



OSTBAYERISCHE
TECHNISCHE HOCHSCHULE
REGENSBURG



MASTERARBEIT

Tamara Szecsey

Entwicklung eines magneto-aktiven Schuhwerks für den humanoiden NAO Roboter zur Verbesserung der Gangstabilität

Fakultät: Elektro- und Informationstechnik
Studiengang: Master Electrical- and Microsystem Engineering
Abgabefrist: 31. März 2021
Betreuung: Prof. Dr. Gareth Monkman
Zweitbegutachtung: Dr. Dirk Sindersberger

Erklärung

1. Mir ist bekannt, dass dieses Exemplar der Masterarbeit als Prüfungsleistung in das Eigentum der Ostbayerischen Technischen Hochschule Regensburg übergeht.
2. Ich erkläre hiermit, dass ich diese Masterarbeit selbstständig verfasst, noch nicht anderweitig für Prüfungszwecke vorgelegt, keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie wörtliche und sinngemäße Zitate als solche gekennzeichnet habe.

Ort, Datum und Unterschrift

Vorgelegt durch: Tamara Szecsey
Matrikelnummer: 3140789
Studiengang: Master Electrical- and Microsystem Engineering
Bearbeitungszeitraum: 1. Juni 2020 – 31. März 2021
Betreuung: Prof. Dr. Gareth Monkman
Zweitbegutachtung: Dr. Dirk Sindersberger

Inhaltsverzeichnis

1 Einleitung	1
2 Theoretischer Hintergrund	2
2.1 Aufbau des NAO	2
2.2 Verwendete Software	6
2.3 Magneto-aktive Polymere	8
2.4 Carbonyleisenpulver	8
3 Versuchsaufbau	10
3.1 Schuhkonstruktion	10
3.2 Herstellung des MAP	11
3.3 Laufstegkonstruktion	12
4 Versuchsdurchführung	14
5 Auswertung und Interpretation	17
6 Fazit und Ausblick	26
7 Anhang	27

1 Einleitung

Kapitel 2.3 erklärt die Definition und Eigenschaften von Magneto-aktiven Polymeren (MAP), welche als Sohle für den Nao Roboter eingesetzt wurden. Diese Erklärungen basieren auf einem Buch von Pelteret und Steinmann [1], welches ich für tiefergehende Lektüren empfehle.

2 Theoretischer Hintergrund

2.1 Aufbau des NAO

NAO ist ein 574 mm großer, humanoider Roboter (siehe Abb. 1) ursprünglich entwickelt von dem französischen Unternehmen Aldeberan Robotics, welche 2015 von Softbank Group aufgekauft [3] und in Softbank Robotics umbenannt wurde. Während NAO's große Schwester Pepper mit ihren 1,20 m mit einem Tablet und Rollen statt Beinen ausgestattet ist [4], gibt es NAO in verschiedenen Ausführungen, sowohl ab der Hüfte aufwärts als auch mit Beinen. Es handelt sich hier um Roboter, die unter anderem Kindern und Jugendlichen die Robotik näher bringen sollen und der Vorführung von Mensch-Roboter Interaktionen dienen. NAO bietet außerdem die Gelegenheit zweibeinige Robotersysteme zu studieren und ist bereits in psychologischen Studien verwendet worden [5].



Abb. 1. NAO V6 [2]

