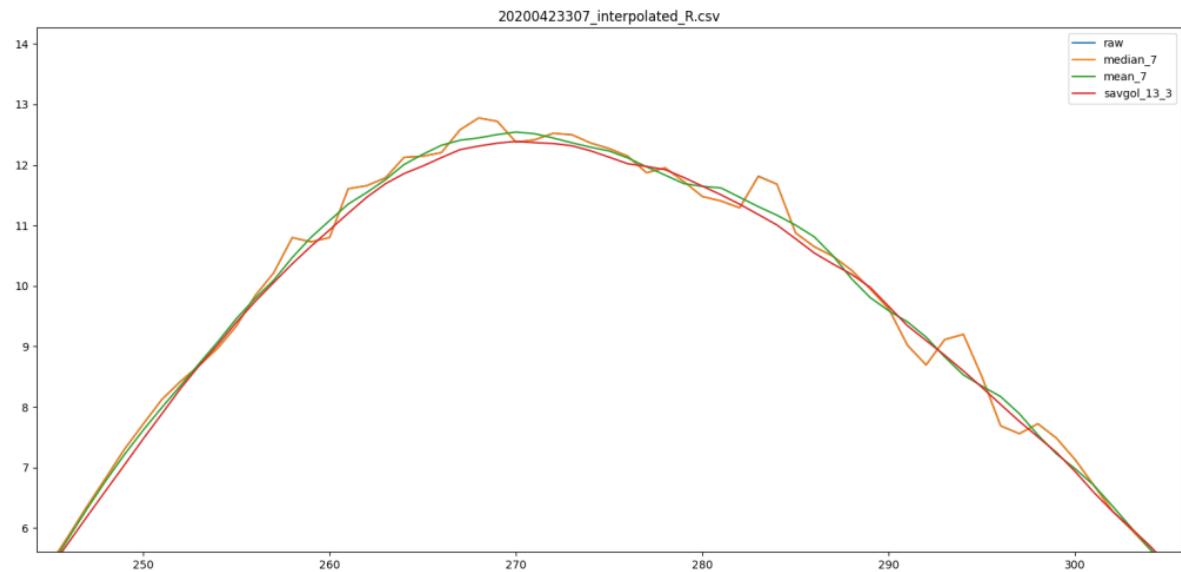


Abbildung 4.3.: Vergleich der verschiedenen Glättungsmethoden und -parameter anhand des Geschwindigkeitsverlaufs der Handgelenktrajektorie

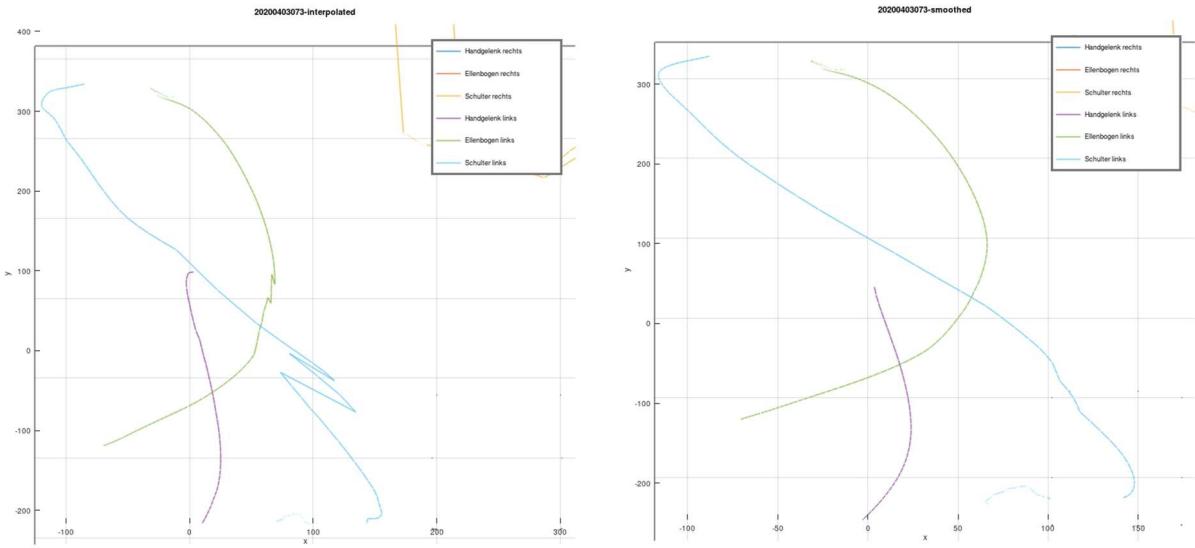


Abbildung 4.4.: Beispielhafte Trajektorien als Vergleich für vor (linke Grafik) und nach (rechte Grafik) dem Bearbeiten des Datensatzes mit dem Glättungsalgorithmus

laufen des Savitzky-Golay Filters. Der extreme Fall des linken Handgelenks (blau) ist im linken Teil der Abbildung deutlich verrauscht; nach der Glättung im rechten Teil der Abbildung sind diese Ausreißer eliminiert.

## 4.4. Beschneidung

Die Rohdaten der Armbewegungen bestehen aus unterschiedlich langen Datensätzen, die grob zugeschnitten vorliegen. Die Datensätze enthalten einen Abschnitt zu Beginn, in dem sich die Position der Handgelenke, der Ellenbogengelenke und der Schultergelenke kaum verändern. Dieser Zustand spiegelt das lose Hängen der Arme und Hände der Testperson vor der eigentlichen Greifbewegung wider. Darauf folgend beginnt die Phase des Greifens, in der sich die Position der jeweiligen Gelenke verändert. Da für die Auswertung des Greifvorgangs die Daten, in welchen keine Bewegung stattfindet, also sich die Koordinaten nicht signifikant ändern, irrelevant sind, sollen diese Daten aus den Datensätzen ausgeschnitten werden. Der Grundgedanke besteht hierbei darin, dass über das Geschwindigkeitsprofil der Trajektorie der Zeitraum des Greifvorgangs ermittelt werden kann.

Die Beschneidung besteht aus folgenden Schritten:

1. Ermittlung der Maximalgeschwindigkeit
2. Einführung eines Schwellwerts
3. Bestimmung der Indizes des Start- und Endzeitpunkts
4. Beschneiden des Datensatzes
5. Überprüfung der Qualität des zugeschnittenen Datensatzes
6. Gegebenenfalls Anpassung des Schwellwerts

Angelehnt an das Vorgehen von Zhao u. a. [19] soll zunächst ein Schwellwert  $\kappa_v$  auf Basis der Maximalgeschwindigkeit der Trajektorie bestimmt werden, der als Stillstand der Bewegung angenommen wird. Ausgehend vom Punkt der höchsten Geschwindigkeit kann anschließend der nächste Punkt vor und nach diesem Zeitpunkt ermittelt werden, zu dem die nach dem Schwellwert als Stillstand angenommene Bewegung auftritt. Damit kann im Idealfall der Datensatz soweit beschnitten werden, dass der erste bzw. der letzte Punkt des Datensatzes den Beginn bzw. das Ende des Greifvorgangs darstellt. Ob dieser Zustand nach der Betrachtung mit dem zu Beginn festgelegten Schwellwert erreicht ist, soll über eine Metrik ermittelt werden. Bei unzureichender Qualität des zugeschnittenen Datensatzes soll eine weitere Beschneidung mit höherem Schwellwert erfolgen.

Zu Beachten ist hierbei, dass die Trajektorien der Handgelenke, der Ellenbogen und der Schultern nur nach Geschwindigkeitsinformationen der Handgelenke beschnitten werden. Dies folgt der Annahme, dass sich an den Handgelenken am klarsten festmachen lässt, wann der Greifvorgang startet und endet.

#### **4.4.1. Ermittlung der Maximalgeschwindigkeit**

Für die Bestimmung der Maximalgeschwindigkeit der Trajektorie muss zunächst die Geschwindigkeit der gesamten Bewegung berechnet werden. Die Datenpunkte als Funktion der Ortskoordinaten in Abhängigkeit von der Zeit ermöglichen eine Ermittlung der Geschwindigkeit durch einmaliges Differenzieren nach der Zeit. Für die Bestimmung der ersten Ableitung einer Funktion diskreter Werte bietet sich ein Verfahren zum numerischen Differenzieren an. Für äquidistante Stützwerte eignet sich der Einsatz der allgemeinen

Gleichung (4.1). Die Gewichte  $w_k$  ergeben sich aus dem Gewichtsvektor beim Lösen der Least-Squares Gleichung [31].

$$\hat{f}_k^{(n)} = \frac{\sum_{i=-p}^q w_{k+i} f_{k+i}}{h^n} \quad \text{mit} \quad f_{k+i} := f(x_k + ih) \quad (4.1)$$

Ein Vergleich der Ergebnisse der Ableitung von drei mit fünf Stützwerten ergibt eine Standardabweichung der Werte von  $s = 0.03$ . Die Nutzung von mehr als drei Stützstellen für die vorliegenden Daten verbessert folglich die Qualität des Ergebnisses nicht wesentlich.

Für die zentrale Differenz  $p = q$  und die erste Ableitung  $n = 1$  ergibt sich bei einer Schrittweite von  $h = 1$  folgende Gleichung (vgl. Gleichung (4.2)). Das heißt, dass für die Berechnung der numerischen Ableitung jedes diskreten Wertes die Hälfte der Differenz des unmittelbar vorangegangenen und nachfolgenden Wertes genutzt wird. An den Randpunkten wird jeweils eine Vorwärts- bzw. eine Rückwärtsdifferenzierung durchgeführt.

$$\hat{f}_k^{(1)} = \frac{f(x_{i+1}) - f(x_{i-1})}{2} \quad (4.2)$$

Die Maximalgeschwindigkeit in jeder Raumdimension ergibt sich als der betragsmäßig größte Wert. Nach der Berechnung der Geschwindigkeiten in die jeweilige Achsenrichtung und damit des Gradienten kann die Geschwindigkeit entlang der Trajektorie bestimmt werden. Dafür wird die euklidische Norm des Gradienten berechnet. Der Zeitpunkt der maximalen Geschwindigkeit des Bewegungsvorgangs entspricht dem (folglich betragsmäßig) größtem Wert des Geschwindigkeitsdatensatzes.

Um fälschlicherweise ermittelte Maximalgeschwindigkeiten aufgrund von großen Ausreißern im Datensatz zu vermeiden, wird der in Abschnitt 4.3 erwähnte Smirnov-Grubbs Filter auf den Ortsdaten temporär angewendet. Dieser Filter basiert auf der Annahme, dass der Datensatz normalverteilt vorliegt und ermittelt diejenigen Punkte, die sich außerhalb einer definierten Signifikanz befinden [21]. Damit kann der Datensatz von durch Ausreißer verursachten zu großen Geschwindigkeiten entfernt werden. Die Geschwindigkeitsdaten der vorliegenden Datengrundlage kann als hinreichend normalverteilt betrachtet werden. Im Anwendungsfall dieser Arbeit genügt eine Eliminierung

von Geschwindigkeiten außerhalb einer Signifikanz von  $\alpha = 0.01$  für eine ausreichend gute Entfernung von Ausreißern. Die Entfernung der Ausreißer erfolgt ausschließlich für die Berechnung der Geschwindigkeit. Die Ortsdaten sollen von diesem Schritt nicht beeinflusst werden, da für die spätere Ermittlung der Qualität der Datensätze die Existenz und Intensität von Ausreißern als Metrik genutzt wird.

#### 4.4.2. Einführung eines Schwellwerts

Der Schwellwert  $\kappa_v$  dient dazu, eine Definition für den Stillstand der Bewegung zu treffen. Dies ist notwendig, da durch Messungenauigkeiten oder natürlich bedingte leichte Bewegungen der Hände in Ruhepositionen auch außerhalb der Greifbewegung niedrige Geschwindigkeiten gemessen werden. Um den Schwellwert vom Datensatz abhängig zu machen, soll ein relativer Bruchteil der Maximalgeschwindigkeit herangezogen werden. Alle Geschwindigkeitswerte unterhalb von  $\kappa_v$  werden als Geschwindigkeit null angenommen, wobei der Schwellwert für eine höhere Präzision des Ergebnisses für jede Dimension separat angewandt wird. Dieser Ansatz verhindert, dass im Falle von Bewegungen in nur einer Dimension eine Beschneidung stattfindet. Denn bei gemeinsamer Betrachtung aller Dimensionen und Bewegung in nur einer Dimension kann der Geschwindigkeitswert unterhalb des Schwellwerts liegen, obwohl die Bewegung Teil des Greifvorgangs ist.

Das Ziel der Wahl des Schwellwerts besteht darin, den  $\kappa_v$  zum einen so niedrig zu wählen, dass keine Abschnitte aus dem Datensatz entfernt werden, die einen markanten Teil der Bewegung definieren. Zum anderen ist der Schwellwert so hoch zu wählen, dass der erste Datenpunkt des beschnittenen Datensatzes den Beginn der Bewegung markiert. Dafür wird der initiale Schwellwert auf das 0.01-fache der Maximalgeschwindigkeit gesetzt. Um zu Ermitteln, ob der Schwellwert auch ausreichend hoch angesetzt wurde, muss eine nachträgliche Bewertung der vorgenommenen Beschneidung anhand einer Metrik durchgeführt werden. Abschnitt 4.4.4 thematisiert diese Methodik.

#### 4.4.3. Bestimmung der Indizes des Start- und Endzeitpunkts

Um den Index von jeweils Start- und Endpunkt der Bewegung festzustellen, wird von der Annahme ausgegangen, dass der Greifvorgang mit durchgehend bewegten Handgelenken stattfindet (vgl. Gleichung (4.3)). Die Bewegung ist damit über das Intervall definiert,

in dem die Punkte mit den Maximalgeschwindigkeiten  $x_{v_{max}}$  der Dimensionen enthalten (4.5) sind und in denen der Betrag des Gradienten größer oder gleich dem Schwellwert  $\kappa_v$  ist (4.6).

$$v_x = \left| \begin{pmatrix} \frac{\partial f}{\partial x_1} & \frac{\partial f}{\partial x_2} & \frac{\partial f}{\partial x_3} \end{pmatrix}^T \right| \quad (4.3)$$

$$v_{x_{start}} = v_{x_{end}} = \kappa_v \quad (4.4)$$

$$x_{v_{max,x_1}}, x_{v_{max,x_2}}, x_{v_{max,x_3}} \in [x_{start}, x_{end}] \quad (4.5)$$

$$\forall x \in [x_{start}, x_{end}] : v_x \geq \kappa_v \quad (4.6)$$

Umsetzbar ist diese Anforderung, indem von den einzelnen Punkten der Maximalgeschwindigkeiten zuerst in Richtung des Startindex und darauf in Richtung des letzten Index des Datensatzes iteriert wird. Bei diesen beiden Durchgängen muss für jeden Wert geprüft werden, ob dieser unterhalb des Schwellwerts  $\kappa_v$  liegt. Ist dies der Fall, ist der Index dieses Wertes der Start- bzw. Endpunkt der Bewegung. Nach der Ermittlung der Start- und Endpunkte in alle Raumdimensionen muss anschließend der kleinste bzw. größte der Werte als Begrenzung des Bewegungsdatensatzes herangezogen und dementsprechend zugeschnitten werden.

Eine weitere Möglichkeit zur Beschneidung besteht in der Betrachtung der Änderung der Geschwindigkeit und damit der Nutzung der zweiten Ableitung. Physikalisch entspricht diese der Beschleunigung der Handbewegung. Erreicht die diese einen über einen Schwellwert definierten Nullpunkt, kann die Bewegung zu diesem Zeitpunkt abgeschnitten werden. Der Vorteil, der sich mit dieser Methode ergibt, ist die deutliche Abtrennung der eigentlichen Greifbewegung von willkürlichen Handbewegungen, die vor dem Greifvorgang stattfinden können und teilweise in diesen übergehen. Dadurch kann die Handbewegung des eigentlichen Greifvorgangs auch mit einer Startgeschwindigkeit  $v_0 > 0$  beginnen. Durch eine Zuschneidung, die nur über die Ruhelage der Hände bestimmt wird, könnten so zu Beginn des Datensatzes pendelnde Hände enthalten sein. Diese würden über die Beschneidung mittels Beschleunigung präziser beschnitten werden. Der Nachteil dieser Methode ist, dass nicht zwischen Ruhezustand und gleichförmiger Bewegung unterschieden werden kann. Damit müsste ein weiterer Schwellwert definiert werden, der eine Geschwindigkeit bestimmt, die niedrig genug ist, dass sie bei niedriger Beschleunigung als Ruhezustand angenommen werden kann.

Im Vergleich mit der Methode des Geschwindigkeitsschwellwerts ergibt sich durch die Methode des Beschleunigungsschwellwerts keine qualitative Verbesserung der Beschneidung der Datensätze (vgl. Tabelle 4.1). Die durchschnittliche Länge der beschnittenen Datensätze ist marginal höher. Damit kann bei der Beschneidung der Datensätze auf die Betrachtung der Beschleunigung verzichtet werden.

Tabelle 4.1.: Vergleich der beiden Methoden zur Beschneidung anhand der Dateilängen von 1170 Datensätzen

Methoden	Minimale Länge	Maximale Länge	Durchschnittslänge
Geschwindigkeit	7	215	130.05
Beschleunigung	7	215	130.48

#### 4.4.4. Überprüfung der Qualität

Nach der Beschneidung mithilfe des initialen Schwellwerts besteht die Möglichkeit, dass einige Datensätze weiterhin längere Datensequenzen enthalten, in denen sich die Position der Handgelenke nicht ändert. Um diese durch Schwanken der Hände verursachte Auslenkung aus einer Ruhelage ebenfalls zu entfernen, empfiehlt es sich, den Schwellwert  $\kappa_v$  für die als Ruhelage definierte Geschwindigkeit höher zu setzen. Für die Pipeline bedeutet das, dass der gesamte Beschneidungsvorgang für den oder die betroffenen Datensätze mit einem erhöhten Schwellwert erneut durchlaufen werden muss.

Dieses Vorgehen muss so lange wiederholt werden, bis über eine Metrik bestimmt wird, dass der Datensatz gut genug beschnitten ist. Anforderung an diese Metrik ist, dass sie keinen Bezug zu den Positionsdaten oder einer Ableitung der Positionsdaten hat, da diese bereits genutzt werden, um die Beschneidung durchzuführen. Eine naheliegende Metrik ist die Länge des beschnittenen Datensatzes: Unterschreitet diese eine gewisse Schranke, wird der Datensatz als ausreichend beschnitten klassifiziert. Nachteile dieser Metrik sind, dass ein Datensatz bei Greifvorgängen, die kürzer sind als die Metrik prüft, nicht gut genug beschnitten wird. Im gegenteiligen Fall eines Greifvorgangs, der länger als die zu prüfende Länge ist, würden Teile des Greifens bei der Beschneidung entfernt werden.

Um letztere Problematik zu umgehen, muss eine obere Grenze für die Erhöhung des Schwellwerts festgelegt werden. Damit ergeben sich folgende Parameter, die für eine gute Qualität der Beschneidung optimiert werden müssen:

- Erhöhungsschritt des Geschwindigkeitsschwellwerts zur Beschneidung
- Maximaler Geschwindigkeitsschwellwert zur Beschneidung
- Prüfmetrik: Schranke für die Länge der Datensätze

Für eine möglichst granulare Abstufung des Schwellwerts, erscheint eine Erhöhung pro Prüfschritt um einen Prozentpunkt sinnvoll. Damit kann garantiert werden, dass die niedrigst mögliche Geschwindigkeit als Stillstand angenommen wird, solange bis der Datensatz die Mindestlänge erreicht hat. In der Abbildung A.1 auf Seite XIII im Anhang finden sich Vergleiche mit verschiedenen Werten der übrigen Parameter und Boxplots über die entsprechenden Datensatzlängen. Es gestaltet sich schwierig, anhand von statistischen Informationen abzulesen, welcher dieser Parameter die beste Qualität der Datensätze hervorbringt. Es zeigt sich aus manueller Analyse der einzelnen Datensätze, dass der Maximalwert des Geschwindigkeitsschwellwerts das 0.3-fache der Maximalgeschwindigkeit nicht unterschreiten sollte und eine Schranke von 150 Datenpunkten als Qualitätsmetrik die besten Ergebnisse liefern. Für die manuelle Analyse werden stichprobenartig zufällige Datensätze und Dateien mit auffälligen Werten betrachtet.

## 4.5. Präfilterung

Die Phase der Präfilterung sortiert anhand einfacher Kriterien und statischer Schwellwerte fehlerbehaftete Messungen aus. Zunächst soll die Wahl des Zeitpunktes in der Verarbeitungspipeline begründet werden.

Die Präfilterung findet getrennt von der späteren metrikbasierten Filterung mit dynamischen Schwellwerten (siehe Abschnitt 4.8) statt, um so früh wie möglich Daten mit eindeutigen, unkorrigierbaren Messfehlern auszusortieren. Dies ist vor der Beschneidung der Daten nicht möglich, da die Messfehler ggf. in abgeschnittenen Zeitintervallen liegen und so ein vermeintlich fehlerbehafteter Datensatz nach Beschneidung nicht mehr fehlerbehaftet ist. Somit ist die vierte Verarbeitungsphase der frühestmögliche Zeitpunkt zur statischen Filterung. Die Zusammenlegung von den Phasen der Präfilterung und Filterung ist nicht zielführend, da durch die frühe Präfilterung bereits Rechenaufwand für die Spiegelung gespart werden kann und die metrikbasierte Filterung auf die Spiegelung und Verschiebung als Vorverarbeitung angewiesen ist.

Als intuitive Kriterien für qualitativ hochwertige Daten lassen sich die **Dateilänge**, **Anzahl an NaN-Werten** im Handgelenk und **Maximalgeschwindigkeit** im Handgelenk heranziehen. Um im Nachhinein transparent feststellen zu können, welche Dateien aus welchem Grund in dieser Phase eliminiert werden, schreibt das verwendete Python-Skript alle aussortierten Dateien in einen separaten Ordner und ergänzt in die Kopfzeile der CSV-Dateien die Ursache der Aussortierung. Im Folgenden soll auf die Festlegung der statischen Schwellwerte für die bereits genannten Kriterien eingegangen werden.