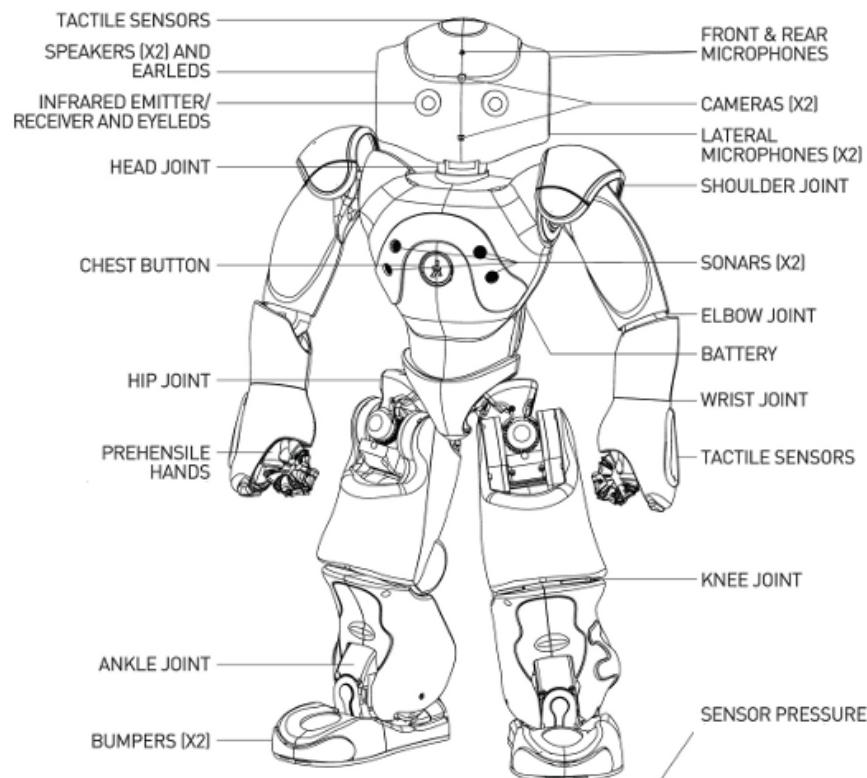


Abb. 2. Sensorenüberblick des NAO-H25 Version 6 [6, in /H25]

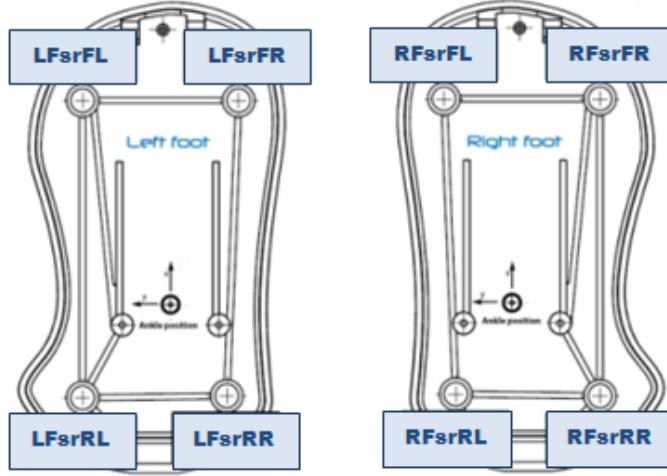


Abb. 3. Drucksensoren und deren Bezeichnung in den Füßen von NAO [2, in /Technical overview/FSRs]

Das hier verwendete Modell ist NAO-H25 Version 6, dessen Sensoren in Abb. 2 zu sehen sind. Im Unterschied zu anderen Ausführungen besitzt NAO-H25 Drucksensoren an Händen und Fußsohlen. Er gehört zu den kommerziellen Robotern deren Gelenke positionsbasierten sind [7], hat 25 Freiheitsgrade und wiegt 5,4 kg. Über die an der Brust angebrachten Sonar Sensoren, die Kameras oberhalb und unterhalb der Augen LEDs, die vorder- und rückseitigen Mikrofone, den Stoßfängern an den Füßen sowie den KontaktSENSOREN an Händen und Kopf kann der NAO mit seiner Umwelt vielseitig interagieren. Jedes Gelenk ist mit Sensoren für die Winkelmessung, den Stromverbrauch und die Temperaturmessung ausgerüstet. In seiner Brust befindet sich außerdem ein Gyroskop. Auf die in dieser Arbeit verwendeten Messausgaben wird im Folgenden genauer eingegangen.

Druckempfindlicher Widerstand

An den Fußsohlen des NAO befinden sich pro Fuß vier sogenannte *Force Sensitive Resistors* (FSR), zu sehen in Abb. 3. Diese ändern ihren Widerstand sobald Druck ausgeübt wird und messen im Bereich von 0 bis 25 N.

Die Ausgaben der Sensoren sind im Dateiverzeichnis des NAO hinterlegt und können jederzeit ausgelesen werden. Dies geschieht hier über dasselbe Pythonprogramm, welches ebenfalls die Bewegung steuert, näheres dazu in Kap. 2.2 und im Anhang 7. Des weiteren können der berechnete zweidimensionale Massenschwerpunkt und das Gesamtgewicht ausgegeben werden. Diese Werte sind allerdings unzuverlässig, sobald wenig oder kein Gewicht auf den Sensoren lastet. In [8] haben Shayan et al. die eingebauten FSR mit barometrischen Drucksensoren verglichen und sind zu dem Schluss gekommen, dass die Sensoren, welche im NAO verbaut wurden, nicht sehr aussagekräftig sind. Deshalb werden hier die Ausgaben kritisch betrachtet, sowie die Balance des Roboters zusätzlich mit dem Gyroskop erfasst.

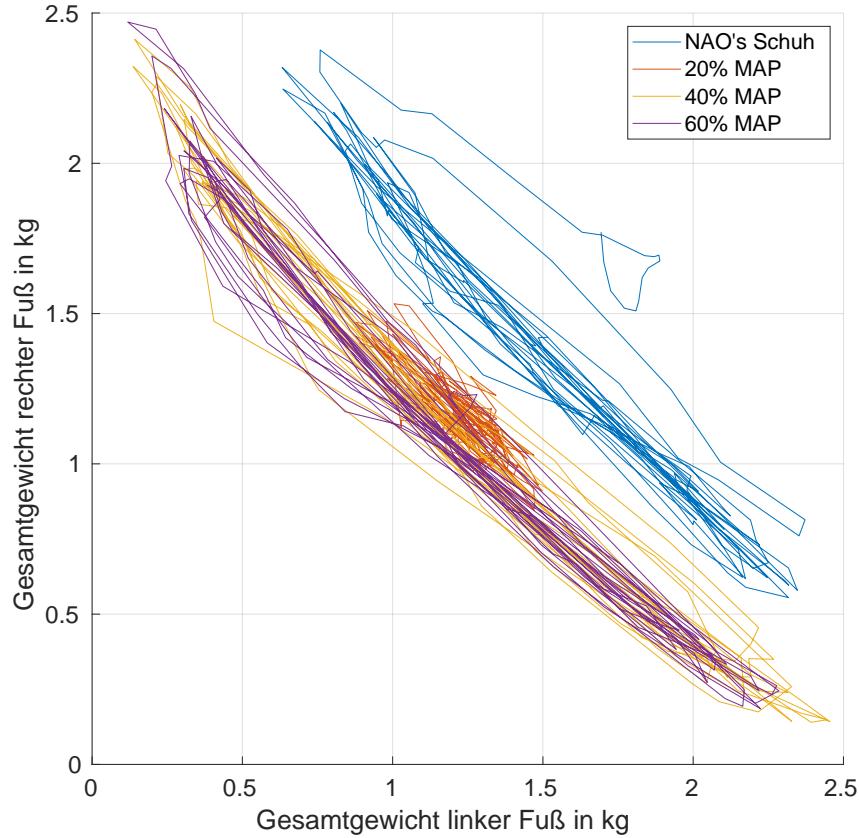


Abb. 4. Berechnetes Gesamtgewicht der Messungen für die normalen Sohlen von NAO in blau und dem Silikon versetzt mit Eisenpartikeln in 20%, 40% und 60% Anteilen. Das Gewicht wird in kg ausgegeben.

Eine weitere Einschränkung der Messungen ist die Kraftverteilung auf die FSRs. Denn in dem ursprünglichen unteren Teil des Schuhs liegt der obere Teil ausschließlich auf den Sensoren und auf den Verbindungszyllindern für die Schrauben auf. Dies ist für die Einlegesohlen aus Silikon bzw. MAP nicht der Fall, zu sehen in den Kapiteln 3.1 und folgende. Dies führt dazu, dass das Gewicht nicht mehr akkurat aufgenommen wird, siehe Abb. 4. Hier wird deutlich, dass sich die Messung des Gewichtes etwa um 0,5 kg von den NAO Schuhsohlen abweicht.

Aktoren und Sensoren der Beinen

Neben dem für diese Arbeit uninteressanten Bumper Sensoren und den bereits beschriebenen Drucksensoren hat NAO eine Ausgabe für jeden eingebauten Aktor mit den Werten:

- .../Position/Actuator/Value (Pos/Act)
- .../Position/Sensor/Value (Pos/Sens)
- .../ElectricCurrent/Sensor/Value (Current)

- .../Temperature/Sensor/Value
- .../Hardness/Actuator/Value
- .../Temperature/Sensor/Status

Hierbei unterscheiden sich die Pfade bei „...“ je nach Aktor. Die Bezeichnungen in Klammern dahinter dienen der abkürzenden Benennung für spätere Kapitel. Der Effekt der Sohlen auf die Aktoren wurde mit AnkleRoll und AnklePitch zu sehen in Abb. 5 aufgenommen.

Die Temperaturausgaben sowie die Starrheit der Motoren liefern keine verwertbaren Messausgaben, da sich beide während einer Messung nicht oder kaum ändern. Pos/Act und Pos/Sens geben ähnliche Werte in rad aus, da ersteres die Ausgabe des Programms vorgibt und zweiteres den tatsächlich gemessenen Wert an dem Gelenk ausgibt. Der Current