Das generelle Ziel der Filterung anhand der **Dateilänge** besteht darin, fehlerhaft beschnittene oder aufgenommene Datensätze auszusortieren, da diese dadurch nicht mit den anderen Daten vergleichbar sind. Fehlerhafte Beschneidung fällt vor allem durch eine untypische Dateilänge direkt auf. Die Mindestdateilänge ergründet sich in der Definition der Anfangsbewegung über die ersten 85 Datenpunkte in Abschnitt 3.2 auf Seite 15 und liegt daher bei 85. Möglich wäre auch, die Mindestdateilänge auf 100 Zeitpunkte festzulegen, da Datensätze mit genau 85 Zeitpunkten nur die Anfangsbewegung enthalten und keinen Mehrwert zum Training des Neuronalen Netzes bieten. Dagegen spricht allerdings, dass auch Datensätze mit einer Länge von 85 für die metrikbasierte Filterung von Nutzen sind. Denn diese verwendet teilweise falls vorhanden die Messdaten beider Körperhälften (siehe Abschnitt 4.8). Je mehr Messungen mit Datensätzen beider Körperhälften vorhanden sind, desto besser funktioniert die spätere Filterung. Daher sollten aufgrund der Dateilänge für das Maschinelle Lernen nicht sinnvolle Datensätze nicht zu früh aussortiert werden. Dies kann besser direkt beim Einlesen der Trainingsdaten vor dem Training des Neuronalen Netzes durchgeführt werden (siehe dazu auf Seite 59).

Die maximale Dateilänge orientiert sich an den Beobachtungen durchschnittlicher Dateilängen korrekt zugeschnittener Daten, deren Verteilung in Abbildung 4.5 als Histogramm dargestellt ist. Aus der Verteilung der Dateilängen ergibt sich als sinnvolle Maximallänge 150, da dies in der Beschneidung als Maximalwert verwendet wird (siehe dazu Abschnitt 4.4 auf Seite 25).

Beim Kriterium der **Anzahl an NaN-Werte** besteht die Anforderung für Handgelenkdaten darin, keine NaN-Werte mehr nach der Beschneidung aufzuweisen. Nach Interpolation und Beschneidung noch existierende NaN-Werte weisen auf nicht korrigierbare Messfehler hin. Die Daten sind für die weitere Verarbeitung unbrauchbar, da sie keine durchgehenden Trajektorien darstellen und die Lücken die maximale Interpolationslänge (siehe Abschnitt 4.2 auf Seite 20) übersteigen. NaN-Werte würden vor allem die dynamische

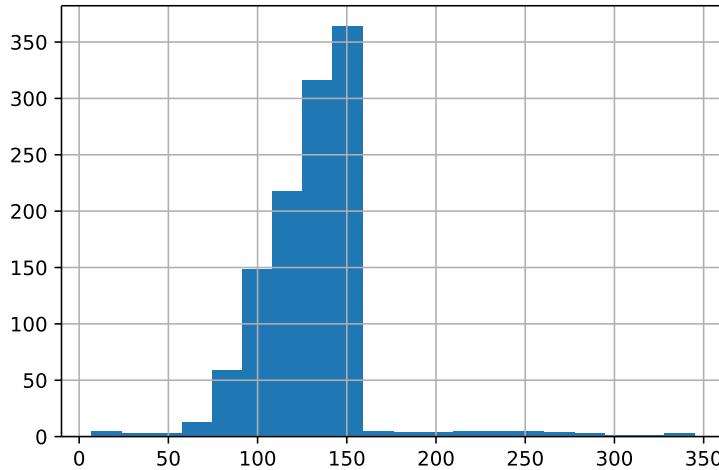


Abbildung 4.5.: Histogramm über Dateilängen vor Präfilterung

Schwellwertbildung in der metrikbasierten Filterung (siehe Abschnitt 4.8 auf Seite 42) signifikant verzerren.

Zur Bestimmung der **Maximalgeschwindigkeit** der Handbewegung müssen zunächst die Geschwindigkeitsverläufe je Dimension, also für x-, y- und z-Richtung ermittelt werden. Diese ergeben sich aus dem Gradienten in entsprechende Richtung, die Geschwindigkeit der Trajektorie aus der euklidischen Norm über die Geschwindigkeiten der drei Dimensionen. Anatomisch betrachtet lässt sich eine realistische Maximalgeschwindigkeit von 5 m/s festlegen, höhere Maximalgeschwindigkeiten weisen auf Messfehler hin. Diese Datensätze können daher ebenfalls aussortiert werden.

## 4.6. Spiegelung

In den vorliegenden Datensätzen sind für einige Greifvorgänge nur jeweils eine Körperhälfte vorhanden. Für eine Vereinheitlichung und Vergrößerung der Daten sollen alle Datensätze auf den Greifvorgang einer Körperhälfte abgebildet werden. In den Datensätzen sind mehr rechte Körperhälften vorhanden, daher ist es sinnvoll, die Datensätze mit den linken Seiten auf die der rechten zu projizieren. Da die Greifvorgänge von jeweils rechtem und linkem Arm als gespiegelt angenommen werden können, bietet sich eine Spiegelung der Greifvorgänge der linken Körperhälften an. Damit sollten anschließend alle Körperhälften dieselbe Greifrichtung und eine näherungsweise gleichartige Trajektorie aufweisen.

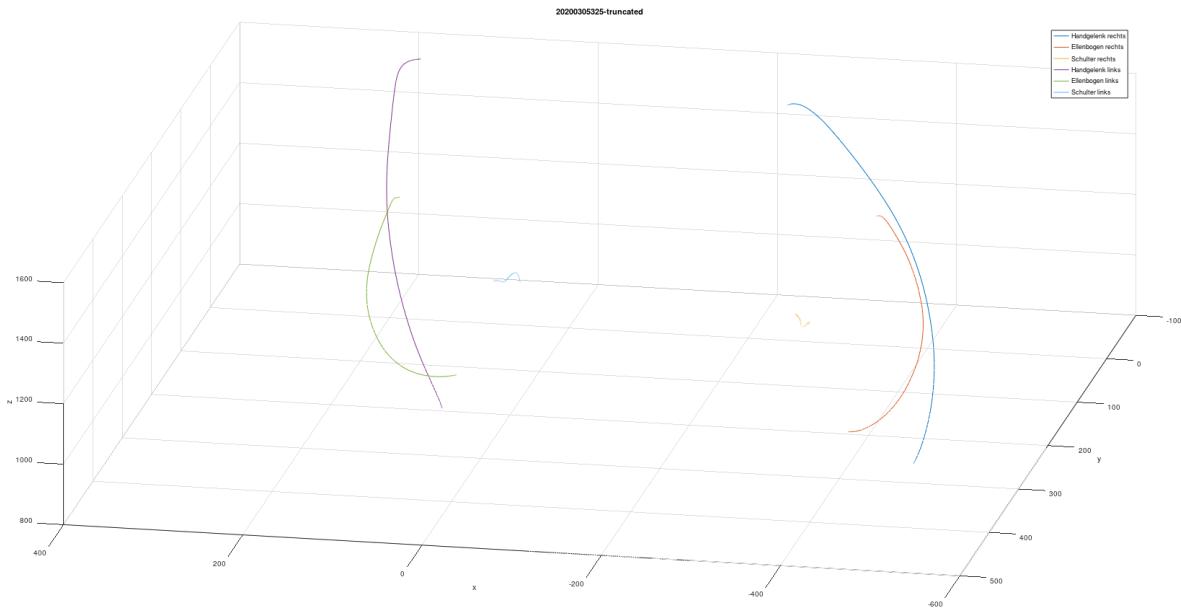


Abbildung 4.6.: Trajektorie der rechten und linken Körperhälfte mit Sicht von oben, Handgelenke in violett und dunkelblau. Der Greifvorgang findet zur linken oberen Bildmitte statt. Erkennbar ist auf beiden Körperhälften ein leichter Bauch in der Bewegung.

Auffällig ist bei den Greifvorgängen, dass der Griff zum Zielobjekt nicht exakt geradlinig erfolgt, sondern ein bauchartiger Bewegungsablauf von der Körpermitte weg stattfindet (vgl. Abbildung 4.6). Bei den Datensätzen der rechten Körperhälfte verläuft dieser leicht nach rechts; bei denen der linken Körperhälfte entsprechend gespiegelt dazu nach links. Ziel soll es nun sein, dass diese Krümmung bei allen Datensätzen in dieselbe Richtung verläuft.

Für die Spiegelung der Trajektorie werden drei Ansätze auf ihre Qualität überprüft. Die erste Möglichkeit ist die Spiegelung an einer Strecke zwischen Start- und Endpunkt des Bewegungsvorgangs. Die zweite ist die Spiegelung an einer Ebene, die zwischen Start- und Endpunkt der Bewegung aufgespannt wird. Für die dritte Möglichkeit wird eine Spiegelebene in der Mitte zwischen beiden Körperhälften aufgespannt.

### 4.6.1. Spiegelung an Strecke

Die Spiegelung an der Strecke zwischen Startpunkt  $P_1$  und Endpunkt  $P_2$  der Bewegung nutzt eine Punktspiegelung für jeden Punkt der Trajektorie an der Stelle der Spiegelgeraden, die ihm am nächsten ist. Damit entspricht die gespiegelte Trajektorie in etwa dem räumlichen Verlauf der Trajektorie der anderen Körperhälfte. Die Spiegelung an einer Geraden ist identisch zu der Spiegelung an zwei Ebenen und damit einer Drehung um  $180^\circ$ . Ein Nachteil ist, dass die Krümmung nicht allein in die erwartete Richtung verläuft, sondern auch vertikal gespiegelt vorliegt. Ein weiterer Nachteil ist, dass die relative Position und Ausrichtung der Trajektorien der verschiedenen Gelenke zueinander nach der Spiegelung verändert ist. Diese Möglichkeit soll dennoch betrachtet werden, da der mathematische Ansatz für diese Methode die Grundlage für die anderen beiden Spiegelmethoden liefert.

Die Schwierigkeit ist, den Punkt  $M_s$  auf der Spiegelstrecke  $P_1P_2$  zu finden, der dem zu spiegelnden Punkt  $P_s$  am nächsten liegt. Berechnet werden kann dieser über folgendes Verfahren:

1. Berechnung des Abstands  $d(P_1, P_s)$  von Startpunkt  $P_1$  zum zu spiegelnden Punkt  $P_s$

$$d(P_1, P_s) = \|P_s - P_1\|_2 \quad (4.7)$$

2. Berechnung des Abstands  $d(P_s, M_s)$  von zu spiegelndem Punkt zum Punkt mit der kürzesten Distanz  $M_s$  auf der Spiegelstrecke  $P_1P_2$ . Diese Strecke entspricht der Höhe des Parallelogramms, das durch die Spiegelstrecke und die Strecke  $P_1P_s$  aufgespannt wird. Eine Berechnung der Flächeninhalts  $A$  des Parallelogramms ist über das Kreuzprodukt möglich. Mittels Division durch Länge der Strecke  $P_1P_2$  ergibt sich die Höhe und damit die Distanz  $d(P_s, M_s)$ . Grafisch ist dieser Schritt in Abbildung 4.7 dargestellt

$$A = |P_1P_2 \times P_1P_s| \quad (4.8)$$

$$d(P_s, M_s) = \frac{A}{d(P_1, P_s)}$$

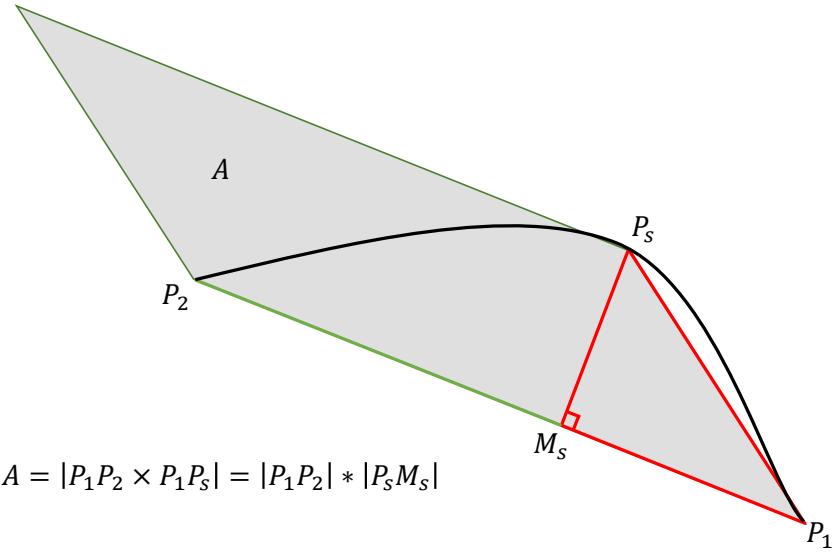


Abbildung 4.7.: Bestimmung der Senkrechten zum zu spiegelnden Punkt  $P_s$  mithilfe eines Parallelogramms. Die Trajektorie ist hier in Schwarz dargestellt.

3. Bestimmung der Distanz vom Startpunkt  $P_1$  zum Punkt  $M_s$  über den Satz des Pythagoras

$$d(P_1, M_s) = \sqrt{d(P_1, P_s)^2 - d(P_s, M_s)^2} \quad (4.9)$$

4. Normalisierung der Spiegelstrecke Start-/Endpunkt  $P_1P_2$  auf Vektor  $\vec{e}$  der Länge  $|\vec{e}| = 1$

$$\vec{e} = \frac{P_2 - P_1}{\|P_2 - P_1\|_2} \quad (4.10)$$

5. Bestimmung der Position des Punkts  $M_s$  über den normierten Vektor  $\vec{e}$  der Strecke  $P_1P_2$

$$M_s = \vec{e} \cdot d(P_1, M_s) \quad (4.11)$$

Nach der Berechnung des Punktes  $M_s$ , der am nächsten zur Spiegelstrecke steht und damit auch orthogonal dazu, ist die Verdopplung der Strecke  $P_sM_s$  vom zu spiegelnden

Punkt  $P_s$  aus notwendig, um Spiegelpunkt  $P'_s$  zu erhalten.

$$P'_s = P_s + 2P_s M_s \quad (4.12)$$

Dieses Verfahren muss für jeden Punkt  $P_s$  der Trajektorie durchgeführt werden, um die Punkte der gespiegelten Trajektorie zu erhalten.

#### 4.6.2. Spiegelung an Ebene

Eine Verbesserung der Methode, die Trajektorie an einer Strecke zu spiegeln, ist die Spiegelung an einer Ebene. Damit kann der Nachteil der Doppelspiegelung vermeiden werden, der durch die Spiegelung an einer Strecke entsteht, wie in Abschnitt 4.6.1 ausgeführt ist.

Die Herangehensweise für die Spiegelung an einer Fläche ist ähnlich zu der Spiegelung an der Strecke: Die Spiegelebene  $E_s$  soll durch den Start- und den Endpunkt des Greifvorgangs ( $P_1$  und  $P_2$ ) verlaufen mit einer Neigung orthogonal zur Ebene des tendenziellen Verlaufs der Trajektorie (vgl. Abbildung 4.8). Die Ebene des Verlaufs  $E_v$  wird vereinfacht angenommen als die Ebene, die sich zwischen Start- und Endpunkt sowie dem Punkt  $P_3$  aufspannt, der am weitesten von der Verbindungsstrecke  $P_1P_2$  entfernt ist. Für die Berechnung der Spiegelebene ist es sinnvoll, eine Gerade heranzuziehen, die in der Ebene  $E_v$  liegt und senkrecht auf die Spiegelebene  $E_s$  steht. Diese Gerade dient als Normalenvektor  $\vec{n}$  der Spiegelebene und verläuft durch den Punkt  $P_3$  und den Punkt  $M_3$  auf der Verbindungsstrecke  $P_1P_2$ , sodass gilt  $P_3M_3 \perp P_1P_2$ . Über die Koordinatengleichung der Spiegelebene lässt sich für jeden Punkt  $P_s$  auf der Trajektorie der entsprechende Spiegelpunkt  $P'_s$  berechnen.

Das Vorgehen für diese Spiegelung ist im Folgenden zusammengefasst:

1. Bestimmung des Normalenvektors  $\vec{n}$  anhand des am weitesten entfernten Punkts  $P_3$  von der Strecke  $P_1P_2$ . Dafür wird der Punkt  $M_3$  auf dieser Strecke gesucht, sodass diese orthogonal zur Strecke  $P_3M_3$  steht. Die Berechnung des Punktes  $M_3$  geschieht nach demselben Verfahren, wie im Vorgehen in Abschnitt 4.6.1 für den Spiegelpunkt

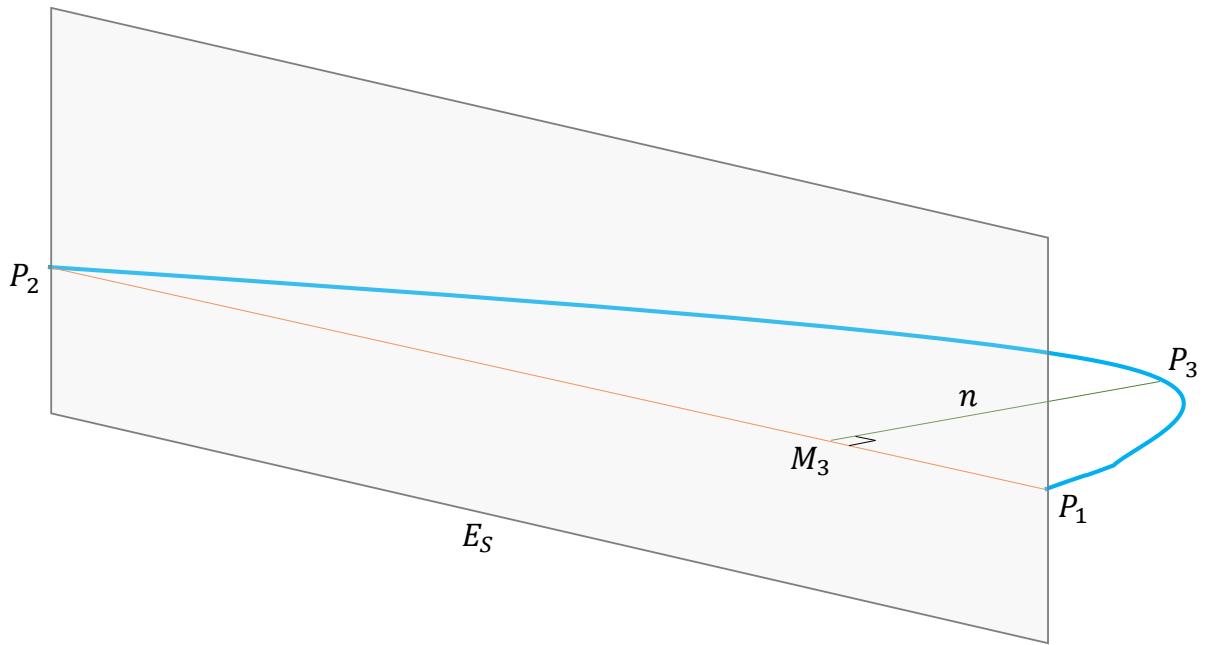


Abbildung 4.8.: Berechnung der Spiegelebene  $E_s$  anhand der Verbindungsstrecke  $P_1P_2$  und des Normalenvektors  $\vec{n}$

$M_s$  beschrieben wird. Anschließend wird diese Strecke normiert.

$$\vec{n} = \frac{M_3 - P_3}{\|M_3 - P_3\|_2} \quad (4.13)$$

2. Ablesen der Koeffizienten  $a$ ,  $b$  und  $c$  der Koordinatengleichung der Spiegelebene über die Parameterdarstellung des Normalenvektors  $\vec{n}$ . Ermittlung von  $d$  über das Skalarprodukt des Stützvektors  $\vec{m}$  zu  $M_3$  und  $\vec{n}$ .

$$E_s = ax_1 + bx_2 + cx_3 + d \quad (4.14)$$

$$\vec{n} = (a \ b \ c)^T$$

$$d = -\vec{m} \cdot \vec{n}$$

3. Wie in Abschnitt 4.6.1 muss nun für den zu spiegelnden Punkt  $P_s$  der Punkt  $M_s$  auf der Spiegelebene berechnet werden, sodass die Strecke  $P_sM_s$  orthogonal zur Spiegelebene steht. Für die Bestimmung von  $M_s$  wird der Schnittpunkt der Strecke und der Ebene ermittelt, indem die Gerade  $p$  durch die Strecke  $P_sM_s$  in

die Ebenengleichung eingesetzt und nach dem Parameter  $t$  aufgelöst wird.

$$\begin{aligned}
 E_s &= ax_1 + bx_2 + cx_3 + d & (4.15) \\
 P_s &= (p_1, p_2, p_3) \\
 M_s &= (m_1, m_2, m_3) \\
 p : \vec{x} &= (p_1 \ p_2 \ p_3)^T + t(a \ b \ c)^T \\
 0 &= a(at + p_1) + b(bt + p_2) + c(ct + p_3) + d \\
 \implies t &= \frac{-(ap_1 + bp_2 + cp_3 + d)}{a^2 + b^2 + c^2}
 \end{aligned}$$

4. Bestimmung des Punkts  $M_s$  und der Spiegelpunkt  $P'_s$  mit dem Parameter  $t$

$$\begin{aligned}
 (m_1 \ m_2 \ m_3)^T &= (p_1 \ p_2 \ p_3)^T + t(a \ b \ c)^T & (4.16) \\
 (p'_1 \ p'_2 \ p'_3)^T &= 2(m_1 \ m_2 \ m_3)^T - (p_1 \ p_2 \ p_3)^T
 \end{aligned}$$

Die Berechnung des Spiegelpunkts muss nun für jeden Punkt  $P_s$  der Trajektorie durchgeführt werden.

### 4.6.3. Spiegelung anhand beider Körperhälften

Ein Nachteil des in Abschnitt 4.6.2 beschriebenen Verfahrens ist die Spiegelung der Trajektorie anhand eines Punktes auf der Trajektorie selbst. Präziser lässt sich die Spiegelachse bestimmen, indem die Bewegungsabläufe beider Körperhälften in Betracht gezogen werden. Eine Herausforderung hierbei ist die unterschiedliche Länge der Datensätze. Eine eventuelle variierende Länge zwischen rechtem und linkem Datensatz gestaltet es schwierig, einen gemeinsamen Start- und Endpunkt zu finden. Zudem funktioniert dieses Verfahren nur unter der Voraussetzung, dass für beide Körperhälften die Datensätze vorliegen.

Der erste Schritt ist somit die Ermittlung, ob je Körperhälfte ein Datensatz vorhanden ist. Ist dies der Fall, müssen diese für die Berechnung der Spiegelachse auf dieselbe Länge getrimmt werden bzw. ein gemeinsamer Start- und Endzeitpunkt ermittelt werden. Diese Beschneidung wird nicht in den Datensätzen persistiert, sondern ist temporär und dient nur zur Berechnung der Spiegelachse. Die Idee ist, die Datensätze anhand ihres

Geschwindigkeitsverlaufs aneinander auszurichten und die jeweils äußersten gemeinsamen Zeitpunkte als Start- und Endpunkt zu definieren. Der Einfachheit wegen sollen nur die Zeitpunkte der höchsten Geschwindigkeiten jeweils in Betracht gezogen werden. Dafür muss die Annahme getroffen werden, dass der Punkt der höchsten Geschwindigkeit bei beiden Körperhälften zur selben Zeit auftritt.