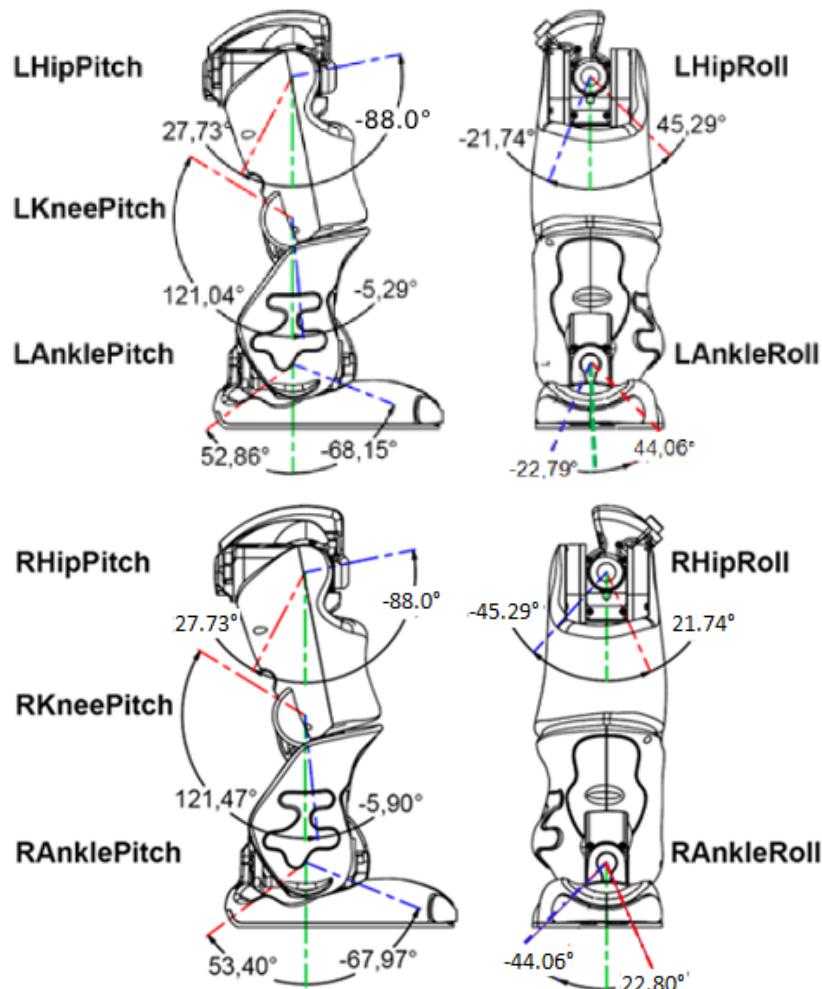


Abb. 5. Das obere Bild zeigt Vorder- und Seitenansicht der Positionen und möglichen Winkel des linken Beins. Die untere Abbildung veranschaulicht dieselben Parameter für das rechte Bein. [2, in /kinematics-data/joints]

Wert wird in Ampère gemessen und gibt an, wie viel Strom für das Erreichen der entsprechenden Aktorposition und Starrheit aufgewendet werden muss. Dies bedeutet, dass dieser Wert unter anderem eine Aussage über den Zustand des Aktors geben kann.

Bei den Probegängen des NAO stellte sich heraus, dass er in einer Kurve nach links läuft bei einem Befehl, der ihn hätte gerade aus laufen lassen sollen. Softbank Robotics betonte, dass es nicht möglich ist, dass NAO komplett gerade läuft. Grund dafür ist zum einen, dass die Motoren nicht immer komplett identisch hergestellt sind. Zum anderen ist der Gehbefehl, welcher hier ausgeführt wird, ohne Rückkopplungsschleife für Einwirkungen der Umgebung ausgestattet. Näheres zur Software wird in Kapitel 2.2 beschrieben.

Gyroskop

Der NAO Roboter verfügt über eine Inertialeinheit, welche sich zusammensetzt aus den 3 Achsen des Gyrometers, im einer Achsengeschwindigkeit von bis zu $500^\circ/\text{s}$, sowie den 3 Achsen des Beschleunigungssensors, mit einer Beschleunigung bis zu 2 g [2, /Technical overview/Inertial unit]. Die Z-Achse des Gyroskops, welche die Höhenorientierung des Roboters anzeigen würde, ist allerdings noch nicht verfügbar. Außerdem wird durch diese beiden Sensoren der Winkel des Torsos bestimmt.

2.2 Verwendete Software

Die Rahmenumgebung NAOqi

NAOqi wird die Hauptsoftware genannt, welche auf diesem Roboter sowie auf Pepper läuft, das Betriebssystem NAOqi OS basiert auf Linux. Der NAOqi Rahmen ist eine plattformübergreifende und sprachenübergreifende Umgebung, mithilfe derer Anwendungen für den NAO erstellt werden können. Sie kann über Windows, MacOS und Linux verwendet werden und es werden die Sprachen Python und C++ unterstützt, wobei letzteres direkt auf dem NAO kompiliert werden kann, während C++ komplizierter benutzt wird, aber mehr Eingriffe erlaubt. [9, /Former NAOqi Framework/Key concepts]

Außerdem gibt es eine Desktop Anwendung namens Choregraphe in der Dialoge und Verhaltensmuster erstellt werden können ohne Code schreiben zu müssen. Hierüber ist der Akkustand und aktuelle Position in Form eines virtuellen Bildes des tatsächlichen NAO einsehbar, sowie der autonome Zustand ein- und ausschaltbar. [9, /Choregraphe Suite/What is Choregraphe]

Für die in dieser Masterarbeit benötigten Anwendungen war Python am besten geeignet. Erforderlich waren eine sich wiederholende Schrittabfolge sowie die Aufzeichnung diverser bereits im NAO verbauten Sensoren. Text verarbeitende Funktionen sind in dieser Sprache leicht zu erhalten und anzupassen. Es wurde sich für den `moveTo()` Befehl als Fortbewegungsmethode entschieden, denn dieser wird durch das Objekt `post` zu einem sogenannten *non-blocking call*. Dies ermöglicht das Aufrufen von weiteren Befehlen zeitgleich zur Bewegung. Der Nachteil ist, dass der Gang dadurch unbeaufsichtigt vorstatten geht, d.h. NAO

kann seine Schritte nicht seiner Umgebung anpassen. Die aufgezeichneten Daten werden in eine csv Datei übertragen und mit Matlab ausgewertet. Im Anhang ist der gesamte Programmcode 1 abgebildet, welcher auf dem NAO ausgeführt wird.

Konstruktionssoftware

Das CAD-Programm Inventor von Autodesk ist für 3D Konstruktionen ausgelegt und bietet einige nützliche Simulationserweiterungen, welche unter anderem einen Shape Generator enthält [10]. Dieser kann Flächen minimieren während die Stabilität erhalten bleibt, sodass ein minimaler Anteil an Material verwendet wird [11].

Für die Herstellung der Prototypen und Gussformen wurde das FDM (fused deposition modeling) 3D Druckverfahren mit PLA oder PETG Filamenten von filamentworld [12] verwendet und für die Vorbereitung auf den Druck der Slicer Cura von Ultimaker [13].

Datenauswertung mit Matlab

Für die Auswertung wurden neben der gewöhnlichen `plot` Funktion von Matlab [14] auch die *Statistics and Machine Learning Toolbox* mit deskriptiver Statistik und Visualisierung verwendet [15]. Eine Messreihe ergibt sich aus 40 Messungswiederholungen, in denen NAO etwa einen Meter zurücklegt. Jede Auswertung enthält das arithmetische Mittel eines Sensors von allen Messungen der Messreihe. Diese Mittel wurden mit dem Befehl `mean()` berechnet und mit Werten aus weiteren Messreihen, mit anderem Schuh oder anderen Untergrundvoraussetzungen aber gleicher Schrittänge und Sensorenmessung verglichen. Dieser Vergleich wurde mit den Funktionen `hist` und `scatterhist` der genannten Toolbox angestellt. Ersteres erstellt ein Histogramm und ist damit optimal für eine Häufigkeitsverteilung. Zweiteres erstellt einen aus zwei Vektoren resultierenden Scatterplot sowie zwei Histogramme, welche die Häufigkeitsverteilung der jeweiligen Vektoren darstellen. Dies ermöglicht eine umfangreiche Verteilungsanalyse für Sensorausgaben, die sich aus zwei Achsen zusammensetzen, wie z.B. der 2 dimensionalen Gyroskop Ausgabe.

Das arithmetische Mittel berechnet Matlab durch [16, S.50]

$$\bar{x} = \frac{1}{n}(x_1 + x_2 + \dots + x_n) = \frac{1}{n} \sum_{i=1}^n x_i \quad (1)$$

und wurde auf eine Messreihe angewendet. Die Darstellung der Verteilungen per Histogramm wurde ergänzt durch ein gleitendes Histogramm, errechnet durch den Kern-Dichteschätzer [16, S.93]

$$\hat{f}(x) = \frac{1}{nh} \sum_{i=1}^n K\left(\frac{x - x_i}{h}\right), \quad x \in \mathbb{R} \quad (2)$$

mit dem Gauss-Kern [16, S.93],

$$K(u) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}u^2\right), \quad u \in \mathbb{R} \quad (3)$$

welcher in Matlab mit 'Kernel' und dann standardmäßig mit 'normal' in der `fitdist()` Funktion aufgerufen wird [17] [18]. Das Scatterhistogramm ermöglicht dieselbe Verwendung dieser Distribution.

2.3 Magneto-aktive Polymere

Der Begriff Magneto-Aktive Polymere (MAP) gehört zur Gruppe der intelligenten, auf magnetische Felder ansprechende Materialien, welche typischerweise Kombinationen aus einer weichen Polymermatrix und darin eingebetteten, magnetischen Partikeln sind. Diese Partikel werden während dem Vernetzungsprozess des Polymers in dieses eingebettet.