Das Vorgehen ist sehr ähnlich zu dem in Abschnitt 4.6.2 ausgeführten Verfahren. Der Unterschied besteht darin, dass die Strecke zwischen Start- und Endpunkt  $P_1P_2$  nicht auf einer Trajektorie liegt, sondern zwischen beiden Körperhälften.  $P_1$  entspricht dem Mittelpunkt des Startpunkts von rechter und linker Körperhälfte;  $P_2$  dem Mittelpunkt des Endpunkts von rechter und linker Körperhälfte. Damit befindet sich die Spiegelebene in der Mitte der beiden Trajektorien. In Abbildung 4.9 ist die berechnete Spiegelebene in blau zwischen den Bewegungsvorgängen der beiden Handgelenke abgebildet. Die Trajektorien sind reduziert auf die Startpunkte  $P_{1R}$  und  $P_{1L}$  der rechten bzw. der linken Hand, die Endpunkte  $P_{2R}$  und  $P_{2L}$  sowie jeweils deren Mittelpunkte  $P_1$  und  $P_2$  und die Gerade  $P_1P_2$ , die durch diese beiden Punkte führt. Ebenfalls eingezeichnet ist der Punkt der maximalen Geschwindigkeit  $P_{3L}$  und die Punkte  $P_{1M}$ ,  $P_{2M}$  und  $P_{3M}$ , die orthogonal auf der Spiegelebene zu den entsprechenden Punkten des linken Handgelenks stehen.

#### 4.6.4. Verfahren in der Phase des Spiegelns

Das abschließende Vorgehen der Phase des Spiegelns der Trajektorien ist wie folgt: Sobald für einen Greifvorgang Datensätze zu beiden Körperhälften vorliegen, soll die Spiegelebene anhand dieser beiden Datensätze ausgerichtet werden (vgl. Abschnitt 4.6.3). Ist nur eine Körperhälfte vorhanden, soll die Spiegelung an einer Ebene durch Start- und Endpunkt berechnet werden, wie in Abschnitt 4.6.2 beschrieben ist.

Trajektorien von rechter und gespiegelter linker Körperhälfte überschneiden sich häufig nicht, da der Bewegungsablauf bspw. durch Griffe nach links oben nicht exakt spiegelsymmetrisch verlaufen. Der Verlauf im Raum und die Krümmung sind hingegen ähnlich. Durch eine Relokation auf einen einheitlichen Startpunkt können diese Abweichungen wieder vereinheitlicht werden.

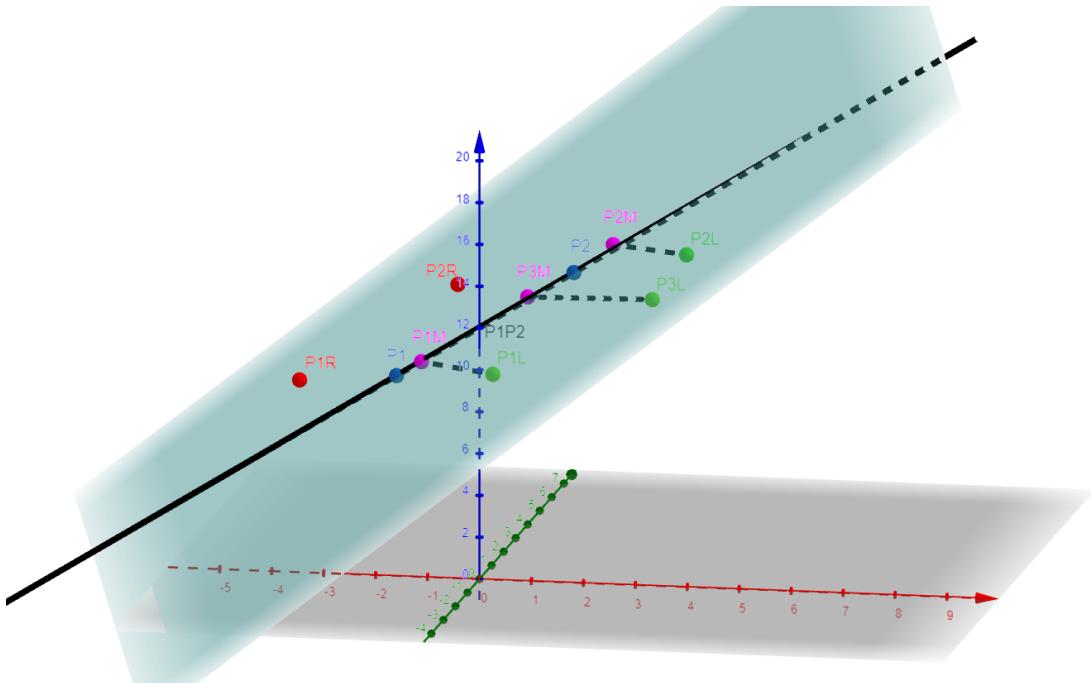


Abbildung 4.9.: Einziehen der Spiegelebene zwischen den relevanten Punkten der Trajektorien beider Handgelenke

## 4.7. Relokation

Die Phase der Relokation ist wie in den Vorüberlegungen in Abschnitt 4.1 beschrieben ein wichtiger Verarbeitungsschritt zur Vereinheitlichung der aufgenommenen Messdaten. Ziel hierbei ist die Verschiebung aller Datensätze im Raum, sodass die Trajektorien der Handgelenke jeweils im Ursprung starten. Dieser Abschnitt begründet die Notwendigkeit dieser Verschiebung.

Die Messdaten stammen von verschiedenen Messtagen mit verschiedenen Messpersonen. Trotz zahlreicher Bemühungen, die Messungen so standardisiert und vergleichbar wie möglich durchzuführen, liegen die Koordinatensysteme der Messergebnisse dennoch leicht zueinander verschoben vor. Dies lässt sich vor allem durch eine Visualisierung der Endpositionen der Bewegungen erkennen, die in gleichen Abständen in einem Würfel angeordnet sein sollten (siehe Kapitel 2.1). Tatsächlich liegen die Würfel der Endpunkte jedes Messtages aber nicht übereinander, sondern sind leicht verschoben wie Abbildung 4.10 auf Seite 42 zeigt. Hier zu sehen ist die nach Messtagen farblich sortierte Anordnung der Endpunkte in einer gitterartigen Struktur, die zur vom Körper der Versuchsperson ent-

fernten Seite sehr diffus wird. Durch die seitliche Perspektive auf die Endpositionswürfel ist ein leichter Versatz der Messstage zueinander erkennbar.

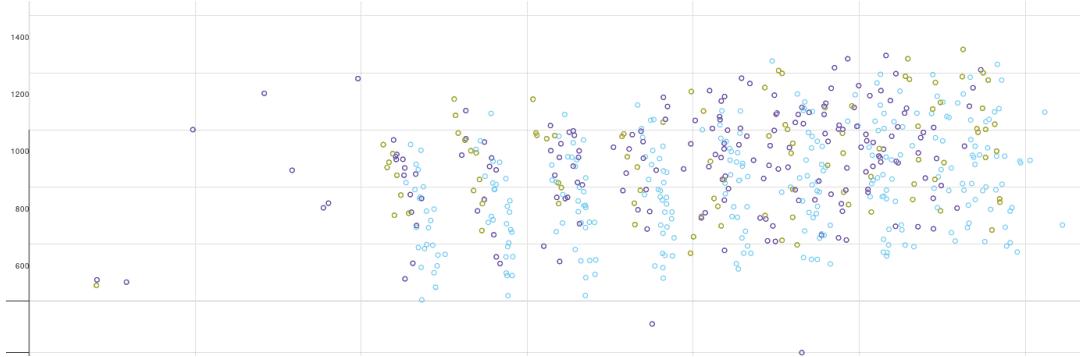


Abbildung 4.10.: Visualisierung der Endpunkte aller Messdaten

Diese fälschliche Verschiebung soll durch die einheitliche Verschiebung aller Handgelenkstrajektorien auf einen gemeinsamen Startpunkt eliminiert werden. Da die Koordinatensysteme der verschiedenen Messungen nicht zueinander rotiert, sondern lediglich verschoben vorliegen, ist eine Translation ausreichend.

Hierbei stellt sich die Frage, ob durch die Translation für das Neuronale Netz die Messdaten verfälscht werden. Dem ist nicht der Fall, da Trajektorienverläufe unabhängig von der absoluten Startposition im Raum betrachtet werden können. Durch die einheitliche Verschiebung gewinnen die Daten an Vergleichbarkeit, da sie in einem gemeinsamen Koordinatensystem vorliegen. Dadurch kann das Netz besser Zusammenhänge lernen.

In der praktischen Anwendung muss der ursprüngliche Startpunkt einer gemessenen Anfangsbewegung gespeichert werden, um eine Retransformation in das ursprüngliche Koordinatensystem nach Vorhersage des Endpunktes im vereinheitlichten Koordinatensystem zu ermöglichen.

## 4.8. Filterung

Diese Phase soll durch systematische Filterung von Datensätzen die Qualität der Messdaten weiter steigern. Zur Feststellung der Qualität und Identifikation von Ausreißern werden verschiedene Metriken eingesetzt, die in Abschnitt 4.8.1 auf Seite 43 vorgestellt werden. Eine Besonderheit hierbei besteht in der dynamischen Ermittlung von Schwellwer-

ten zur Identifikation der Ausreißer, wie bereits in Abschnitt 4.5 erwähnt. Abschnitt 4.8.2 ab Seite 45 befasst sich mit dieser Methodik. Im Anschluss daran erläutert Abschnitt 4.8.3 ab Seite 48 das Verfahren zur Bestimmung der auszusortierenden Menge an Datensätzen - basierend auf den Intensitäten von in Metriken auffälligen Ausreißern.

### 4.8.1. Metriken

Die Filterung wird anhand drei Metriken durchgeführt, deren Wahl und Umsetzung in diesem Abschnitt vorgestellt wird. Generell sortiert keine dieser Metriken auffällige Datensätze direkt aus, sondern markiert diese lediglich als in dieser Metrik auffällig. Die drei Metriken sind ähnlich aufgebaut und ermitteln zunächst für alle Zeitpunkte einen Wert, beispielsweise eine Distanz im Raum. Anschließend ergibt sich der Metrikwert eines Datensatzes durch die Bildung des arithmetischen Mittels aller Werte. Hierbei fällt die Wahl des Lagemaßes auf das arithmetische Mittel, da dieses im Vergleich zum Median nicht robust gegenüber Ausreißern ist [32]. Dadurch spiegeln sich beispielsweise hohe Distanzen zu einzelnen Zeitpunkten eines Datensatzes im Metrikwert des gesamten Datensatzes wider. Die Metrikwerte aller  $d$  Datensätze in Metrik  $m$  werden fortan mit  $\gamma_m \in \mathbb{R}^d$  bezeichnet. Nach Durchlauf aller Metriken erfolgt anhand aller Ergebnisse eine Aussortierung, die in Abschnitt 4.8.3 behandelt wird.

#### Ellenbogen-Handgelenk-Distanz

Der Abstand von Handgelenk und Ellbogen ist durch die feste Länge von Elle und Speiche bei jeder Testperson im Laufe von Bewegungen annähernd konstant. Daher eignet sich dieses Maß gut als Merkmal zur Qualitätsprüfung und Identifizierung fehlerbehafteter Datensätze. Die erste Metrik der Filterung berechnet daher für jeden Zeitpunkt  $i$  eines Datensatzes die euklidische Distanz der Koordinaten von Ellbogen und Handgelenk. Der Metrikwert der Ellbogen-Handgelenk-Metrik des Datensatzes entspricht dem arithmetischen Mittel der Distanzen aller  $n$  Zeitpunkte und wird fortan als  $\gamma_1$  bezeichnet. Die Schreibweise  $x_{i,H} \in \mathbb{R}^d$  steht für die x-Koordinaten der Handgelenkzentren aller  $d$  Datensätze zum Zeitpunkt  $i$ , ein  $E$  im Index steht für ein Datum des Ellbogenzentrums.

$$\gamma_1 = \frac{1}{n} \sum_{i=1}^n \sqrt{(x_{i,H} - x_{i,E})^2 + (y_{i,H} - y_{i,E})^2 + (z_{i,H} - z_{i,E})^2} \quad (4.17)$$

Generell besteht bei dieser Metrik ein Problem darin, dass ein auffälliger Metrikwert sowohl durch fehlerbehaftete Handgelenks- als auch Ellbogendaten verursacht sein kann und eine eindeutige Identifikation der Ursache nicht möglich ist.

### **Handgelenk-Handgelenk-Distanz**

Diese Metrik berechnet für Datensätze, die für beide Hände vorliegen, die Abstände der beiden Handgelenke zu allen Zeitpunkten  $n$  zwischen dem linken Datensatz und dem rechten Datensatz und bildet darüber das arithmetische Mittel. Der Metrikwert wird nachfolgend mit  $\gamma_2$  bezeichnet und nur für Datensätze mit beiden Körperhälften definiert.

$$\gamma_2 = \frac{1}{n} \sum_{i=1}^n \sqrt{(x_{i,H}^{links} - x_{i,H}^{rechts})^2 + (y_{i,H}^{links} - y_{i,H}^{rechts})^2 + (z_{i,H}^{links} - z_{i,H}^{rechts})^2} \quad (4.18)$$

Aufgrund der vorhergehenden Spiegelung der Trajektorien der linken Körperhälfte (siehe Abschnitt 4.6 auf Seite 33) sollten diese Distanzen möglichst gering sein. Ähnlich wie bei der Handgelenk-Ellbogen-Distanz liegt auch bei dieser Metrik das Problem vor, dass aus einem Ausreißer nicht geschlossen werden kann, welche der beiden Hand-Koordinaten Messfehler aufweist. Zudem liegen aufgrund von Messproblemen nicht für alle Datensätze Daten beider Körperhälften vor, sodass die Metrik nur auf eine Teilmenge aller Datensätze Anwendung findet. Ein weiteres Problem stellt die Abhängigkeit von der Präzision der Spiegelung dar; die Metrik ist im Gegensatz zur Ellbogen-Handgelenk-Distanz nicht unabhängig von anderen Berechnungen. Eine Ermittlung der Metrikergebnisse ohne vorherige Spiegelung ist nicht sinnvoll, da während einer seitlichen beidhändigen Greifbewegung die Handgelenkdistanzen naturgemäß variieren.

### **Vergleich der z-Koordinaten**

Die Handtrajektorien beider Körperseiten weisen bei natürlichen seitlichen Greifbewegungen zu gleichen Zeitpunkten ähnliche z-Koordinaten auf, da die Hände für gewöhnlich gleichmäßig auf einer Höhe bewegt werden. Deshalb bildet diese Metrik das arithmetische Mittel über den absoluten Differenzen der z-Koordinaten je Zeitpunkt  $i$  der Handbewegungen beider Körperseiten und wird mit  $\gamma_3$  bezeichnet. Sie ist wie die Handgelenk-

Handgelenk-Distanz nur für Datensätze mit beiden Körperhälften definiert.

$$\gamma_3 = \frac{1}{n} \sum_{i=1}^n |z_{i,H}^{links} - z_{i,H}^{rechts}| \quad (4.19)$$

Die Aussagekraft der letzten beiden Metriken basiert auch auf passgenauer Beschneidung der Datensätzen. Diese muss sicherstellen, dass die Koordinaten bei gleichem Zeitpunkt innerhalb des Datensatzes auch tatsächlich zum gleichen Zeitpunkt stattgefunden haben.

#### 4.8.2. Dynamische Schwellwertbestimmung

Dieser Abschnitt thematisiert die dynamische Schwellwertbestimmung beispielhaft an der Ellbogen-Handgelenk-Metrik. Für die anderen beiden Metriken wird das im Folgenden erläuterte Verfahren analog angewandt.

Die Idee der Wahl von Schwellwerten abhängig von den Daten ergründet sich in der Beobachtung, dass aufgrund verschiedener Testpersonen an verschiedenen Messtagen unterschiedliche Ellbogen-Handgelenk-Abstände vorliegen. Dies zeigt das Histogramm der Metrikwerte der Ellbogen-Handgelenk-Metrik aller Datensätze in Abbildung 4.11 auf Seite 46, die Spitzen zwischen 20 und 30cm drücken verschiedene Messpersonen/-tage aus. Die Ursache an den Unterschieden der Spitzen liegt zum einen an den anatomischen Unterschieden der Testpersonen, zum anderen an der je Messtag leicht unterschiedlichen Ausrichtung der Marker am Körper. Somit ist für jede Testperson und für jeden Messtag eine separate Betrachtung der Metrikwerte erforderlich. Da eine präzise manuelle Festlegung der oberen und unteren Schwellwert für jede dieser Verteilungen  $v$  einen hohen Aufwand darstellt und kaum einheitlich erfolgen kann, wird im Folgenden der Ansatz verfolgt, die Schwellwerte automatisch anhand der vorliegenden Daten zu ermitteln. Der untere Schwellwert der separat betrachteten Verteilung  $v$ , der durch Ansatz  $a$  für Metrik  $m$  berechnet wird, soll fortan als  $\kappa_{a,m,v}^{unten}$ , der obere Schwellwert des Ansatzes  $a$  als  $\kappa_{a,m,v}^{oben}$  bezeichnet werden. Generell gilt die Regel, dass ein Datensatz als Ausreißer in einer Metrik  $m$  betrachtet wird, sofern für dessen Metrikwert gilt:

$$\gamma_m < \kappa_{a,m,v}^{unten} \vee \gamma_m > \kappa_{a,m,v}^{oben} \quad (4.20)$$

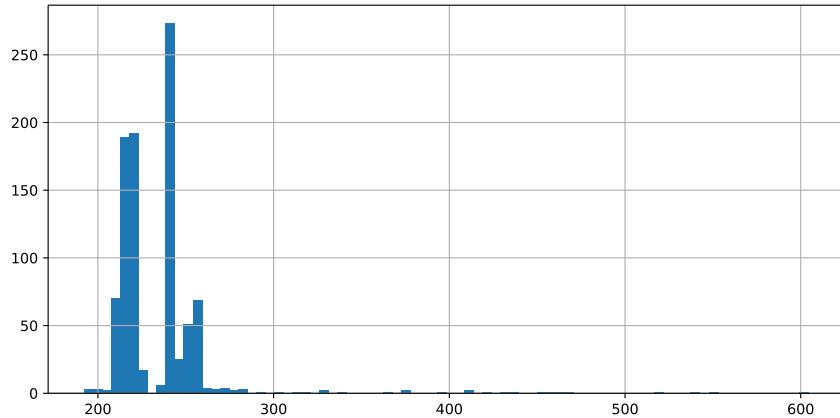


Abbildung 4.11.: Gemittelte Ellbogen-Handgelenkdistanzen aller Datensätze

Der folgende Absatz stellt drei verschiedene Ansätze zur dynamischen Bestimmung der Schwellwerte vor, deren Ergebnisse exemplarisch an einer Verteilung der Ellbogen-Handgelenks-Metrik in Abbildung 4.12 auf Seite 47 dargestellt sind. Die dargestellte Verteilung ist bereits von extremen Ausreißern bereinigt, was in Abschnitt 4.8.3 auf Seite 48 erläutert wird.

Ein Ziel der dynamischen Schwellwertbestimmung besteht darin, die Schwellwerte abhängig der Breite der Verteilung der Metrikwerte  $\gamma_m$  zu wählen. Denn breiter verteilte Metrikwerte stellen ein Indiz für stärker fehlerbehafteten Messdurchlauf dar, da die Metrikwerte bei idealen Messdaten konstant sein sollten. Bei stärker verrauschten Daten sollen mehr Daten aussortiert werden, als bei Daten mit allgemein sehr guter Qualität. Somit ist der erste Ansatz zur Wahl der Schwellwerte die Bestimmung des arithmetischen Mittels  $\bar{\gamma}_m^v$  der Verteilung  $v$  und anschließenden Subtraktion bzw. Addition der Standardabweichung  $s_m^v$  der Verteilung auf den Mittelwert für den unteren bzw. oberen Schwellwert (siehe Formeln 4.21 und 4.22).

$$\kappa_{1,m,v}^{unten} = \bar{\gamma}_m^v - s_m^v \quad (4.21)$$

$$\kappa_{1,m,v}^{oben} = \bar{\gamma}_m^v + s_m^v \quad (4.22)$$

Diese Art der Schwellwertbildung setzt allerdings eine symmetrische Verteilung der Metrikwerte voraus, die in der Praxis nicht gegeben ist. Besonders durch Messfehler entstandene Ausreißer sind nicht symmetrisch verteilt. Deshalb ist diese Methodik nicht geeignet. Die Ergebnisse dieses Ansatzes sind in Abbildung 4.12 grün visualisiert.