Die wesentlichen Verhaltensweisen, die MAP in der heutigen Zeit attraktiv für seine Verwendung macht, wurde bereits in den 80er Jahren von Rigbi und Jilken [19] sowie Rigbi und Mark [20] beschrieben. Ein Jahrzehnt später wurde eine genauere Analyse zum ersten Mal von Ginder und Jolly et al. [21] veröffentlicht. Diese aus mehreren Komponenten bestehenden Materialien stechen durch zwei Schlüsseleigenschaften heraus. Zum einen das magnetostriktive Verhalten, bei dem es sich um das Phänomen der Verformung eines Materials handelt, welches durch ein Magnetfeld hervorgerufen wird [22]. Zum anderen die veränderbaren Materialeigenschaften wie Elastizität und Dämpfungsfaktor, welche hauptsächlich mit der Mikrostruktur des Grundlagenmaterials zusammenhängen. [23] [24]

Außerdem ist entscheidend, wie die magnetischen Partikel in das Polymer eingebettet werden. Je nachdem ob während des Vernetzungsprozesses ein Magnetfeld wirkt, können sich die Partikel kettenförmig ausrichten und dadurch dem MAP eine anisotropisches Verhalten zuführen. Isotropisches MAP hingegen enthält keine gerichteten Partikel. Diese verschiedenen Ausrichtungsarten und Herstellungsarten können sowohl die Steifigkeit verändern, als auch bestimmen, ob das MAP in einem Magnetfeld ausgedehnt oder zusammengedrückt wird.

Weiche, mit einem Feld manipulierbare, Polymere haben diverse Anwendungsbereiche in akademischen und industriellen Bereichen. Angefangen von anpassungsfähiger Vibrationsabsorption in der Luftfahrt und Automobilindustrie durch das Einsetzen durch Scherung [25] [26], Windung [27] und Kompression bzw. Elongation [28] und Vibrationsisolatoren [29] sowie Sensoren [29] [22], Ventile und Aktoren [30] [31] und anpassungsfähige, sandwichartige Strukturen [32] [33] [34] bis hin zur Anwendung in der Bionik wie zum Beispiel durch Mikro- und Nanoroboter und Schwimmroboter [35] [36] [37] [38], Schlauchradpumpen [39] und Erschütterungsdämpfer [40].

Des Weiteren wird die Verhärtung bei Anlegen eines Magnetfeldes für Greifer genutzt [41], ebenso wird der 3D Druck von magneto-aktiven Polymeren [42] bereits erforscht.

2.4 Carbonyleisenpulver

Bei den in dieser Arbeit verwendeten magnetischen Partikel für die Herstellung des MAP handelt es sich um das von BASF vor über 85 Jahren entdeckte Carbonyleisenpulver (engl. *carbonyl iron powder*), CIP) [43]. Es wird durch thermische Zersetzung von Eisenpentacarbonyl hergestellt und besteht aus Teilchen in einer Größenordnung von 5 bis 1 μm .

Die magnetischen Eigenschaften dieses Materials lassen sich durch die Suszeptibilität [44, Gleichung (2)]

$$\chi = \frac{3\chi_p}{3 + \chi_p} \varphi \quad (4)$$

mit χ_p als Suszeptibilität der einzelnen Teilchen und einem Partikel Volumen φ welches sehr viel kleiner als 1 ist, beschreiben. CIP zählt zu den ferromagnetisch weichen Materialien, was sich durch Untersuchungen der Hysteresekurve veranschaulichen lässt, wie zum Beispiel von Liu u.a. [45] zu sehen in Abb. 6

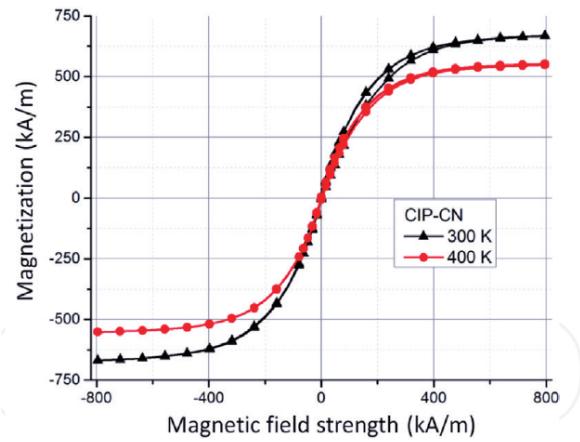


Abb. 6. Hysteresekurve von CIP entnommen aus [45, Abb.4]

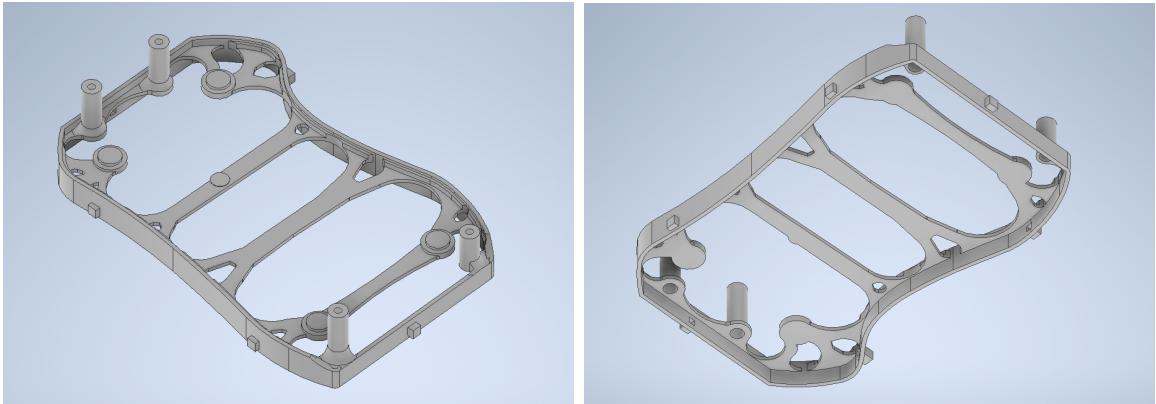


Abb. 7. Linker Schuh in Autodesk Inventor, *Links von Oben, Rechts von Unten*.

3 Versuchsaufbau

In diesem Kapitel wird auf alle selbstkonstruierten Komponenten eingegangen, welche für die Messungen benötigt wurden. Zum einen gibt es eine Schuhkonstruktion, die es erlaubt, die in einer Gussform angefertigten MAP Sohlen einzuhängen, zum anderen wurde ein Laufsteg mit Rampenfunktion entworfen, welcher es erlaubt, Magnete unter die Lauffläche zu montieren.

3.1 Schuhkonstruktion

Die Hülle eines Fußes von NAO besteht aus einem zweiteiligen Oberteil, welches das Fußgelenk abschließt und einem unteren Teil, welcher mit 4 Schrauben angebracht wird. Ohne Schraubenbefestigung liegt der Fuß nicht fest an. Die in Kapitel 2.1 beschriebenen vier Drucksensoren liegen dabei in dem unteren Teil auf Erhöhungen auf. Um die Auswirkungen anderer Sohlen für NAO messen zu können, muss der untere Teil des Fußes ausgetauscht werden. Dieser „Schuh“ welcher die ursprüngliche Fußsohle ersetzt, enthält wiederum einen Steckplatz für die MAP Sohlen in verschiedenen Stärkegraden.

Zur Formabnahme wurden sowohl die obere als auch die untere Hülle des Fußes von NAO 3D gescannt und in Polygonnetzformat (engl. *mesh*) bei Autodesk Inventor eingespeist. Somit konnten alle Maße direkt abgenommen und eingehalten werden.

In Abb. 7 sind links die vier flachen Zylinder zu sehen, auf denen die Drucksensoren aufliegen. Die vier Zylinder an den Seiten sind die Führung der Schrauben, welche an das obere Teil des Fußes von NAO geschraubt werden. Die Außenform umschließt den oberen Teil während eine zweite Erhöhung, welche an den Innenseiten verläuft, auf dem Rand des oberen Teils aufsitzt, sodass die Passform fest ineinander greift. Da das gesamte Gewicht des NAO auf den 4 Drucksensoren lastet, kann Druckmaterial für die restliche Gesamtfläche bis auf stabilitätserhaltende Streben eingespart werden. Diese Form wurde mit dem Shape Generator von Autodesk Inventor generiert, sodass sie längs bis zu einem 45° Winkel ohne zu brechen gebogen werden kann. Die instabilsten Stellen sind die Zylinder der Schraubvorrichtung, welche durch Fillets verstärkt wurden. Diese Instabilität ist auf den

schichtweise Druckvorgang durch das FDM Verfahren geschuldet und kann durch kleinere Schichthöhen ausgeglichen werden.

Die Unterseite, zu sehen in Abb. 7 rechts, ist ein Hohlraum für die MAP Sohle zusammen mit den viereckigen Steckeinlässen für die Halterung. Die Seiten des Schuhs sind so hoch, dass das MAP etwa 1 mm herausragt. Andernfalls würde NAO auf der Schuhkante laufen und nicht auf dem MAP.