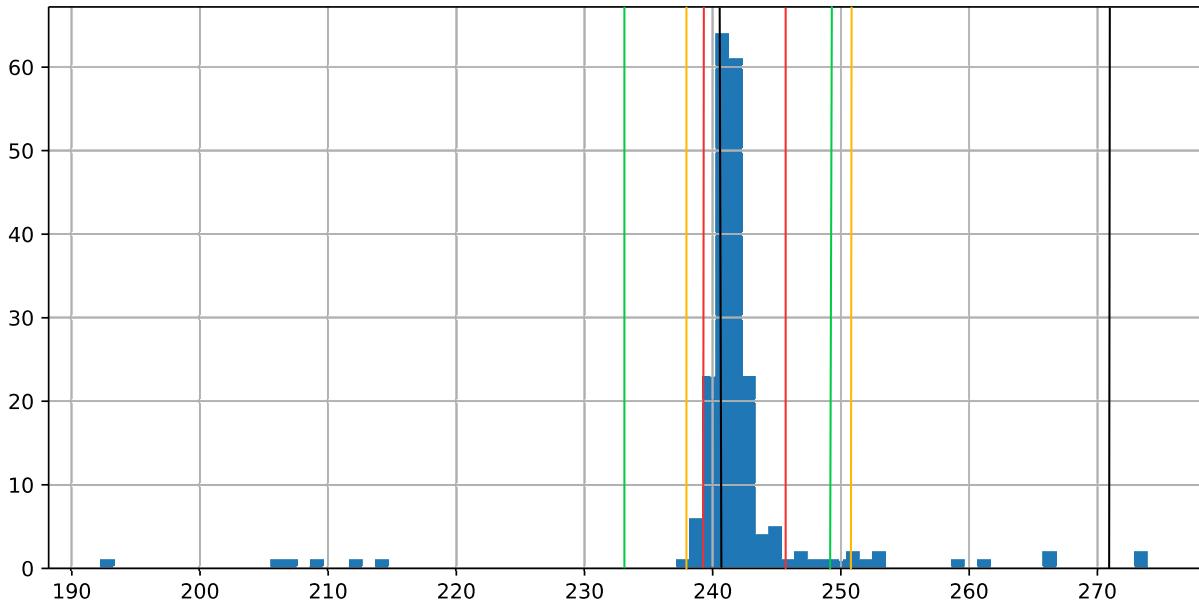


Abbildung 4.12.: Histogramm der Metrikwerte der Ellbogen-Handgelenks-Metrik einer Verteilung mit dynamisch bestimmten Schwellwerten: Ansatz 1 (grün), Ansatz 2 (gelb), Ansatz 3 (rot), Ansatz 3 ohne Extremwertfilterung (schwarz)

Ein weiterer Ansatz zur Wahl der Schwellwert besteht in der Bestimmung von Quantilen, dessen Ergebnisse in Abbildung 4.12 gelb dargestellt sind. Beispielsweise kann das 5%-Quantil als unterer Schwellwert und das 95%-Quantil der Verteilung als oberer Schwellwert verwendet werden. So wird keine symmetrische Verteilung angenommen, sondern werden in jeder Verteilung jeweils die 5% niedrigsten Metrikwerte und die 5% höchsten Metrikwerte der jeweiligen Verteilung  $v$  aussortiert (siehe Formeln 4.23 und 4.24).

$$\kappa_{2,m,v}^{\text{unten}} = Q(0.05) \quad (4.23)$$

$$\kappa_{2,m,v}^{\text{oben}} = Q(0.95) \quad (4.24)$$

Dieser Ansatz weist jedoch den Nachteil auf, dass immer 10% der Daten als Ausreißer klassifiziert werden, unabhängig davon wie breit die Metrikwerte verteilt sind - also wie fehlerbehaftet die Messdaten dieser Verteilung sind.

Deshalb soll als Methodik eine Kombination der beiden genannten Ansätze verwendet werden, indem der Unterschreitungsanteil der Quantile abhängig von der Standardabweichung  $s_m^v$  der jeweiligen Verteilung bestimmt wird. Hierzu muss ein statischer

Verrechnungsfaktor  $\delta_m$  festgelegt werden, durch den der Wert der Standardabweichung  $s_m^v$  in den Unterschreitungsanteil umgerechnet wird. Dieser Verrechnungsfaktor muss für jede Metrik  $m$  separat empirisch bestimmt werden, am Beispiel der Ellbogen-Handgelenk-Metrik ergibt sich  $\delta_1 = 0.009$  als sinnvoller Verrechnungsfaktor. Das bedeutet in diesem Fall, dass der Unterschreitungsanteil des Quantils für den unteren Schwellwert 0.9% der Standardabweichung der jeweiligen Verteilung entspricht (siehe Formel 4.25). Die Verrechnungsfaktoren der anderen beiden Metriken sind  $\delta_2 = 0.005$  und  $\delta_3 = 0.008$ . Für den Unterschreitungsanteil des Quantils zur Bestimmung des oberen Schwellwerts gilt  $1 - \delta_m s_m^v$ , wie Formel 4.26 zeigt.

Bei der dynamischen Bestimmung des Unterschreitungsanteils von Quantilen muss auch der Wertebereich des Unterschreitungsanteils beachtet werden, der Werte von 0 bis 1 einschließt. Um diesen Wertebereich einzuhalten, muss das Produkt  $\delta_m s_m^v$  zwischen 0 und 0.5 liegen, da der Unterschreitungsanteil für beide Quantile verwendet wird. Die Standardabweichung kann aufgrund der Wurzel von reellen Zahlen nicht negativ sein, folglich sind Werte kleiner 0 ausgeschlossen. Werte größer 0.5 können bei extrem breiten Verteilungen mit einer Standardabweichung von über  $s_m^v = 62.5$  auftreten, was in der Praxis unüblich ist. Dennoch wird programmatisch bei Werten oberhalb von 0.5 der Wert 0.5 gesetzt. So lässt sich letztendlich ein abhängig von der Standardabweichung bestimmter Prozentsatz an Daten vom oberen und unteren Ende der Verteilung als Ausreißer markieren. Die Ergebnisse dieses Ansatzes sind in Abbildung 4.12 rot dargestellt.

$$\kappa_{3,m,v}^{unten} = Q(\delta_m s_m^v) \quad \text{mit } \delta_m s_m^v \in [0, 0.5] \quad (4.25)$$

$$\kappa_{3,m,v}^{oben} = Q(1 - \delta_m s_m^v) \quad \text{mit } \delta_m s_m^v \in [0, 0.5] \quad (4.26)$$

### 4.8.3. Aussortierungsverfahren

Nach Durchlauf aller Metriken muss die zu aussortierende Menge an Datensätze bestimmt werden. Ein grundsätzliches Problem besteht dabei darin, dass eine Auffälligkeit in einer Metrik nicht zwangsläufig die niedrige Qualität eines Datensatzes impliziert. Das liegt an der Tatsache, dass beispielsweise in der Ellbogen-Handgelenk-Metrik ein auffälliger Abstand von Ellbogen und Handgelenk einen fehlerhaften Ellbogen- oder Handgelenkdatensatz zugrunde haben kann. Welcher der beiden Datensätze fehlerbehaftet ist, lässt sich durch die Metrik allein nicht feststellen. Deshalb erfolgt die Aussortierung erst nach Durchlauf aller Metriken, um die Ergebnisse aller dabei zu berücksichtigen. Im Folgenden

sollen verschiedene Ansätze zur Bestimmung der fehlerbehafteten Datenmenge diskutiert werden.

### Bildung der Aussortierungsmenge

Ein intuitiver Ansatz besteht in der Bildung der Schnittmenge über allen in einer beliebigen Metrik auffälligen Dateien. Die Mächtigkeit dieser Menge ist allerdings relativ gering, da die wenigsten Datensätze, die nicht schon in der Phase der Präfilterung (siehe Abschnitt 4.5) aussortiert wurden, so stark fehlerbehaftet sind, sodass sie in allen Metriken als Ausreißer markiert werden.

Ein weiterer Ansatz liegt in der Gewichtung der Auffälligkeiten bei verschiedenen Metriken mit statischen Gewichten. Jeder Datensatz besitzt hierbei einen Fehlerwert beginnend bei Null. Wird der Datensatz von einer Metrik als Ausreißer markiert, so addiert sich das Gewicht dieser Metrik auf den Fehlwert des Datensatzes. Überschreitet der Fehlerwert nach Prüfung aller Metriken eine bestimmte Fehlerschranke, so gilt der Datensatz als fehlerhaft und wird aussortiert. Dieses Verfahren funktioniert in der Praxis gut und unkompliziert. Es weist allerdings den Nachteil auf, dass das Gewicht bei Auffälligkeit in einer Metrik einen konstanten Wert hat und nicht die Stärke der Auffälligkeit widerspiegelt.

Deshalb soll in einem dritten Ansatz das Gewicht der jeweiligen Metriken mit einem dynamischen Anteil versehen werden, der sich abhängig der Stärke des Ausreißers der jeweiligen Metrik bestimmt. Dies ermöglicht eine präzise Gewichtung der Metriken, abhängig von der tatsächlichen Fehlerintensität.

### Ermittlung der Ausreißerstärke

Zur vergleichbaren Ermittlung der Ausreißerstärke je Metrik muss bei Berechnung der Metrikergebnisse die Fehlerstärke quantifiziert und normiert werden. Ziel der Normierung ist die Ausreißerstärken je Metrik auf das einheitliche Intervall von 0 bis 1 abzubilden, um so die Vergleichbarkeit der Ausreißerintensitäten zu gewährleisten und diese direkt in die Aussortierungsrechnung zu integrieren. Dieser Absatz befasst sich mit der Normierung der Werte, in dem nachfolgendem Absatz wird die Wahl der Fehlerschranke und Gewichte erläutert.

Die Normierung erfolgt nur für in einer Metrik als Ausreißer markierte Datensätze besteht

aus zwei Schritten. Zunächst wird als Quantifizierung der Ausreißerstärke für jeden auffälligen Metrikwert  $\gamma_m$  der Metrik  $m$  der Abstand zum Mittelwert  $\bar{\gamma}_m^v$  der jeweiligen spezifischen Verteilung  $v$  herangezogen (siehe Gleichung (4.27)). Dieser Abstand wird im Folgenden als  $\omega_m \in \mathbb{R}^d$  bezeichnet. Als Mittelwert gilt hierbei der Mittelwert der jeweiligen Verteilungen  $v$  pro Testperson und Testtag, wodurch die Ausreißerstärke individuell je Verteilung festgelegt werden kann.

$$\omega_m = |\bar{\gamma}_m^v - \gamma_m| \quad (4.27)$$

Als Alternative kann der gesamte Mittelwert einer Metrik verwendet werden, die Abstände hierzu gelten allerdings als weniger aussagekräftig, da die Verteilungen unterschiedliche Mittelwerte aufweisen und ein Ausreißer in einer Verteilung nicht zwangsläufig auch in einer anderen Verteilung als solcher bewertet wird.

Im Anschluss daran werden die Abstände aller Metrikwerte auf das Intervall von 0 bis 1 abgebildet, wie Gleichung (4.28) zeigt. So ergibt sich schlussendlich bei  $d$  Datensätzen und für jede Metrik  $m$  ein Vektor  $\rho_m \in \mathbb{R}^d$ , der die Intensität des Ausreißers in der jeweiligen Metrik im Intervall von 0 bis 1 quantifiziert.

$$\rho_m = \frac{\omega_m - \min \gamma_m}{\max \gamma_m - \min \gamma_m} \quad (4.28)$$

Hierbei fällt auf, dass daraufhin zahlreiche Datensätze die normierte Ausreißerstärke von  $\rho_m = 1$  aufweisen, wodurch die gesamte Verteilung im Intervall deutlich verzerrt wird. Der Grund hierfür liegt in extremen Ausreißern, die sehr hohe Abstände  $\omega_m$  aufweisen und dadurch die Auswirkung der eigentlich gesuchten Ausreißer in der Gewichtungsrechnung schmälern. Deshalb wird jeder Metrikberechnung und Normierung eine statische Filterung extremer Ausreißer anhand eines Maximalwerts vorangesetzt. Diese markiert Metrikwerte, die eindeutig auf Messfehler hinweisen, direkt als Ausreißer und versieht sie mit der maximalen Ausreißerstärke von  $\rho_m = 1$ . So berücksichtigt die anschließende Metrikberechnung und Normierung extreme Ausreißer nicht mehr und verteilt die Stärke der dortigen normierten Auffälligkeiten gleichmäßiger im Intervall von 0 bis 1.

Am Beispiel der Ellbogen-Handgelenk-Metrik sind die extremen Ausreißer im Histogramm in Abbildung 4.11 auf Seite 46 klar ersichtlich, der realistische Wertebereich für Ellbogen-Handgelenkdistanzen liegt Beobachtungen zufolge zwischen 20 und 30cm und variiert je nach Testperson und exakten Position der Marker. Alle Werte oberhalb von 30cm lassen

sich direkt als extreme Ausreißer markieren. Zudem berücksichtigt die Methodik auch die Standardverteilung und sortiert Datensätze mit sehr breiten Verteilungen bezüglich des Metrikwertes ebenfalls aus. Neben der verbesserten Normierung steigert dieses Verfahren auch die Präzision der dynamischen Schwellwertbestimmung, da sich diese folglich auf die noch aufwendigere, feine Trennung von qualitativ hochwertigen Datensätzen und auffälligen Datensätzen fokussieren kann. Dieser Effekt zeigt auch Abbildung 4.12 durch die schwarz markierten Schwellwerte, welche durch Anwendung der dynamischen Schwellwertbestimmung auf den nicht von Extremwerten bereinigten Datensatz bestimmt werden. Hierbei ist klar die trotz angepasstem Wert des Verrechnungsfaktor die mangelnde Präzision und Qualität der Schwellwert ersichtlich. Zur Bestimmung der extremen Ausreißer ist keine dynamische Schwellwertbestimmung erforderlich, da diese eindeutig auffällig sind und direkt als solche markiert werden können.

### **Manuelle Aussortierung als Referenz**

Zur allgemeinen Einstellung der Hyperparameter der Metriken, wie dem Verrechnungsfaktor  $\delta_m$ , wird für ca. 1000 Datensätze eine manuelle Aussortierung vorgenommen. Diese dient fortan als Referenzdatensatz und lässt sich auch zur generellen Bewertung der Effektivität der Datenvorverarbeitung und Aussortierung heranziehen. Eine manuelle Filterung aller Datensätze ist nicht sinnvoll und widerspricht zudem grundlegend dem Ansatz der automatischen Datenvorverarbeitung.

Die manuelle Aussortierung erfolgt teilautomatisiert durch ein in Python entwickeltes Tool, dessen Ablauf in folgendem Abschnitt beschrieben wird. Ein Python-Skript stellt zu jedem der ca. 1000 Datensätze das Geschwindigkeitsprofil dar. Anhand dessen kann entschieden werden, ob der Datensatz fehlerbehaftet ist oder nicht. Fehlerhafte Datensätze zeichnen sich durch signifikante Ausreißer oder Sprünge im Geschwindigkeitsprofil oder einer ungewöhnlichen Form des Profils, z.B. eine stark von der erwartbaren Glockenform abweichenden Form, aus. Die gefällte Entscheidung lässt sich im Tool durch Auswahl des entsprechenden Buttons bestätigen. Daraufhin speichert das Skript die getroffene Entscheidung zum dargestellten Datensatz ab und visualisiert das Geschwindigkeitsprofil des nächsten Datensatzes. Sind alle Datensätze durchlaufen, liefert das Python-Skript als Ausgang zwei CSV-Dateien mit den Dateinamen aller fehlerbehafteten bzw. aller fehlerfreien Datensätze.

Das Ergebnis der manuellen Aussortierung soll auch in der Filterung der Datensätze berücksichtigt werden und dient als direktes Kriterium für die Aussortierung eines Datensatzes.

### **Wahl der Gewichte und Fehlerschranke**

Letztendlich kann durch die quantifizierten Ausreißerintensitäten eine gewichtete Fehlerstärke  $\epsilon \in \mathbb{R}^d$  für alle  $d$  Datensätze berechnet werden. Der hierzu verwendete Algorithmus ist in Algorithmus 1 auf Seite 53 dargestellt.

Die Wahl der Gewichte der einzelnen Metriken und die Höhe Fehlerschranke ist so gestaltet, dass die Auffälligkeit in einer einzigen Metrik nicht ausreicht, um die Fehlerschranke zu überwinden. So soll mit dem grundsätzlichen Problem, dass beispielsweise eine auffällige Ellbogen-Handgelenk-Distanz nicht eindeutig auf eine fehlerhafte Handgelenks- oder fehlerhafte Ellbogentrajektorie zurückgeführt werden kann, umgegangen werden.

Grundsätzlich wird der  $\epsilon$ -Wert des jeweiligen Datensatzes erhöht, wenn der Datensatz in einer Metrik als Ausreißer markiert ist. Der Wert des Inkrements berechnet sich hierbei aus dem Gewichtungsfaktor  $\pi_m$  der Metrik und der Ausreißerintensität  $\rho_m \in [0, 1]$  des Datensatzes in dieser Metrik und soll im Intervall  $[\pi_m, 2\pi_m]$  liegen. So ergibt sich pro Metrik eine Inkrementierung nach Gleichung (4.29).

$$\epsilon = \epsilon + \pi_m(1 + \rho_m) \quad (4.29)$$

$\pi_m$  gewichtet dabei die Relevanz der Metriken. Höchste Relevanz haben die Ergebnisse der Handgelenk-Handgelenk-Metrik, da Ausreißer in dieser Metrik klar auf Fehler in den für das maschinelle Lernen relevanten Handgelenkdaten zurückzuführen sind. Geringere Relevanz weisen die Ausreißer der Ellbogen-Handgelenk-Metrik und der z-Differenzen-Metrik auf, deren Aussagekraft ist vor allem unter der Beobachtung, dass viele Ellbogendatensätze fehlerbehaftet sind, für die Qualität der Handgelenkdaten geringer.

Ist ein Datensatz in der Menge der manuell aussortierten Daten enthalten, so wird dieser direkt aussortiert.

```

für alle relokierter Dateien tue
   $\epsilon = 0;$ 
  wenn Datei in Ellbogen-Handgelenk-Metrik als Ausreißer markiert dann
    |  $\epsilon_+ = 2 \cdot (1 + \text{Ausreißerstärke } \rho_1 \text{ der Datei});$ 
  Ende
  wenn Datei in Handgelenk-Handgelenk-Metrik als Ausreißer markiert dann
    |  $\epsilon_+ = 4 \cdot (1 + \text{Ausreißerstärke } \rho_2 \text{ der Datei});$ 
  Ende
  wenn Datei in z-Differenz-Metrik als Ausreißer markiert dann
    |  $\epsilon_+ = 2 \cdot (1 + \text{Ausreißerstärke } \rho_3 \text{ der Datei});$ 
  Ende
  wenn Datei manuell aussortiert dann
    |  $\epsilon_+ = 4.01;$ 
  Ende
  wenn  $\epsilon > 4$  dann
    | Sortiere Datei aus;
  Ende
Ende

```

**Algorithmus 1:** Gewichtete Aussortierung fehlerhafter Datensätze

## 4.9. Rotation

Die Phase der Rotation dient der weiteren Vereinheitlichung der Daten und hat das Ziel, die Handgelenktrajektorien so zu rotieren, dass die Anfangsverläufe der Bewegungen alle in dieselbe Richtung zeigen. Damit soll die Möglichkeit zur Generalisierung des Neuronalen Netzes verbessert werden.

### 4.9.1. Idee

Wie in Abschnitt 3.2 beschrieben dient eine feste Anzahl an Datenpunkten vom Startpunkt der Trajektorie ab als Eingang für das zu trainierende Neuronale Netz. Diese Menge an Datenpunkten soll als Anfangsbewegung bezeichnet werden, der letzte Datenpunkt der Anfangsbewegung wird in folgendem als Rotationspunkt  $D$  bezeichnet.

Da die Trajektorien in der Phase der Rotation bereits auf den Ursprung verschoben vorliegen (siehe Abschnitt 4.7), repräsentiert der Ortsvektor des Rotationspunktes die Richtung der Anfangsbewegung. Bei Visualisierung der Trajektorien fällt auf, dass die Anfangsbewegungen in teilweise sehr unterschiedliche Richtung verlaufen und nicht gleich-

mäßig im Raum verteilt sind. Dieses Problem soll durch die Rotation der Kurven behoben werden, sodass anschließend die Anfangsbewegungen aller Trajektorien einheitlich in x-Achsenrichtung zeigen. Um eine Retransformation zu ermöglichen und einem Informationsverlust vorzubeugen, muss die ursprüngliche Position des Rotationspunktes in den Metadaten der rotierten Trajektorien mitgespeichert werden.

#### 4.9.2. Umsetzung

Die Koordinatentransformation der Rotation lässt sich mathematisch durch die Linksmultiplikation einer Rotationsmatrix mit den Koordinaten der Trajektorie realisieren. Zur Berechnung der Matrix muss der Drehwinkel und die Drehachse der Rotation bekannt sein, welche im Folgenden hergeleitet werden.

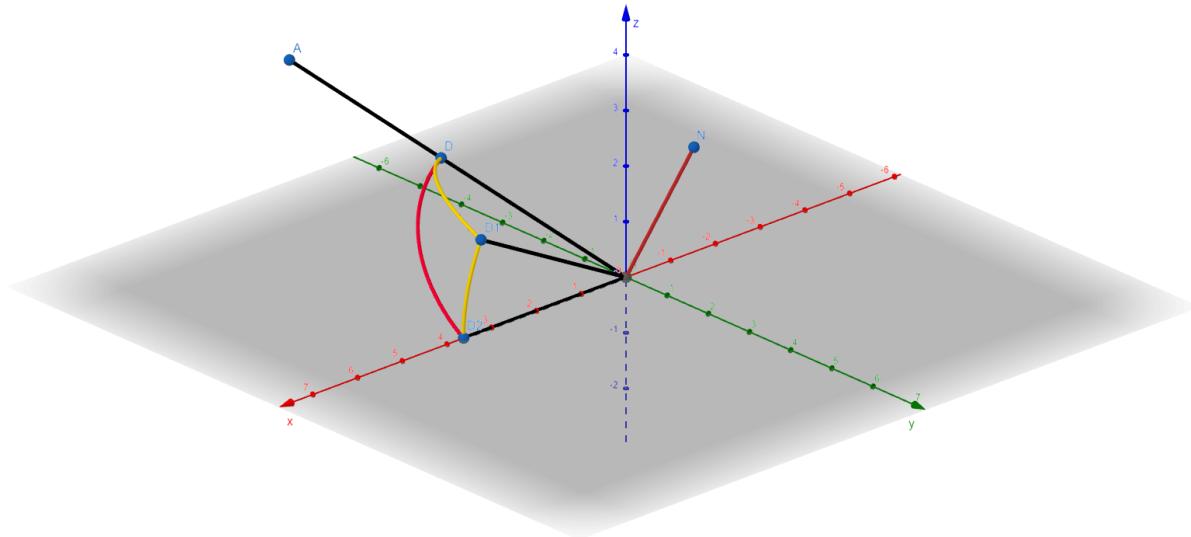


Abbildung 4.13.: Dreidimensionale Visualisierung von Rotationswinkeln und Drehachse

Abbildung 4.13 stellt die Situation im dreidimensionalen Raum zum besseren Verständnis dar. Punkt  $D$  ist hierbei der gewählte Rotationspunkt der Trajektorie, der auf den Punkt  $D_2$  auf der x-Achse rotiert werden soll. Dazu gibt es, wie die Abbildung zeigt, zwei verschiedene Ansätze. Zum einen sind zwei Rotationen um die z- und y-Achse denkbar, wofür die gelb dargestellten Rotationswinkel errechnet werden müssten. Dieser Ansatz

birgt den Vorteil, dass die Drehachsen sehr leicht zu ermitteln und die Rotationsmatrizen durch Achsenrotationen einfach gehalten sind. Allerdings sind zur Ermittlung der beiden Drehwinkel weitere geometrische Berechnungen notwendig, da zunächst der Ortsvektor  $\vec{d}_1$  vom Punkt  $D_1$  bestimmt werden muss.

Der zweite Ansatz sieht eine direkte Rotation auf die x-Achse vor. Der in der Abbildung rot dargestellte Drehwinkel  $\psi$  lässt sich als Raumwinkel zwischen x-Achse und dem bekannten Ortsvektor  $\vec{d}$  des Rotationspunktes der Trajektorie durch die Formel 4.30 ermitteln.

$$\cos(\psi) = \frac{x}{\sqrt{d_x^2 + d_y^2 + d_z^2}} \quad (4.30)$$

Die Rotationsachse ist geometrisch betrachtet die Normale  $\vec{n}$  der von x-Achse und Rotationspunkt-Ortsvektor  $\vec{d}$  aufgespannten Ebene und ergibt sich mathematisch durch das Kreuzprodukt der beiden Vektoren (siehe Formel 4.31). Die sich ergebende Normale liegt immer in der y-z-Ebene, da sie zur x-Achse einen rechten Winkel aufspannt und aufgrund der Relokation der Trajektorie durch den Ursprung verläuft. Zur Einsetzung in die Rotationsmatrix ist zuvor eine Normierung der Normalen erforderlich.

$$\vec{n} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \times \vec{d} \quad \vec{e}_n = \frac{\vec{n}}{|\vec{n}|} \quad (4.31)$$

Für die Rotation aller Trajektorien muss für jeden Datensatz die Rotationsmatrix aufgestellt werden und mit allen Punktkoordinaten des Datensatzes linksmultipliziert werden. Da die x-Komponente der Drehachse immer 0 ist, kann die allgemeine Rotationsmatrix wie in 4.33 gezeigt vereinfacht werden.

$$\text{mit } \vec{e}_n = \begin{pmatrix} e_{n1} \\ e_{n2} \\ e_{n3} \end{pmatrix} = \begin{pmatrix} 0 \\ e_{n2} \\ e_{n3} \end{pmatrix} \quad (4.32)$$

$$R = \begin{pmatrix} \cos(\psi) & -e_{n3} \sin(\psi) & e_{n2} \sin(\psi) \\ e_{n3} \sin(\psi) & e_{n2}^2 (1 - \cos(\psi)) + \cos(\psi) & e_{n2} e_{n3} (1 - \cos(\psi)) \\ -e_{n2} \sin(\psi) & e_{n3} e_{n2} (1 - \cos(\psi)) & e_{n3}^2 (1 - \cos(\psi)) + \cos(\psi) \end{pmatrix} \quad (4.33)$$

Nach der Rotationsphase ist die Datenpipeline beendet. Die Daten liegen in einem einheitlichen Format vor und können zum Training des Neuronalen Netzes verwendet werden.

# 5. Vorhersagen über die Trajektorie

Dieses Kapitel erläutert die Vorhersage von Trajektorien und thematisiert dabei vor allem das Maschinelle Lernen zur Prädiktion von Endpunkt und Bewegungsdauer in den Abschnitten 5.1 und 5.2 an. Abschnitt 5.3 erläutert die daran anschließende Berechnung der Bewegungspfade anhand des Minimum Jerk Models.

## 5.1. Ansätze für Machine Learning

Wie in Abschnitt 3.2 beschrieben soll im Rahmen dieser Arbeit durch Maschinelles Lernen die Endposition sowie Dauer einer Bewegung anhand der Datensätze der Anfangsbewegung vorhergesagt werden. Zur Erreichung der Lernziele kann das Szenario sowohl als Regressions- als auch als Klassifikationsproblem aufgefasst werden. Im Folgenden sollen diese beiden Ansätze zur Erreichung der Lernziele vorgestellt und diskutiert werden.

### 5.1.1. Regressionsproblem

Die Vorhersage der Endposition und Dauer lässt sich zunächst als Regressionsproblem interpretieren. Für die Vorhersage der Endposition dienen die kartesischen x-,y- und z-Koordinaten des letzten Datenpunktes der Trajektorie als Lernziel im überwachten Lernprozess. Im Fall der Dauer, stellt die Anzahl der äquidistanten Zeitpunkte des Datensatzes das Lernziel dar.

Die Entscheidung, mit Maschinellem Lernen lediglich die Endposition und Dauer statt direkt die gesamten Trajektorie vorherzusagen, ergründet sich in der Anzahl der zu vorhersagenden Freiheitsgrade. Denn die Vorhersage der Endposition und Dauer weist lediglich vier Freiheitsgrade auf, wohingegen bei der Prädiktion der gesamten Trajektorie allein pro Zeitpunkt drei Freiheitsgrade und einen für die Dauer der Bewegung vorliegen. Somit lässt sich der erstere Ansatz bei geringer Trainingsdatenmengen als deutlich besser lösbar bewerten.

### 5.1.2. Klassifikationsproblem

Aus der Praxis heraus betrachtet ist eine exakte Vorhersage der Endpositionen oder Trajektorienverläufe allerdings kaum notwendig. Zur Kollisionsvermeidung in der kollaborativen Robotik ist wie Ding u. a. [20] beschreiben die Information des groben Aufenthaltsbereichs der Hand ausreichend, da selbst zu einer exakten vorhergesagten Position ein Sicherheitsabstand vom sich bewegenden Roboter eingehalten werden müsste. Diese Betrachtungsweise führt zur Auffassung als Klassifikationsproblem, in dem der vom Roboter erreichbare Raum in würfelförmige Teilräume aufgeteilt wird. So kann das Neuronale Netz statt der exakten Position der Hand - ob für alle Zeitpunkte bei der Trajektorienprädiktion oder nur für den letzten bei der Prädiktion der Endpunkte - die Aufenthaltswahrscheinlichkeiten der Hand für jeden Teilwürfel des Raumes vorhersagen. Die Granularität der Raumwürfel ergibt sich dabei aus der Menge und Qualität der Trainingsdaten, sodass eine hohe Präzision gewährleistet ist.

Raumwürfel mit einer einen Schwellwert übersteigenden Aufenthaltswahrscheinlichkeit zum jeweiligen Zeitpunkt müssen vom kollaborativen Roboter gemieden werden, wodurch eine Kollisionsfreiheit gewährleistet ist. Der Schwellwert ist auf Basis der Qualität des trainierten Netzes festzulegen.

Auch die Vorhersage der Dauer lässt sich als Klassifikation auffassen, da die diskrete Anzahl an Zeitpunkten vorhergesagt werden soll und diese sich auf einen Wertebereich von 85 bis 150 beschränken lässt.

## 5.2. Umsetzung

Dieses Kapitel stellt die konkrete Umsetzung des Regressionsansatzes vor, da dieser in der Literatur bisher kaum verfolgt wird. Deshalb soll im Rahmen dieser Arbeit der Ansatz über Klassifikation nicht weiter thematisiert werden, hierfür sei verwiesen auf Ding u. a. [20]. Aufgrund unterschiedlicher Architekturen, Hyperparameter und Kostenfunktionen soll für die Vorhersage von **Endposition** und **Dauer** jeweils ein separates Neuronales Netz trainiert werden.

### 5.2.1. Eingangsdaten

Als Eingangsdaten können alle gefilterten Datensätze mit einer Mindestlänge 120% der Länge der Anfangsbewegung verwendet werden. Bei kürzeren Datensätzen liegen die zu vorhersagenden Koordinaten noch innerhalb oder knapp hinter der Anfangsbewegung und hätten daher nur geringe Aussagekraft.

Anschließend kann die Eingangsmatrix konstruiert werden, in der jede Zeile einem Trainingsdatensatz entspricht. Folglich sind die Koordinatenangaben aller Dimensionen jedes Datensatzes in einer Zeile sequenziell zu finden. Bei einer Anfangsbewegung der Länge 85 und  $n$  Datensätzen kann so eine Eingangsmatrix  $E \in \mathbb{R}^{n \times 3 \cdot 85}$  konstruiert werden, es gibt folglich 255 Eingangsmerkmale. Diese Eingangsmatrix gilt für beide neuronale Netze. Für die **Endpositionen** ergibt sich eine erwartete Ausgangsmatrix  $A_{Endposition} \in \mathbb{R}^{n \times 3}$ , die je Trainingsdatensatz  $n$  eine Zeile mit den drei Koordinaten der zu vorhersagenden Endposition aufweist. Das neuronale Netz zur Vorhersage der **Dauern** erhält als Ausgangsmatrix  $A_{Dauer} \in \mathbb{R}^n$  mit der Anzahl an Zeitpunkten je Datensatz  $n$ .

### 5.2.2. Vorbereitung

In Phase der Datenvorbereitung liegt das Ziel darin, die zuvor konstruierten Eingangs- und Ausgangsmatrizen für das Training aufzubereiten. Für die neuronalen Netze zur Vorhersage von Endposition und Dauer wird jeweils dasselbe Verfahren zu Vorverarbeitung der Daten genutzt.

Dazu ist zunächst die zeilenweise Mischung der Matrizen notwendig, damit durch die Aufnahmereihenfolge ungewollt bedingte Zusammenhänge durch eine neue zufällige Reihenfolge eliminiert werden und das neuronale Netz unabhängig von der Reihenfolge der Daten trainieren kann.

Nach der Mischung der Daten erfolgt die Aufteilung in Trainings- und Testdaten, die fortan getrennt behandelt werden. Nach dem Trainings soll anhand der Trainingsdaten die Generalisierungseigenschaft des Modells bewertet werden. Dazu ist es wichtig, dass das neuronale Netz die dafür verwendeten Daten nicht im Trainingsprozess verwendet, da sonst die Aussagekraft über die Fähigkeit zur Generalisierung sinkt. Zur Mischung und Aufteilung der Daten nach vorgegebenem Prozentsatz wird die `train_test_split`-Methode aus dem `model-selection`-Package von `sklearn` verwendet.

Um ein bestmögliches Trainingsergebnis zu erzielen ist es wichtig, dass die Daten in einem einheitlichen Wertebereich vorliegen. Bevor das Training beginnt, werden die Trainings- und Testdaten deshalb mittels `MinMaxScaler()` aus dem Preprocessing-Package von *sklearn* in den Wertebereich von 0 bis 1 transformiert. Die Transformation von Trainings- und Testdaten erfolgt dabei separat, da sonst eine gegenseitige Beeinflussung möglich wäre und die beiden Datenmengen nicht als unabhängig betrachtet werden können.

### 5.2.3. Training

Vor dem eigentlichen Training ist die Modellarchitektur festzulegen. Für die Vorhersage der **Endposition** hat sich empirisch ein einfaches Neuronales Netz mit Dense-Layern als sinnvoll erwiesen, siehe dazu Abbildung 5.1. In den verborgenen Schichten verwendet das Netz als Aktivierungsfunktion Rectified Linear Units (ReLUs) und in der Ausgabeschicht eine lineare Aktivierungsfunktion. Die Fragezeichen in der Abbildung stehen für eine beliebige Anzahl an Datensätzen.

Das Neuronale Netz zur Vorhersage der **Dauer** ist ähnlich aufgebaut, besitzt allerdings eine Schicht weniger.

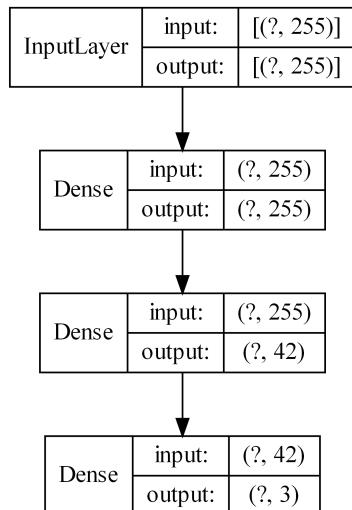


Abbildung 5.1.: Architektur des neuronalen Netzes zur Vorhersage der Endpositionen

Bei der Festlegung der Kostenfunktion ist beispielsweise Mean Squared Error (MSE) oder Mean Absolute Error (MAE) möglich, also der mittlere quadratische oder absolute Abstand des vorhergesagten zum tatsächlichen Wert. Diese Kostenfunktionen sind beide

für die Vorhersage von dreidimensionalen Endpunkten nicht passend, da sie die drei Koordinaten als unabhängige Größen betrachten und den Abstand je Koordinate minimieren würden (siehe Formel 5.1).  $Y \in \mathbb{R}^{n \times 3}$  ist hierbei die Matrix der tatsächlichen x-, y- und z-Koordinate für jeden der  $n$  Datensätze des Batches,  $\hat{Y} \in \mathbb{R}^{n \times 3}$  stellt die Matrix mit den drei vorhergesagten Koordinaten für jeden Datensatz dar. Die beiden Matrizen werden komponentenweise voneinander abgezogen, anschließend wird aus den drei quadrierten Differenzen der Mittelwert gebildet, sodass sich für jeden Datensatz  $i$  genau ein Fehlwert ergibt.

$$C_{mse} = \frac{1}{3} \sum_{j=1}^3 (y_{ij} - \hat{y}_{ij})^2 \quad \text{mit } c_{mse} \in \mathbb{R}^n \quad \text{und } i \in [1, \dots, n] \quad (5.1)$$

Sinnvoller ist hierbei als Kostenfunktion für das Neuronale Netz zur Vorhersage der Endpunkte die euklidische Distanz zwischen dem vorhergesagten und dem tatsächlichen Endpunkt heranzuziehen, da diese den inhaltlichen Zusammenhang der drei Größen besser widerspiegelt. Diese soll hierbei noch quadriert werden, um hohe Fehlerdistanzen stärker zu gewichten. Eine Mittelwertbildung wie bei MSE ist nicht notwendig, da die quadrierte euklidische Distanz bereits pro Datensatz  $i$  einen Fehlerwert, die Fehlerdistanz, liefert.

$$C_{euclid} = \left( \sqrt{\sum_{j=1}^3 (y_{ij} - \hat{y}_{ij})^2} \right)^2 \quad \text{mit } c_{euclid} \in \mathbb{R}^n \quad \text{und } i \in [1, \dots, n] \quad (5.2)$$

$$= \sum_{j=1}^3 (y_{ij} - \hat{y}_{ij})^2 \quad (5.3)$$

Der Unterschied beider Kostenfunktionen besteht zusammenfassend lediglich im Faktor  $\frac{1}{3}$ . Dennoch lassen sich mit der angepassten euklidischen Kostenfunktion bessere Trainingsergebnisse erzielen, weshalb diese zur Vorhersage der Endpositionen verwendet wird. Im Falle der Vorhersage der Dauern erweist sich MAE als geeignete Kostenfunktion, da keine räumlich interpretierbare Daten vorliegen.

Zur kontinuierlichen Überprüfung des Trainingsfortschritts wird zudem ein Menge an Datensätzen zur Validierung abgetrennt.

### 5.2.4. Nachbereitung

Nach dem Training muss die Qualität des trainierten Modells geprüft werden. Dazu sollen nachfolgend drei Verfahren am Beispiel der Vorhersage der Endpositionen aufgezeigt werden, diese lassen sich angepasst auch auf die Vorhersage der Dauern übertragen. Die Verfahren stellen in Summe den Trainingserfolg sicher. Für die Verfahren werden anhand des trainierten Modells Vorhersage zu bekannten Daten getroffen und diese anschließend mit den tatsächlichen Werten verglichen.

Zunächst lässt sich die Qualität des Modells anhand von **Metrikwerten** festmachen. Dazu kann die euklidische Distanz zwischen vorhergesagten und tatsächlichen Endpositionen ermittelt werden, nach Rücktransformation auf den ursprünglichen Wertebereich kann der Mittelwert der Distanzen als durchschnittliche Distanz in Millimeter interpretiert werden. Als weiteres Bewertungskriterium der Trainingsqualität kann das Verhältnis der Trainings- und Testfehler anhand dem Mittelwert der euklidischen Distanzen betrachtet werden. Dieses sollte idealerweise bei Eins liegen. Wenn bei den Testdaten deutlich höhere Fehler auftreten als bei den Trainingsdaten, so liegt eine Überanpassung des Modells an die Trainingsdaten und damit einhergehende schlechte Generalisierungsfähigkeit vor. In diesem Fall ist die Epochenzahl oder Modellkomplexität zu reduzieren oder Menge der Trainingsdaten zu erhöhen.

Eine weitere Möglichkeit den Trainingserfolg auch nach den einzelnen Koordinaten differenziert zu betrachten, stellt eine **Visualisierung der Trainings- und Testergebnisse** dar. In einem zweidimensionalen kartesischen Koordinatensystem können dazu auf der x-Achse die vorhergesagten Wert einer Koordinate, beispielsweise der x-Koordinate, und auf der y-Achse die tatsächlichen Werte dieser Koordinate vom Endpunkt aufgetragen werden. Die sich ergebenden Punkte sollten für eine hohe Trainingsqualität möglichst auf der Ursprungsgerade oder in der dazu parallelen Umgebung liegen. Der Vorteil dieser Visualisierung besteht darin, dass sie für die x-, y-, und z-Koordinate separat durchgeführt werden kann. So lassen sich Auffälligkeiten in einer Dimension sowie der erlernte Zusammenhang leicht erkennen und bewerten. Die drei Dimensionen können über die euklidische Norm des Weiteren in einer Visualisierung aggregiert dargestellt werden. Die Visualisierung bietet sich sowohl für Test- als auch Trainingsdaten an, bei den Trainingsdaten muss zunächst anhand des Modells die Vorhersage der Endpunkte erfolgen. Die

beschriebenen Schaubilder werden im Ergebniskapitel in den Abbildungen A.2 und A.3 ab Seite XIV vorgestellt.

Als dritte Möglichkeit können zufällig drei vorhergesagte Endpositionen mit den zugehörigen tatsächlichen aus den Testdaten selektiert werden und manuell verglichen werden. Dies ermöglicht die **stichprobenartige Einschätzung** des Trainingserfolgs.

### 5.2.5. Einstellung der Hyperparameter

Für ein erfolgreiches Training ist die Einstellung zahlreicher Hyperparameter erforderlich, die in diesem Abschnitt erläutert werden soll.

Unter Hyperparameter fallen zum einen Angaben zur Aufteilung der Daten in Trainings-, Test- und Validierungsdaten. Zum anderen muss auch die Anzahl der Trainingsepochen, die Batch-Size und die Länge der Anfangsbewegung festgelegt werden, anhand derer die Endposition vorhergesagt werden soll. Weitere Hyperparameter stellen auch der gewählte Rotationspunkt zur Rotation der Daten sowie die Länge der Anfangsbewegung dar, auf die das Modell trainiert wird. Der Rotationspunkt muss nicht zwangsläufig der letzte Punkt der Anfangsbewegung sein, sondern kann auch schon davor liegen. Ein Rotationspunkt hinter dem letzten Punkt der Anfangsbewegung macht keinen Sinn, da dieser in der Praxis nicht bekannt und somit eine Rotation in diese Richtung nicht möglich wäre.

Zur bestmöglichen Bestimmung dieser Parameter bietet sich eine Automatisierung des Trainingsprozesses an, sodass automatisiert Permutationen aus jeweils vorgegebenen Werten je Parameter getestet werden können. Nach jedem Trainingsdurchlauf muss dazu lediglich die getestete Parameterkonfiguration mit den erzielten Ergebnissen gemessen an den Metrikwerten bzw. der Kostenfunktion in eine Datei ausgeschrieben werden. So können nach erfolgten Trainingsdurchläufen empirisch die bestmögliche Parameterbelegungen ermittelt werden. Ausschlaggebend für die Auswahl qualitativ hochwertiger Trainingsergebnisse ist dabei zum einen eine möglichst kurze Anfangsbewegung als Eingangsdaten sowie ein möglichst geringer Wert in der festgelegten Kostenfunktion. Interessante Konfiguration können manuell anhand der in 5.2.4 beschriebenen Visualisierungen und des Stichprobentests validiert werden.

Die so ermittelte bestmögliche Trainingskonfiguration wird in Kapitel 6 vorgestellt.

## 5.3. Pfadberechnung

Mittels denen in diesem Kapitel prognostizierbaren Endpunkten und Bewegungsdauern soll nun mit dem Minimum Jerk Modell der Verlauf der Trajektorie anhand der aufgenommenen Anfangsbewegung vorausberechnet werden.

Die zur Verfügung stehenden Methoden umfassen die in Abschnitt 3.1.2 genannten verschiedenen Verfahren des Minimum Jerk Modells. In den Datensätzen dieser Arbeit wird das Zielobjekt von links und rechts seitlich gegriffen. Dadurch beschreibt die Bahn der Bewegung keine geradlinige Strecke mehr, sondern eine leichte Kurve. Das Minimum Jerk Model für unbeschränkte Punkt-zu-Punkt Bewegungen von Flash und Hogan [12] eignet sich für die Datensätze dieser Arbeit daher nicht. Für eine verbesserte Approximation der Bahnkurve bietet sich das Modell für gekrümmte Bewegungen von denselben Autoren an sowie das Modified Minimum Jerk Model von Zhao u. a. [19]. Ein direkter Vergleich dieser beiden Modelle kann aus Kapazitätsgründen nicht mehr im Rahmen dieser Arbeit durchgeführt werden, daher fällt die Entscheidung auf das erstgenannte Modell für gekrümmte Punkt-zu-Punkt-Bewegungen.

Notwendig hierfür ist neben der Information des Start- und des Endpunkts  $P_1$  und  $P_2$  der Bewegung sowie der Dauer des Greifvorgangs auch die Angabe eines dritten Punkts  $P_C$ , durch den die Bahnkurve verlaufen soll. In der idealen Modellierung der realen Trajektorie mit diesem Verfahren ist dies der Punkt der größten Entfernung von der Strecke zwischen Start- und Endpunkt. Da dieser bei der Betrachtung der Anfangsbewegung noch nicht aufgetreten sein muss, soll die Annahme getroffen werden, dass derjenige Punkt der Anfangsbewegung mit dem größten Abstand von der Strecke zwischen Startpunkt und dem durch das ML-Modell prognostizierten Endpunkt diesen dritten Punkt  $P_C$  darstellt.

Das Vorgehen zur Bestimmung der gekrümmten Trajektorie sieht vor, eine Berechnung der Bahnkurve zeitlich vor und nach dem Punkt  $P_C$  durchzuführen. Die mathematischen Grundlagen hierfür sind in Gleichung (3.5) genannt. Die Berechnung der Bahnkurve zeitlich vor dem Punkt  $P_C$  kann vor dem Hintergrund, dass eine Modellierung der bereits aufgezeichneten Bahnkurve keine Rückschlüsse auf den zukünftigen Verlauf der Handbewegung gibt, ignoriert werden. Um einen flüssigen Übergang von der Kurve  $P_1P_C$  vor dem Punkt  $P_C$  zur Kurve  $P_CP_2$  nach diesem Punkt zu erreichen, sollen in dieser Arbeit beide Kurvenabschnitte mit den angegebenen Formeln dennoch modelliert werden.

# 6. Ergebnisse

Dieses Kapitel soll die Ergebnisse der in den vorherigen Kapiteln methodisch beschriebenen Datenvorverarbeitung und vor allem des Machine Learnings vorstellen sowie analysieren. Kapitel 7 bewertet anschließend die Ergebnisse kritisch.

## 6.1. Datenvorverarbeitung

### 6.1.1. Statistische Analyse der Aussortierungsgründe

Von allen gesammelten Messdaten sortieren die einzelnen Phasen der Datenvorverarbeitung insgesamt ca. 24% der Datensätze aus. In folgendem Abschnitt soll dargelegt werden, welche Gründe statistisch betrachtet die Aussortierungen bedingen und in welchen Phasen diese erfolgen. Abschnitt 7.1 auf Seite 75 diskutiert daran anschließend die generelle Effektivität der Datenvorverarbeitung.

Die in Kapitel 4 vorgestellte Pipeline zur Datenvorverarbeitung ist so konzipiert, dass lediglich in den Phasen der statischen Präfilterung und der metrikbasierten Filterung mit dynamisch bestimmten Schwellwerten Datensätze aussortiert werden. Daher ist eine Betrachtung dieser Phasen für die Analyse der Aussortierungsgründe ausreichend, Tabelle 6.1 auf Seite 66 visualisiert eine Übersicht dieser Gründe. Für die Phase der Präfilterung bedingt eine Auffälligkeit in einem der Kriterien direkt die Aussortierung des Datensatzes, weshalb die Anteile in der Tabelle die Anteile an den aussortierten Datensätzen aufgrund der jeweiligen Gründe widerspiegeln. In der Phase der metrikbasierten Filterung erfolgt die Entscheidung der Aussortierung anhand einer Gewichtung der Intensitäten verschiedener Auffälligkeiten. Deshalb geben die Anteile bei deren Gründen die Anteile an allen Messdaten an, nicht an den aussortierten.

In der Phase der **statischen Präfilterung** werden 15.6% aller Datensätze aussortiert. Für fast 80% dieser gefilterten Datensätze ist das Kriterium, keine NaN-Werte im Handgelenk zu besitzen, ausschlaggebend für die Aussortierung. Daraus lässt sich schließen, dass insgesamt betrachtet mehr als jeder zehnte aufgenommene Datensatz auch nach

Tabelle 6.1.: Statistische Analyse der Aussortierungsgründe von Datensätzen

Phase	Anteil an allen Datensätzen	Grund	Anteil
<b>Präfilterung</b>	15.6%	Nan-Werte im Handgelenk	79.8%
		Datensatzlänge (zu kurz)	13.1%
		Datensatzlänge (zu lang)	4.9%
		Maximalgeschwindigkeit	2.2%
<b>Filterung</b>	8.4%	Ellbogen-Handgelenk-Distanz	12.9%
		Handgelenk-Handgelenk-Distanz	6.5%
		z-Koordinaten-Differenz	5.0%
		manuelle Aussortierung	21.5%

Interpolation und Beschneidung auf den relevanten Bereich noch Nan-Werte in den Handgelenkdaten aufweist. 13% der gefilterten Datensätze sind zu kurz, um für das Maschinelle Lernen sinnvoll verwendet zu werden. 5% sind zu lang, was auf fehlerhafte Beschneidung hindeutet, und lediglich 2% der in der Präfilterung aussortierten Daten weisen zu hohe Maximalgeschwindigkeiten auf. Hierzu muss ergänzt werden, dass die statische Filterung in dieser Reihenfolge erfolgt. Das heißt, dass beispielsweise ein Datensatz mit Nan-Werten im Handgelenk direkt aussortiert wird und nicht mehr auf Länge oder Maximalgeschwindigkeit überprüft wird.

Die Phase der **metrikbasierten Filterung** sortiert weitere 8.4% von allen Datensätzen aus. In dieser Phase fällt auf, dass ca. 12% aller Daten auffällige Werte in der Ellbogen-Handgelenk-Distanz aufweisen. Über 60% dieser Ausreißer sind zudem so stark, dass sie eine durchschnittliche Handgelenk-Ellbogen-Distanz von über 30cm besitzen und als extreme Ausreißer bereits vor der dynamischen Schwellwertbestimmung statisch aussortiert werden. Dieser Anteil der extremen Ausreißer ist bei den anderen Metriken deutlich geringer.

Zudem sollte noch der hohe Anteil manueller aussortierter Daten analysiert werden. Wie in Abschnitt 4.8 erläutert wird zur Bewertung der Effektivität der Pipeline und Einstellung der Hyperparameter der Metriken eine manuelle Aussortierung von ca. 1000 Datensätzen anhand ihrer Geschwindigkeitsprofile vorgenommen. Dabei fallen 21.5% der betrachteten Datensätze als fehlerbehaftet auf. Über zwei Drittel dieser manuell aussortierten Daten sortiert die Phase der Präfilterung in der automatisierten Datenvorverarbeitung ebenfalls direkt aus, weitere 4% der manuell aussortierten Daten eliminiert die Phase der metrikbasierten Filterung. Die weiteren 32% der manuell aussortierten

Daten werden allerdings nicht auch automatisch aussortiert. Gründe hierfür sollen in Abschnitt 7.1 diskutiert werden.

Im Rahmen der Rotation werden vereinzelt weitere Datensätze aussortiert, da lediglich Datensätze, die mehr Werte nach dem gewählten Rotationspunkt aufweisen, rotiert werden können. Das betrifft ca. 3% aller Datensätze. Diese Datensätze werden nicht in der Phase der Präfilterung bereits aussortiert, da sie beispielsweise beim Handgelenk-Handgelenk-Distanz in der Phase der metrikbasierten Filterung als zweite Körperseite noch sinnvoll verwendet werden können.

### 6.1.2. Spiegelung

Um die Menge an Trainingsdaten zum Training des Neuronalen Netzes zu erhöhen, werden die Datensätze der linken Körperhälfte so gespiegelt, dass diese sich wie Bewegungsabläufe der rechten Körperhälften verhalten. Das Verfahren entspricht dem in Abschnitt 4.6.3 beschriebenen: Wenn die Datensätze aus rechter und linker Körperhälfte vorhanden sind, soll die Spiegelebene danach ausgerichtet werden. Ist nur die linke vorhanden, wird die Spiegelebene zwischen Start- und Endpunkt der Bewegung eingezogen. Zum Spiegeln wird nur die Trajektorie des Handgelenks betrachtet; Ellbogen und Schulter spiegeln sich an der Ebene des Handgelenks, um die Abstände zwischen den Gelenken nicht zu verändern.

Insgesamt können von allen zu spiegelnden Datensätzen etwa 90% anhand beider Körperhälften und jeder zehnte lediglich anhand der linken Körperhälfte gespiegelt werden. In den vorliegenden Daten gibt es keinen Datensatz, der zu wenig Datenpunkte für die Auswahl einer geeigneten Spiegelachse enthält. Ein Beispiel für einen gespiegelten Datensatz anhand beider Körperhälften ist in Abbildung 6.1 zu sehen. Die Spiegelachse ist ein wenig zu stark geneigt, sodass sich die Trajektorien der Schultergelenke überlagern. Im Idealfall hätte die Bahnkurve des Ellbogens und der Schulter des originalen zum gespiegelten Datensatz denselben Abstand wie die Handgelenke.

Die Trajektorien der gespiegelten linken Ellbogen und Schultern verlaufen nach der Spiegelung teilweise nicht dort, wo sich auch die entsprechend rechten Ellbogen und Schultern befinden, sondern sind räumlich rotiert und verschoben, wie in Abbildung 6.2 visualisiert ist. Dort ist erkennbar, dass das gespiegelte linke Handgelenk vom Verlauf und

der Position her annähernd mit dem rechten Handgelenk übereinstimmt, der Ellbogen und die Schulter hingegen an anderen Positionen im Raum sind.

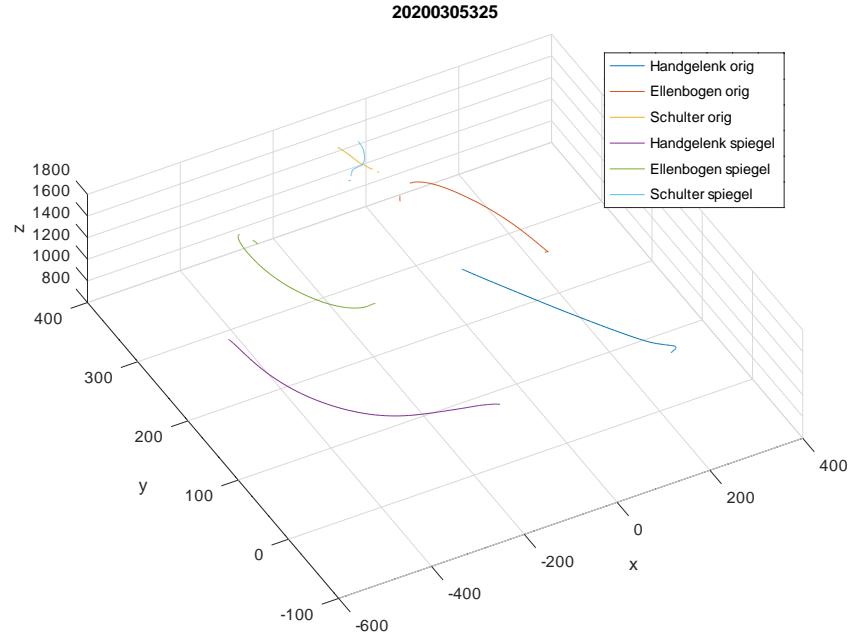


Abbildung 6.1.: Vergleich des originalen mit dem gespiegelten Datensatz

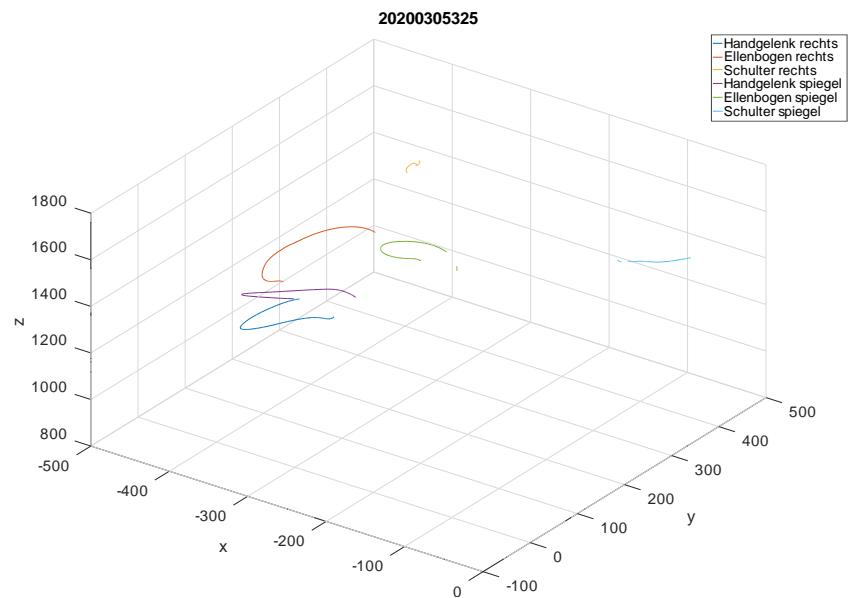


Abbildung 6.2.: Trajektorien der gespiegelten linken Körperhälfte im Vergleich zu den originalen Verläufen der rechten Körperhälfte

### 6.1.3. Rotation

Die beiden Abbildungen 6.3 und 6.4 zeigen die Auswirkung der Rotation auf die Ausrichtung der Trajektorien im Raum. In Abbildung 6.3 ist zu erkennen, dass die Trajektorien vor der Rotation ungleichmäßig im Raum verteilt in verschiedene Richtungen zeigen. Abbildung 6.4 visualisiert Trajektorien nach erfolgter Rotation, die Vereinheitlichung der Richtungen der Anfangsbewegungen ist klar ersichtlich. Dennoch existieren auch nach der Rotation noch einige Trajektorien, die in ungewöhnliche Richtungen zeigen. Diese fehlerbehafteten Trajektorien sind vermutlich falsch beschnitten, da sie trotz Rotationspunkt auf der x-Achse teilweise sogar in entgegengesetzte Richtung zeigen.

Der Vorteil der Vereinheitlichung zeigt sich in einer Verbesserung der Trainingsergebnisse bei sonst gleichen Hyperparametern und Modellkonfiguration. Gemessen an der durchschnittlichen euklidischen Distanz der vorhergesagten zu den tatsächlichen Endpunkten verbessert sich der Wert von ca. 23.8cm auf durchschnittlich 13.7cm Genauigkeit bei den Testdaten deutlich, die Genauigkeit der Trainingsdaten verbessert sich von auf 15.7cm auf ca. 10.7cm. Somit gilt die in Abschnitt 4.9 formulierte These, eine Vereinheitlichung der Anfangsrichtungen verbessere die Generalisierungsfähigkeit des neuronalen Netzes, als bestätigt. Die signifikante Verbesserung des Ergebnisses in den Testdaten zeigt die gesteigerte Generalisierungsfähigkeit.

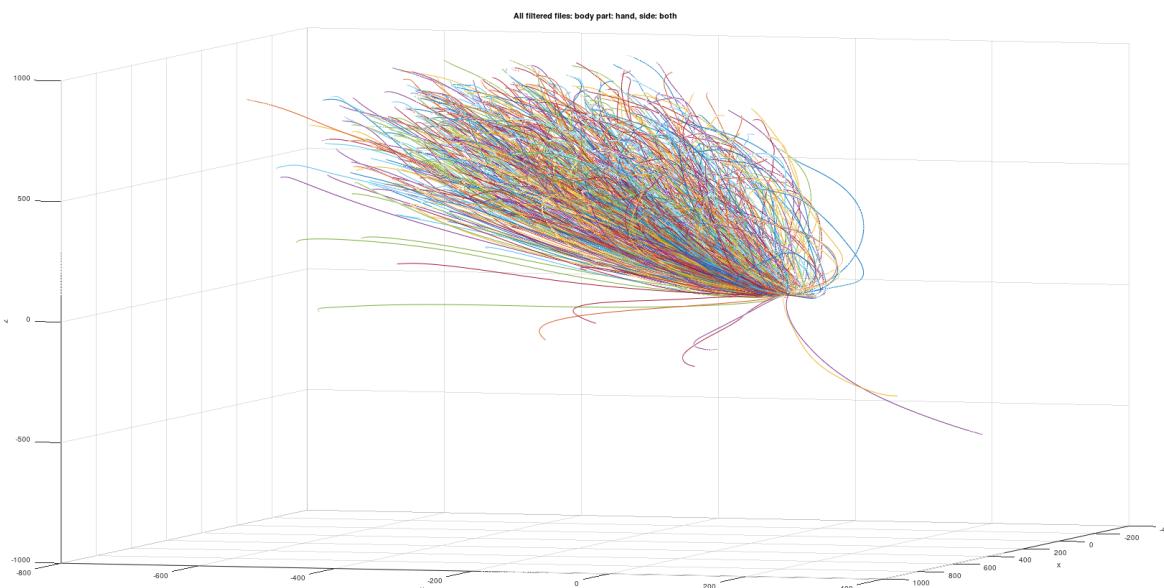


Abbildung 6.3.: Visualisierung aller Handgelenktrajektorien nach der Phase der Filterung

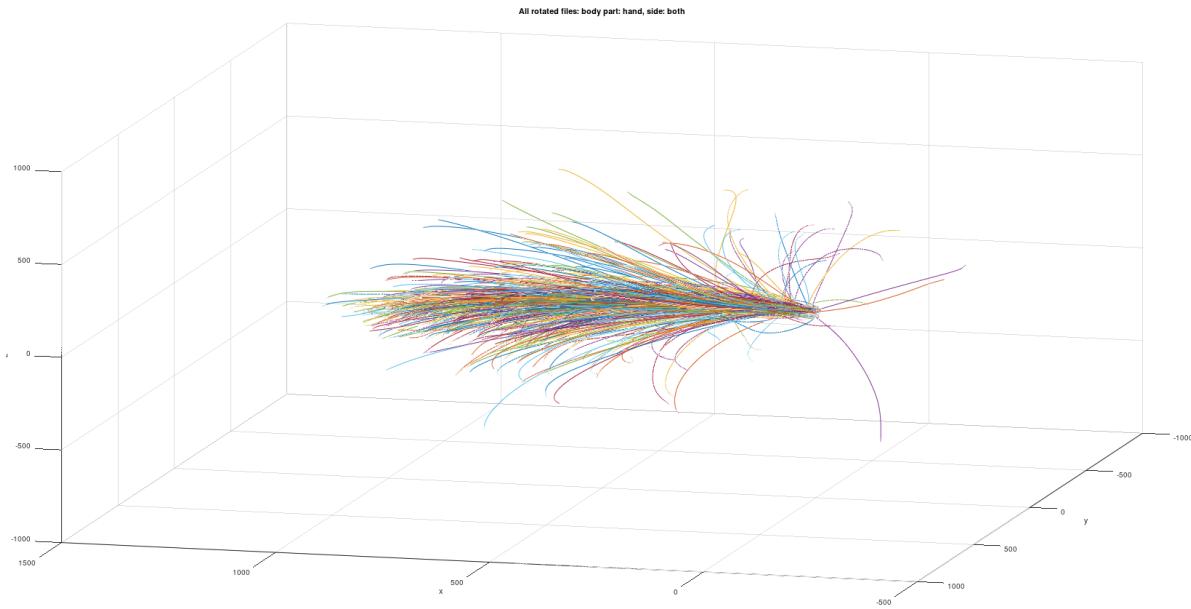


Abbildung 6.4.: Visualisierung aller Handgelenktrajektorien nach der Phase der Rotation

## 6.2. Machine Learning

### 6.2.1. Hyperparameter

In diesem Abschnitt sollen die nach den in Abschnitt 5.2.5 vorgestelltem empirischen Verfahren bestimmten Werte der Hyperparameter exemplarisch am Neuronalen Netz zur Vorhersage der Endpositionen vorgestellt werden.

Als bestmögliche Konfiguration erweist sich eine Anfangsbewegung der Länge 85, somit werden durchschnittlich 65% der Trajektorie als Eingangsdaten zur Vorhersage der Endposition benötigt. Der Rotationspunkt, nach dem sich die Rotation ausrichtet, liegt dabei beim 70. Punkt der Trajektorie. Als Epochenzahl des Trainings stellt sich 350 als sinnvolle Größe heraus, da bei dieser Epochenzahl das Neuronale Netz bei Testdaten ähnlich gute Ergebnisse wie bei Trainingsdaten erzielt und noch keine starke Überanpassung auf die Trainingsdaten vorliegt. Als Batch-Size lässt sich 250 festlegen, von den Trainingsdaten werden 25% als Testdaten und 5% als Validierungsdaten verwendet.

### 6.2.2. Trainingsergebnisse

Dieser Abschnitt beschreibt die erreichten Trainingsergebnisse exemplarisch an der Vorhersage der Endpositionen sowie ihre Qualität anhand der in Abschnitt 5.2.4 definierten drei Qualitätskriterien der Metriken, dem Visualisierungseindruck sowie stichprobenartigen Überprüfungen.

Bezüglich der Metrikergebnisse können bei bereits beschriebener Trainingskonfiguration die Endpunkte durchschnittlich bei den Trainingsdaten auf 10.7cm und bei den Testdaten auf 13.7cm genau vorhergesagt werden, die Werte ergeben sich als durchschnittliche euklidische Distanz der letztendlich vorhergesagten und tatsächlichen Endpunkte. Bewegungsduern können auf bis zu 10 Zeitpunkten bei Trainings- und Testdaten vorhergesagt werden.

Die Abbildung A.2 auf Seite XIV im Anhang visualisiert als zweites Qualitätskriterium wie in Abschnitt 5.2.4 beschrieben die Trainingsergebnisse, Abbildung A.3 auf Seite XV im Anhang die Testergebnisse. Die rote Ursprunggerade stellt in den Abbildungen den optimalen Lernerfolg dar, Punkte innerhalb der gelben Gerade liegen im sehr guten Bereich und maximal 10cm von einer perfekten Vorhersage in der jeweiligen Koordinate entfernt. In allen Koordinaten lässt sich ein klarer Lernerfolg feststellen, das Trainingsergebnis könnte dabei vor allem in der y-Koordinate verbessert werden.

Als drittes Qualitätskriterium soll stichprobenartige die absoluten Abweichungen der vorhergesagten Endpunkte von den tatsächlichen betrachtet werden. Dazu werden drei Datensätze aus den Testdaten zufällig ausgewählt und die absolute Differenzen je Koordinate berechnet, was Tabelle 6.2 darstellt.

Um eine graphische Vorstellung der Ergebnisqualität zu erlangen, visualisiert Abbildung 6.5 auf Seite 72 stichprobenweise eine Teilmenge von 20 der vorhergesagten und tatsächlichen Endpunkte in einem kartesischen Koordinatensystem. Der vorhergesagte und tatsächliche Endpunkt sind dabei durch eine Linie verbunden.

Tabelle 6.2.: Stichprobenartige Prüfung der Trainingsqualität, die Angaben der Werte sind in Millimetern

		x-Koordinate	y-Koordinate	z-Koordinate
<b>Stichprobe 1</b>	Realität	698.31	-39.03	-231.58
	Vorhersage	797.84	-18.89	-325.03
	<i>abs. Differenz</i>	99.53	20.14	93.45
<b>Stichprobe 2</b>	Realität	464.61	57.24	-45.21
	Vorhersage	594.51	74.22	-123.26
	<i>abs. Differenz</i>	129.91	16.98	78.05
<b>Stichprobe 3</b>	Realität	797.96	177.98	-71.02
	Vorhersage	826.79	223.04	27.89
	<i>abs. Differenz</i>	28.83	45.06	98.91

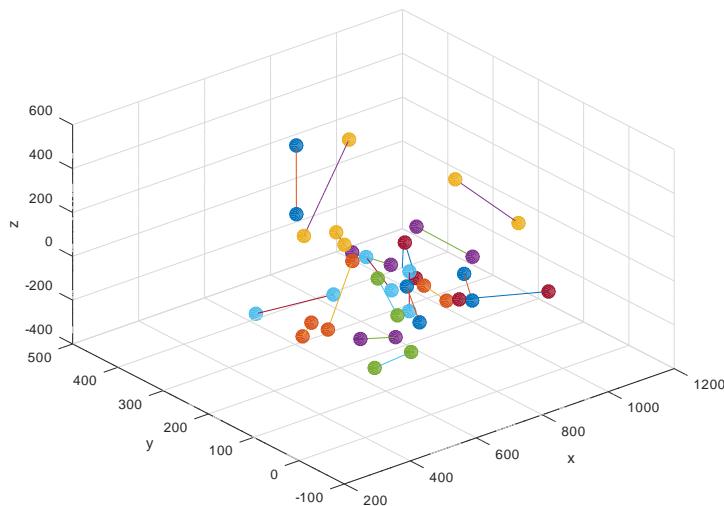


Abbildung 6.5.: Visualisierung exemplarischer Ergebnisse in Koordinatensystem

### 6.3. Pfadberechnung

Die Berechnung der Bahnkurve durch das Minimum Jerk Model für gekrümmte Punkt-zu-Punkt Bewegungen anhand der Anfangsbewegung und dem vorhergesagten Endpunkt und Dauer ergibt einen ähnlichen Verlauf im Vergleich zur tatsächlichen Trajektorie. In Abbildung 6.6 auf Seite 73 ist eine beispielhafte Gegenüberstellung zu sehen. Abgebildet ist zum einen der Verlauf der im Versuch aufgenommenen Trajektorie in blau und zum anderen die anhand dieser Daten und der Vorhersagen des Neuronalen Netzes berechneten Bahnkurve in orange.

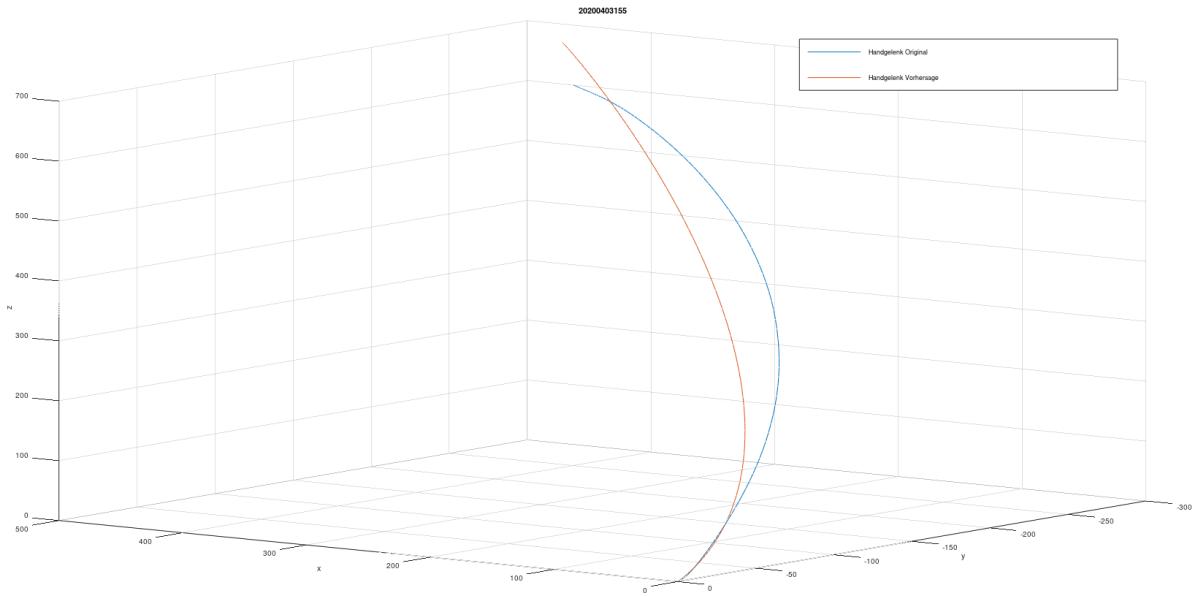


Abbildung 6.6.: Vergleich der tatsächlichen (blau) zur berechneten Trajektorie (orange)

Bei der Betrachtung der berechneten Geschwindigkeitsprofile ist zu sehen, dass der grobe Verlauf der Glockenform und deren maximale Amplitude mit denen der tatsächlichen Bewegung übereinstimmen. Auch der Positionsverlauf ist ähnlich. In Abbildung 6.7 sind die dimensionsweisen Verläufe der Positionen und Geschwindigkeiten von der tatsächlichen (oben) und der berechneten Bewegung (unten) gegenübergestellt.

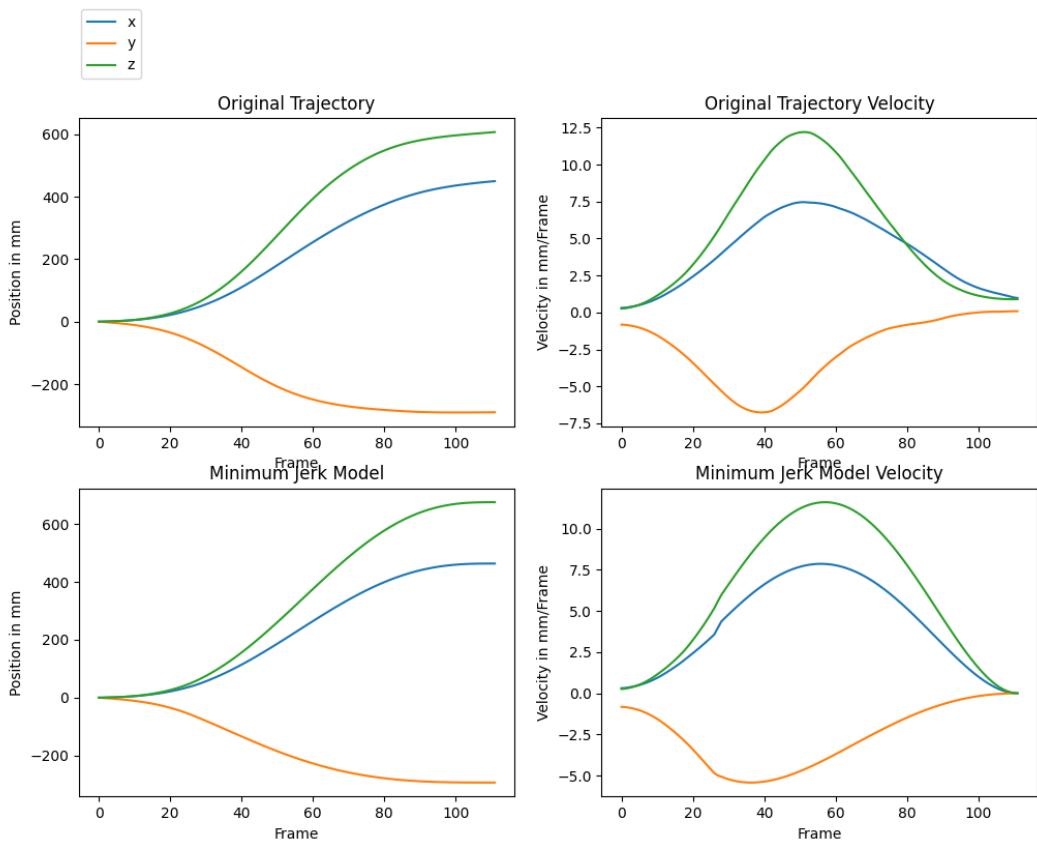


Abbildung 6.7.: Gegenüberstellung der Positions- und Geschwindigkeitsprofile der tatsächlichen und der berechneten Bewegung

# 7. Diskussion

Dieses Kapitel dient der Diskussion der in Kapitel 6 vorgestellten Ergebnisse. Hierbei soll neben der Datenvorverarbeitung in Abschnitt 7.1 und dem Maschinellen Lernen in Abschnitt 7.2 auch der Fokus auf die Praxistauglichkeit der erarbeiteten Ergebnisse ab Seite 78 gelegt werden.

## 7.1. Datenvorverarbeitung

In diesem Abschnitt soll die Qualität der Rohdaten und die Effektivität der Pipeline zur Datenvorverarbeitung diskutiert werden.

### 7.1.1. Qualität der Rohdaten

Allgemein ist auffällig, dass während der Datenvorverarbeitung jeder vierte Datensatz aufgrund mangelnder Datenqualität aussortiert werden muss. Für fast die Hälfte dieser aussortierten Datensätze, also ca. 12% aller Daten, ist das Vorliegen von NaN-Werten in den Trajektorien der Handgelenk die Ursache für die Filterung. Deshalb lässt sich feststellen, dass NaN-Werte in den Messdaten ein grundlegendes Problem sind. Würden zur Aussortierung nicht nur die NaN-Werte der Handgelenkbewegungen, sondern auch der Ellbogen und Schulterbewegung betrachtet werden, so müssten über 25% aller Daten allein aufgrund von NaN-Werten aussortiert werden. Ohne die zuvor erfolgte Interpolation kurzer NaN-Strecken, wäre der Anteil vermutlich noch höher. Jedoch ist die Anzahl der NaN-Werte nur ein Kriterium für qualitativ hochwertige Daten, sodass vermutlich bei Betrachtung von Handgelenk, Ellbogen und Schulter und allen Filterkriterien deutlich mehr als 25% der Messdaten für das Training aussortiert würden.

Diese Beobachtung gibt Anlass, die Ursachen der NaN-Werte in den aufgenommenen Datensätzen zu erörtern und die Aufnahmequalität der Daten zu verbessern.

Bezüglich der Datenqualität der aufgenommenen Daten fällt auch auf, dass im Vergleich von Handgelenk- und Ellbogendaten vor allem die Ellbogendaten besonders fehlerbehaftet

sind. Dies wird durch den hohen Anteil der in der Ellbogen-Handgelenk-Metrik markierten Datensätze sichtbar. Zudem werden bei den Ellbogendaten wie in Abschnitt 6.1.1 auf Seite 65 vorgestellt über 60% der Ausreißer bereits statisch als extreme Ausreißer mit einem Metrikwert von  $\gamma_1 > 300\text{mm}$  markiert. Der Anteil extremer Ausreißer an allen Ausreißern einer Metrik ist bei den anderen beiden Metriken wesentlich geringer. Deshalb sollte vor allem auch die Aufnahmeequalität der Ellbogendaten deutlich gesteigert werden. Über die Qualität der Schulterdaten kann an dieser Stelle keine Einschätzung getätigt werden, da diese in der vorliegenden Arbeit nicht verwendet werden.

Darüber hinaus lässt sich beobachten, dass zu etwa 15% der aufgenommenen Datensätze nur eine Körperseite vorliegt. Dies erschwert die Spiegelung der Daten und verhindert die Anwendung mancher Metriken, die sich auf beide Körperhälften beziehen. Auch diese Beobachtung sollte weitergehend untersucht werden.

### 7.1.2. Beschneidung

Die in dieser Arbeit definierte Metrik zur Bestimmung der Güte der Beschneidung von Datensätzen ist lediglich auf einen empirisch ermittelten Wert begründet. Wie in Abschnitt 4.4 ausgeführt, ist eine Definition der Metrik über statistische Daten der Länge der Datensätze schwierig zu begründen. Daher sollte auch die Festlegung der Parameter auf die in diesem Kapitel angegebenen Werte kritisch hinterfragt werden. Eine Aussage über die tatsächliche Güte der Datensätze nach der Beschneidung findet nur durch eine manuelle Analyse der Geschwindigkeitsverläufe und der Trajektorien statt. Diese entspricht allerdings nicht dem Gedanken der vollständigen Automatisierung der Datenvorverarbeitung und ist mit erheblichem Aufwand verbunden.

Denkbar ist eine weitere automatische Analyse der Datensätze, die über einen einheitlich festgelegten ausreichend großen Geschwindigkeitsschwellwert die Beschneidung der Datensätze testet. Zum Beispiel kann damit bestimmt werden, wie lange die Bewegung vor und nach diesem Schwellwert andauert; ist diese zu groß, muss weiter beschnitten werden. Durch diese dynamische Metrik ergibt sich auch eine verteilte Länge der Datensätze.

### 7.1.3. Spiegelung

Eine Verbesserung der Spiegelung kann erreicht werden, wenn die Spiegelachse nicht nur anhand den Handgelenkdaten sondern auch anhand der Ellbogen- und Schulterdaten ermittelt wird. Allerdings ist der vorliegende Datenbestand im Bereich der Ellbogen und Schultergelenke zu unzuverlässig, sodass diese Methode weiteren aufwendigen Prüfmechanismen in Form von Analysen der Geschwindigkeitsverläufe oder Ähnlichem bedarf, welche die Qualität der Spiegung bewerten.

### 7.1.4. Filterung

Anhand der manuell aussortierten Datensätze lässt sich teilweise die allgemeine Effektivität der automatisierten Datenaussortierung bewerten. Wie in Abschnitt 4.8.3 auf Seite 51 erläutert, zieht die manuelle Aussortierung als Anhaltspunkte für fehlerbehaftete Daten vor allem Geschwindigkeitsprofile mit Sprüngen oder harten Kanten oder ungewöhnlicher Form heran. Die automatisierte Aussortierung verwendet dagegen statisch Kriterien, wie Anzahl der NaN-Werte, Länge der Dateien oder die Maximalgeschwindigkeit, und stellt auffällige Abstände im Raum, wie zwischen Ellbogen und Handgelenk oder zwischen beiden Handgelenken, fest. Daher ist eine vollständige Übereinstimmung der Menge der manuell aussortierten mit der automatisch aussortierten Datensätze nicht erwartbar. Dennoch sollte die Schnittmenge relativ groß sein.

Die automatisierte Präfilterung erfüllt diese Erwartung; 88% der dort aussortierten Datensätze werden auch manuell aussortiert. Im Bereich der Filterung unterscheiden sich die Mengen der aussortierten Datensätze allerdings deutlich. Lediglich ca. 12% der in der Filterung aussortierten Daten werden auch manuell aussortiert. Diese Abweichung lässt sich anhand der unterschiedlichen Anhaltspunkte für fehlerbehafteten Daten erklären. Dennoch ist wünschenswert die Kriterien der manuellen Filterung auch in die automatisierte Pipeline zu integrieren, um auch automatisiert möglichst viele fehlerbehaftete Datensätze herauszufiltern. Eine Stetigkeitsanalyse der Geschwindigkeitsverläufe oder Ähnliches ist bislang nicht in der Datenvorverarbeitung enthalten.

Außerdem lässt sich die Aussagekraft der gewählten Metriken diskutieren. In stichprobenartigen Visualisierungen als Ausreißer markierter Datensätze fällt auf, dass zahlreiche Ausreißer bezüglich der Handgelenk-Handgelenk-Distanz keine visuell erkennbaren Fehler

aufweisen. Allerdings gilt auch zu diskutieren, inwiefern die Fehlerhaftigkeit eines Datensatzes anhand der dreidimensionalen Visualisierung des Trajektorienverlaufs manuell erkannt werden kann. In der visuellen Aussortierung kann lediglich die Glattheit der Kurve und ungewöhnliche Gesamtverläufe beurteilt werden. Generell fällt auf, dass von einer Metrik aussortierte Datensätze oft visuell beurteilt nicht auffällig sind. Die visuell auffälligen Datensätze werden meist in der Phase der Präfilterung direkt aussortiert. Es stellt sich die Frage, ob ein Datensatz mit glattem, gewöhnlich wirkendem Kurvenverlauf und ungewöhnlichen Metrikwerten fehlerbehaftet ist oder nicht.

## 7.2. Machine Learning

Es lässt sich der generelle Erfolg des Maschinellen Lernens diskutieren, da fast zwei Drittel der Trajektorie als Eingangsdaten benötigt werden, um den Endpunkt und die Dauer vorherzusagen. Zudem ist eine Vorhersage der Dauer mit einer Präzision von 10 Zeitpunkten unter der Betrachtung, dass aufgrund der statischen Präfilterung noch maximal 90 Zeitpunkte, durchschnittlich jedoch lediglich 45 weitere Messpunkte folgen, relativ unpräzise.

Da der Fokus dieser Arbeit allerdings auf der Datenvorverarbeitung liegt, ist dieses Ergebnis akzeptabel. Wie in Abschnitt 6.3 auf 72 exemplarisch gezeigt lassen sich trotz verbesserungsfähiger Präzision gute Ergebnisse in der Berechnung der weiteren Trajektorie erzielen. Zudem gilt zu beachten, dass in der Praxis zur Kollisionsvermeidung ohnehin ein Sicherheitsbereich um die berechnete Trajektorie gebildet werden muss. Dadurch lassen sich Ungenauigkeiten in der Berechnung etwas ausgleichen. Wie gut sich diese Methodik eignet, sei in den Ausblick gestellt.

## 7.3. Praxistauglichkeit

Dieser Abschnitt soll die Anwendbarkeit der Ergebnisse in der Praxis erörtern.

Die Koordinaten der ersten 85 Zeitpunkte dienen als Eingangsdaten für das neuronale Netz, müssen allerdings zuvor zur Vergleichbarkeit vorverarbeitet werden und dazu einen Teil der Datenpipeline durchlaufen. Hierbei besteht das Problem, dass diese Verarbeitung in Echtzeit erfolgen muss, da durchschnittlich auf den 85. Messwert noch ca. 45 folgen, bis

die Bewegung die Endposition erreicht. Bei einem verwendeten Kamerasystem mit 120Hz bleiben folglich zur Datenverarbeitung und Endpunktvorhersage lediglich 375ms Zeit. Deshalb müssen die Phasen der Datenvorverarbeitung priorisiert werden. Abbildung 7.1 zeigt als Flussdiagramm den vollständigen Ablauf der Transformationen in der praktischen Anwendung.

Höchste Priorität haben dabei Phasen, die der Vereinheitlichung der Daten dienen, wie die **Relokation** auf den Ursprung und **Rotation** der Anfangsrichtung in x-Achsenrichtung, da ohne diese Phasen die gemessenen Eingangsdaten nicht mit den Trainingsdaten vergleichbar sind. Bei Daten der linken Körperhälfte ist auch die Phase der **Spiegelung** unabdingbar. Die Beschneidung der Daten, sodass diese zu ähnlichen Zeitpunkten starten und vergleichbar sind, lässt sich anhand der Anfangsbewegung nicht durchführen. Die Ursache hierfür besteht darin, dass das in 4.4 vorgestellte Verfahren zur Beschneidung den vollständigen Geschwindigkeitsverlauf der Trajektorie benötigt. Deshalb muss durch die Sensortechnik sichergestellt werden, dass die Bewegung unmittelbar zum Beginn der 85 Datenwerte startet.

In Priorität zwei sind Verarbeitungsschritte, die lediglich die Datenqualität steigern. Darunter fallen beispielsweise Interpolation, Glättung, Präfilterung und Filterung der Daten. Sensorfehler können jederzeit auftreten, weshalb Interpolation und Glättung in der Praxis auch wichtig sind. Sollte eine Messung in der praktischen Anwendung so fehlerbehaftet sein, dass sie aussortiert werden würde, muss eine Meldung ausgegeben werden. Die Endposition kann in diesem Fall nicht vorhergesagt werden und könnte überall sein.

Nach Vereinheitlichung der Daten können diese auf das Intervall von 0 bis 1 skaliert und als Eingang für die Neuronalen Netze verwendet werden. Anhand der im Training erlernten Gewichte sagen diese die Endposition und Dauer der Trajektorie voraus, die anschließend zunächst reskaliert und dann im Falle der Endposition in das Koordinatensystem der Realität retransformiert werden muss. Die Retransformation beinhaltet die Rotation der Endposition zurück in die ursprüngliche Richtung der Trajektorie. Hierfür kann dieselbe Drehachse wie zuvor verwendet werden, lediglich das Vorzeichen des Drehwinkels muss invertiert werden. Daraufhin erfolgt die Translation der Endposition mit den entgegengesetzten Werten der Relokation. So befindet sich die vorhergesagte Endposition wieder im ursprünglichen Koordinatensystem und kann mit der vorhergesagten Dauer zu

Berechnung der weiteren Trajektorie anhand des Minimum Jerk Models für gekrümmte Punkt-zu-Punkt-Bewegungen eingesetzt werden.

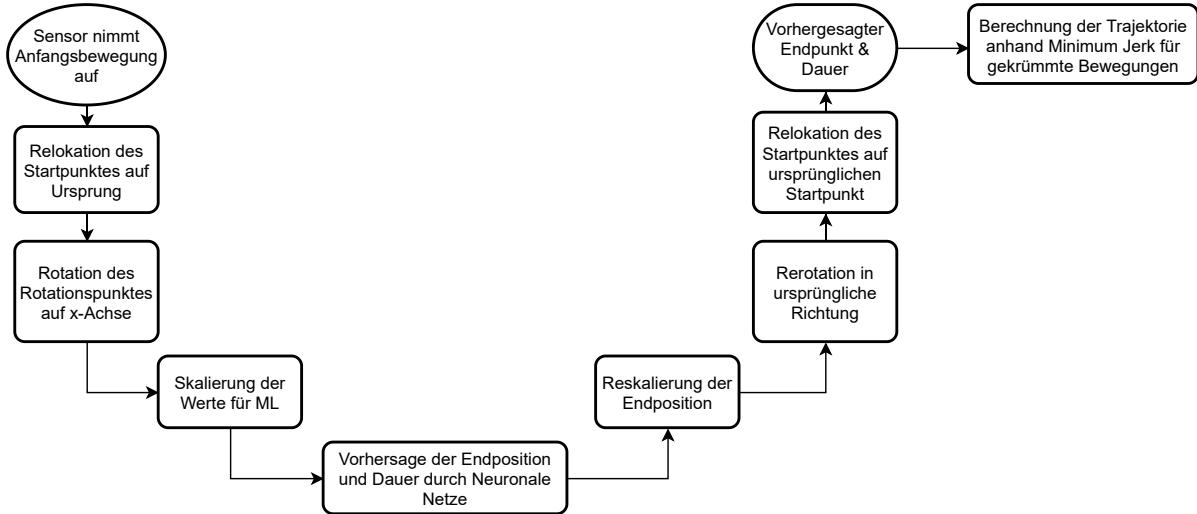


Abbildung 7.1.: Ablauf zur Anwendung in der Praxis

In Tests können die beschriebene Verarbeitungsschritte der höchsten Priorität sowie die Vorhersage des Endpunkts und parallel der Dauer sowie anschließende Retransformation in durchschnittlich 52ms, d.h. innerhalb der theoretisch zur Verfügung stehenden 375ms durchgeführt werden.

Die Berechnung der vorherzusagenden Bahnkurve der Hand über das Minimum Jerk Model nimmt in Tests mit etwa 250ms viel Zeit in Anspruch. Über eine numerische Optimierung der Gleichung und der Berechnung sowie Approximationen, wie beispielsweise eine Betrachtung jedes zehnten Frames, ließe sich die Berechnungszeit auf schätzungsweise ein Zehntel davon minimieren. Die gesamte Vorausberechnung der Trajektorie der Hand liegt damit ungefähr bei 80ms exklusive der Zeit, die der Roboterarm benötigt, um Konsequenzen aus diesen Informationen zu berechnen.

Somit ist eine praktische Anwendung der Vorhersage möglich, sofern weitere Performanceoptimierungen für schnellere Verarbeitungszeiten erfolgen.

# 8. Fazit & Ausblick

## 8.1. Fazit

Zusammenfassend zeigt die vorliegende Arbeit ein Gesamtkonzept zur Vorausberechnung von Handbewegungen auf. Hierbei deckt die Arbeit von der Datenaufnahme, über die Datenvorverarbeitung und dem letztendlichen Training eines Neuronalen Netzes zur Vorhersage von Endpunkt und Dauer von Handbewegungen alle wichtigen Phasen des Maschinellen Lernens ab. Dieser Prozess wird zudem durch die anschließende mathematische Berechnung der Trajektorie sinnvoll in das praktische Anwendungsszenario der Kollisionsvermeidung in der kollaborativen Robotik eingebettet.

Besonders hervorzuheben ist die entwickelte modulare Pipeline zur automatisierten Datenvorverarbeitung, die über zahlreiche Schritte verrauschte, fehlerbehaftete Messdaten zu vereinheitlichten, qualitativ hochwertigen Trainingsdaten transformiert. Durch die transparente Definition der Pipeline ist eine weitere Verwendung dieser denkbar.

## 8.2. Ausblick

Dieser Abschnitt soll einen Ausblick über mögliche Erweiterungen und Fortführungen der vorliegenden Arbeit geben und geht dabei auf die Themenbereiche der Datenvorverarbeitung, des Maschinellen Lernens und der möglichen praktischen Anwendung ein.

Die **Datenvorverarbeitung** kann weiter verbessert und optimiert werden. Zum einen ist eine grafische Filterung fehlerbehafteter Trajektorien denkbar, da wie in Abbildung 6.3 auf Seite 69 zu sehen in der Visualisierung aller Trajektorien Ausreißer leicht erkennbar sind. Zum anderen lässt sich überlegen, die Filterung um weitere Kriterien wie beispielsweise die Glattheit der Kurven zu erweitern, um die Differenz von manuell und automatisch aussortierten Datensätzen zu verringern.

Grundsätzlich bietet sich auch Maschinelles Lernen zur Aussortierung von fehlerbehafteten Rohdaten an. Anhand der durch die Pipeline und durch manuelle Filterung erzeugten

gelabelten Daten ließe sich in einem überwachten Lernverfahren zu diesem Zweck ein Klassifikator trainieren. Dadurch kann die Filterungsfähigkeit der Datenvorverarbeitung verbessert werden, Phasen zur Vereinheitlichung oder Steigerung der Datenqualität wie Glättung, Beschneidung oder Spiegelung sind jedoch weiterhin notwendig.

Die **Neuronalen Netze** zur Vorhersage der Endpositionen und Dauern können ebenfalls weiter verbessert werden. So ist eine Verfolgung des in Abschnitt 5.1.2 beschriebenen Klassifikationsansatzes vor allem für die Vorhersage der Bewegungsdauern denkbar. Auch die Verwendung von Recurrent Neural Networks oder Convolutional Neural Networks (CNNs) kann als Alternative zur den verwendeten Dense-Layern zur Vorhersage der Endpositionen herangezogen werden.

Auch eine **Anwendbarkeit in der Praxis** ohne markerbasierte Handgelenkserkennung durch ein Kamerasystem und Software wie *OpenPose*, einem Open-Source-Echtzeitsystem zur 2D-Positionserkennung von anatomischen Schlüsselpunkten wie beispielsweise Handgelenken [33], ist denkbar. Die 2D-Positionsdaten müssten beim aktuell trainierten Modell allerdings durch mehrere Kameras in 3D-Positionsdaten umgerechnet werden.

Darüber hinaus kann die Möglichkeit der Aufnahme von Handbewegungen über eine Kinect-Kamera wie in [34] beschrieben für eine bessere Praxistauglichkeit weiter verfolgt werden.

Dennoch lässt sich der generelle Ansatz der Kollisionsvermeidung durch Vorhersage der Bewegungsabläufe des Menschen und anschließenden Korrektur der Roboterbewegung kritisch betrachten. Der Ansatz basiert auf der Annahme, dass der Mensch seine Handbewegung unabhängig von Einflüssen der Umwelt deterministisch durchführt. Menschen reagieren allerdings auf äußere Veränderungen, wie beispielsweise eine Routenanpassung oder Pausierung der Roboterbewegung, und können daraufhin ihre Bewegung unmittelbar auch ändern, sodass sie nicht mehr der vorausberechneten Trajektorie entspricht.

Dieser Rückkopplungseffekt muss bei weiteren Überlegungen und dem praktischen Einsatz des Verfahrens bedacht werden.

# Literaturverzeichnis

- [1] Keibel, A. „Mensch-Roboter-Kollaboration in der Medizin“. In: *Mensch-Roboter-Kollaboration*. Hrsg. von Buxbaum, H.-J. Wiesbaden: Springer Fachmedien Wiesbaden, 2020, S. 133–143.
- [2] Markis, A. u. a. „Sicherheit in der Mensch-Roboter-Kollaboration“. In: *White Paper. Fraunhofer Austria* (2016).
- [3] Oliver Rettig, Silvan Müller, Marcus Strand. *Determination of posture comfort zones for robot-human handover tasks*.
- [4] Murray, I. A. „Determining upper limb kinematics and dynamics during everyday tasks“. Newcastle: Newcastle University, 1999.
- [5] *GNU Octave*. 2021. URL: <https://www.gnu.org/software/octave/index> (Einsichtnahme: 14.05.2021).
- [6] *GeoGebra*. 2021. URL: <https://www.geogebra.org/> (Einsichtnahme: 14.05.2021).
- [7] *Pandas*. 2021. URL: <https://pandas.pydata.org/> (Einsichtnahme: 14.05.2021).
- [8] *NumPy*. 2021. URL: <https://numpy.org/> (Einsichtnahme: 14.05.2021).
- [9] *Matplotlib*. 2021. URL: <https://matplotlib.org/> (Einsichtnahme: 14.05.2021).
- [10] Keras, Hrsg. *About Keras*. 2021. URL: <https://keras.io/about/> (Einsichtnahme: 13.05.2021).
- [11] Messier, J./ Kalaska, J. F. „Comparison of variability of initial kinematics and endpoints of reaching movements“. eng. In: *Experimental brain research* 125.2 (1999). Comparative Study Journal Article Comparative Study Journal Article, S. 139–152. eprint: 10204767.
- [12] Flash, T./ Hogan, N. „The coordination of arm movements: an experimentally confirmed mathematical model“. In: *The Journal of Neuroscience* 5.7 (1985), S. 1688–1703.
- [13] Campos, F./ Calado, J. „Approaches to human arm movement control—A review“. In: *Annual Reviews in Control* 33.1 (2009). PII: S1367578809000091, S. 69–77.

- [14] Admiraal, M. A./ Kusters, M. J. M. A. M./ Gielen, S. C. A. M. „Modeling kinematics and dynamics of human arm movements“. eng. In: *Motor control* 8.3 (2004). Journal Article, S. 312–338. eprint: 15322310.
- [15] Fligge, N./ McIntyre, J./ van der Smagt, P. „Minimum jerk for human catching movements in 3D“. In: *2012 4th IEEE RAS and EMBS International Conference on Biomedical Robotics and Biomechatronics*. 2012 4th IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob 2012) (Rome, Italy, ). Hrsg. von Author, I. C. IEEE, 6/24/2012 - 6/27/2012, S. 581–586.
- [16] Smith, C./ Jensfelt, P. „A predictor for operator input for time-delayed teleoperation“. In: *Mechatronics* 20.7 (2010). PII: S0957415810000528, S. 778–786.
- [17] Fitts, P. M. „The information capacity of the human motor system in controlling the amplitude of movement“. In: *Journal of Experimental Psychology* 47.6 (1954), S. 381–391.
- [18] „Two-Third Power Law“. In: *Encyclopedia of Neuroscience*. Hrsg. von Binder, M. D./ Hirokawa, N./ Windhorst, U. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, S. 4146.
- [19] Zhao, J. u. a. „Human arm motion prediction in human-robot interaction based on a modified minimum jerk model“. In: *Advanced Robotics* 35.3-4 (2021), S. 205–218.
- [20] Ding, H. u. a. „Human arm motion modeling and long-term prediction for safe and efficient Human-Robot-Interaction“. In: *2011 IEEE International Conference on Robotics and Automation (ICRA)*. 2011 IEEE International Conference on Robotics and Automation (ICRA) (Shanghai, China, ). IEEE / Institute of Electrical and Electronics Engineers Incorporated, 2011, S. 5875–5880.
- [21] Grubbs, F. E. „Sample Criteria for Testing Outlying Observations“. In: *The Annals of Mathematical Statistics* 21.1 (1950), S. 27–58.
- [22] Edvarsson, A. *Online Predictions of Human Motion*. 2017.
- [23] Liu, H./ Wang, L. „Human motion prediction for human-robot collaboration“. In: *Journal of Manufacturing Systems* 44 (2017). PII: S0278612517300481, S. 287–294.

- [24] Reddy, K. K./ Liu, J./ Shah, M. „Incremental action recognition using feature-tree“. In: *Computer Vision, 2009 IEEE 12th International Conference on*. 2009 IEEE 12th International Conference on Computer Vision (ICCV) (Kyoto, ). IEEE, 2009, S. 1010–1017.
- [25] Feng, K.-p./ Yuan, F. „Static hand gesture recognition based on HOG characters and support vector machines“. In: *2013 2nd International Symposium on Instrumentation and Measurement, Sensor Network and Automation (IMSNA)*. 2013 2nd International Symposium on Instrumentation & Measurement, Sensor Network and Automation (IMSNA) (Toronto, ON, Canada). IEEE, 122013, S. 936–938.
- [26] Wang, W. u. a. „Bionic control of exoskeleton robot based on motion intention for rehabilitation training“. In: *Advanced Robotics* 33.12 (2019), S. 590–601.
- [27] Bütepage, J. u. a. *Deep representation learning for human motion prediction and classification*. This paper is published at the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017. 24.02.2017.
- [28] Maeda, Y./ Hara, T./ Arai, T. „Human-robot cooperative manipulation with motion estimation“. In: *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No.01CH37180)*. RSJ/IEEE International Conference on Intelligent Robots and Systems (Maui, HI, USA). IEEE, 2001, S. 2240–2245.
- [29] Kaiser, J. „Dealing with Missing Values in Data“. In: *Journal of Systems Integration* (2014), S. 42–51.
- [30] Arce, G. R. *Nonlinear Signal Processing. A Statistical Approach*. eng. Hoboken, NJ: Wiley, 2005. 1 online resource.
- [31] Gröll, L. „Einführung in Numerik“. (Unveröffentlichtes Skript). 2019.
- [32] Lange, S./ Bender, R. „Median oder Mittelwert?“ In: *Deutsche Medizinische Wochenschrift* 126.S T (2001).
- [33] Cao, Z. u. a. „OpenPose: Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields“. eng. In: *IEEE transactions on pattern analysis and machine intelligence* 43.1 (2021). Journal Article, S. 172–186. eprint: 31331883.
- [34] Elgendi, M. u. a. „Arm movement speed assessment via a Kinect camera: a preliminary study in healthy subjects“. eng. In: *Biomedical engineering online* 13 (2014). Journal Article Research Support, Non-U.S. Gov’t, S. 88. eprint: 24968711.

# Appendices

## **A. Weitere Abbildungen**

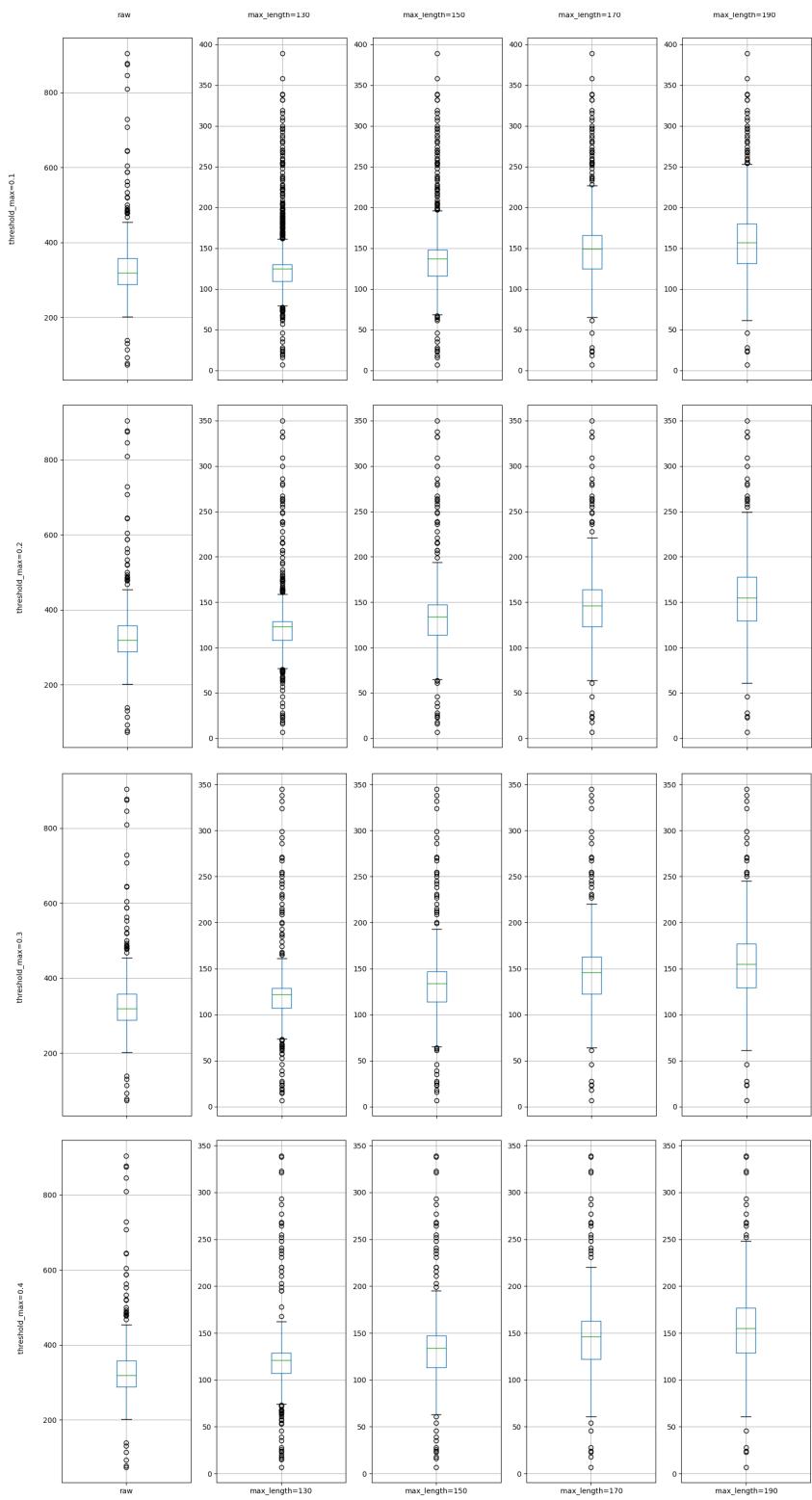


Abbildung A.1.: Boxplots für die Länge der Datensätze nach ihrer Beschneidung geordnet nach den verschiedenen Parametern

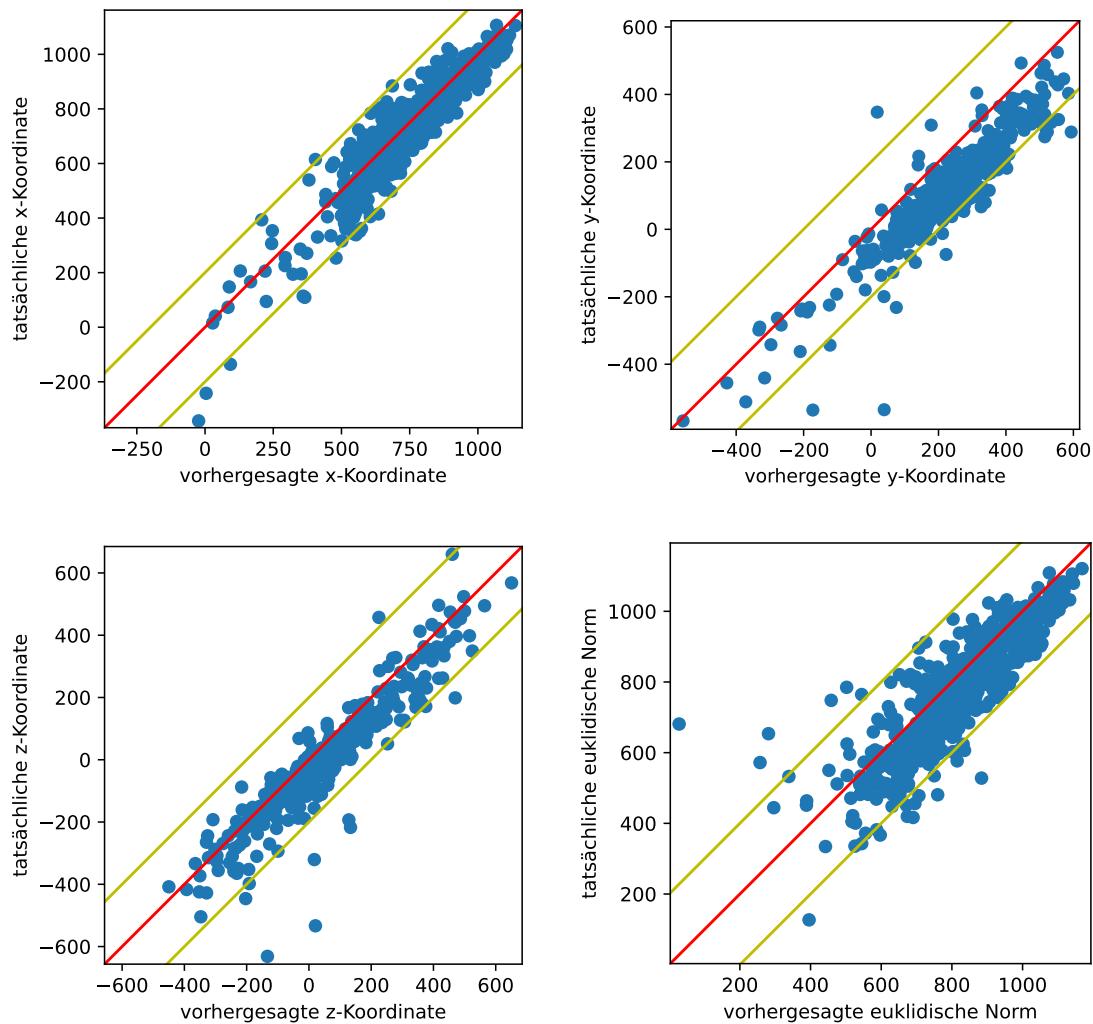


Abbildung A.2.: Visualisierung des Lernerfolgs anhand der Trainingsdaten

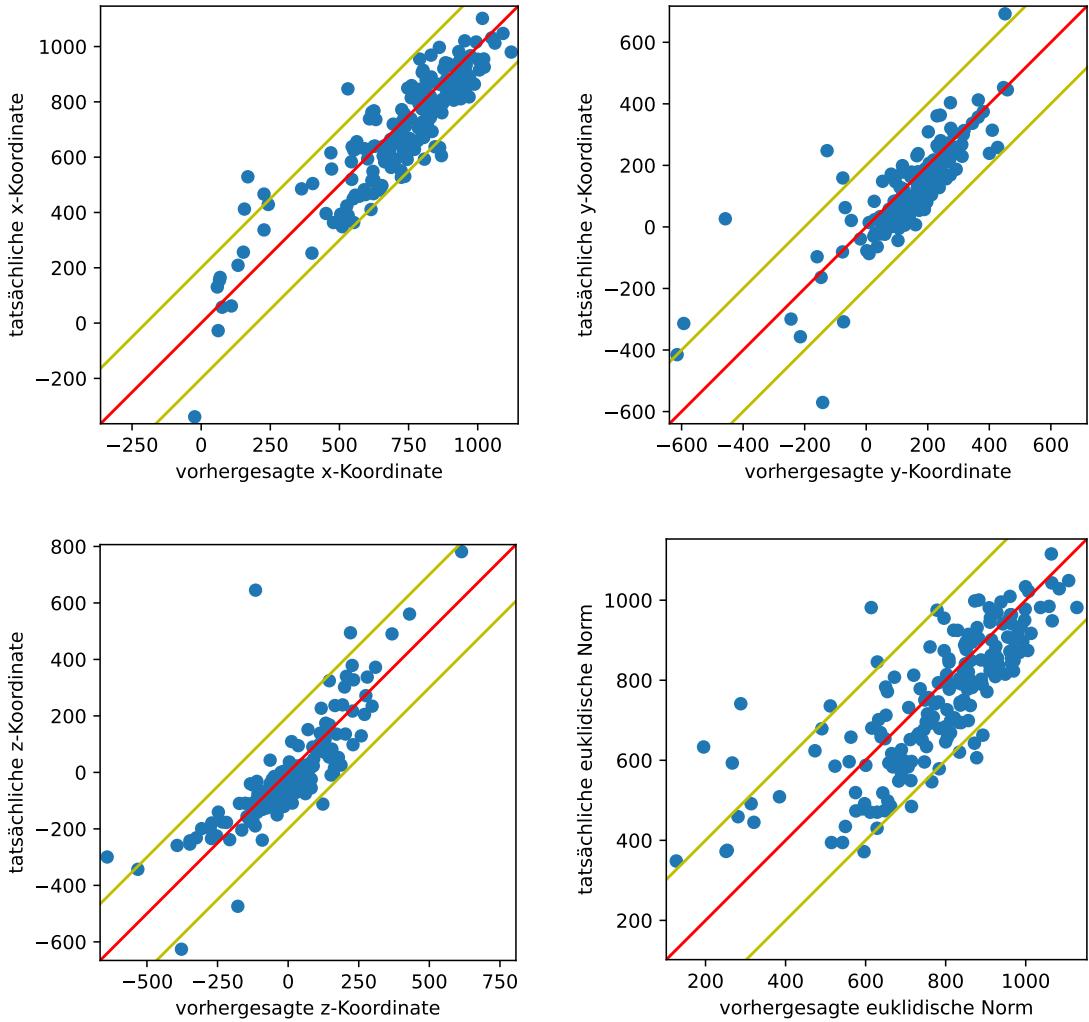


Abbildung A.3.: Visualisierung des Lernerfolgs anhand der Testdaten