3.2 Herstellung des MAP

Der Herstellungsprozess allein dauert nicht mehr als eine halbe Stunde. Zunächst muss das Verhältnis für den Anteil des CIPs bestimmt werden. Die Masse ergibt sich aus

$$m_{CIP} = \frac{Ratio_{CIP}[\%]}{100\%} \cdot m_{ges}, \quad (5)$$

Die beiden additiven Komponenten A und B, welche auch als Basis und Katalysator bezeichnet werden, sind im Verhältnis 1:1 zu mischen. Das Volumen in ml ergibt sich aus:

$$V_A = \frac{m_A}{\rho_A}, \quad V_B = \frac{m_B}{\rho_B} \quad (6)$$

mit $\rho_A = 1.071$ sowie $\rho_B = 1.046$ sowie

$$m_A = \left(1 - \frac{Ratio_{CIP}[\%]}{100\%}\right) \cdot \frac{m_{ges}}{\alpha + \beta} \cdot \alpha, \quad m_B = \left(1 - \frac{Ratio_{CIP}[\%]}{100\%}\right) \cdot \frac{m_{ges}}{\alpha + \beta} \cdot \beta, \quad (7)$$

Die Platzhalter α und β stehen für das Mischverhältnis der jeweiligen Komponenten und sind hier Beide gleich 1, sodass die Formeln vereinfacht werden können in:

$$m_A = \left(1 - \frac{Ratio_{CIP}[\%]}{100\%}\right) \cdot \frac{m_{ges}}{2}, \quad m_B = \left(1 - \frac{Ratio_{CIP}[\%]}{100\%}\right) \cdot \frac{m_{ges}}{2}. \quad (8)$$

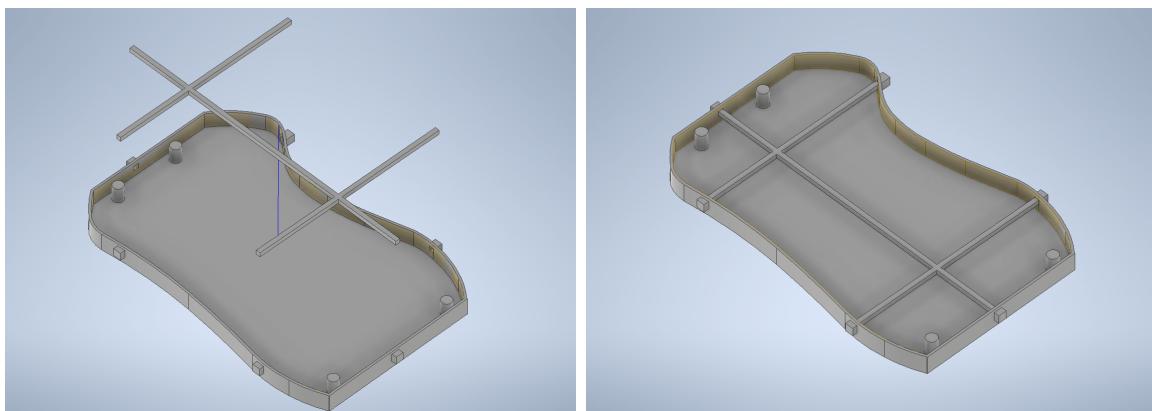


Abb. 8. Gussform der MAP Sohlen. *Links* ist die Innenhalterung herausgenommen. *Rechts* ist sie eingespannt in den sechs Eckhalterungen.

Mit der Laborwaage ABT 120-5DM von Kern wird m_{CIP} in einem Becher abgemessen. Mit den Spritzen lassen sich V_A und V_B auf die ersten Nachkommastelle genau beifügen. Es handelt sich um SF13 2k-Silikon vom Hersteller Silikon Fabrik. Der Becherinhalt wird dann mit einem kleinen Mixstab gemischt, um eine gleichmäßige Verteilung der beiden Komponenten zu erreichen und somit eine optimale Vernetzung zu gewährleisten. Anschließend wird die Probe in einen Exsikkator gestellt, welcher mit einer Vakuumpumpe entgast wird, sodass das Silikon entgast ist. Schließlich kann das bis dahin noch flüssige MAP in die Gussform gegossen werden, nach spätestens einem Tag ist die Sohle dann komplett vernetzt.

Da Silikon selbst sich nur sehr schlecht durch etwaige Klebstoffe nach der Vernetzung verkleben lässt wird hier wie in Abb. 8 zu sehen ist, eine 2 mm dicke Stangenkonstruktion eingehängt, welche bis auf die 6 Enden mit MAP umschlossen wird. Dieses aus PLA gedruckte Konstrukt ist flexibel und kann deshalb durch Verbiegen in die Verankerungen gedrückt werden. Nach der vollständigen Vernetzung kann die Sohle aus der Form entnommen und in den Schuh aus dem vorherigen Kapitel eingesetzt werden.

Die vier Zylinder dienen als Platzhalter um die sechs Ecken in der Halterung des Schuhs für einen besseren Halt festzukleben und dann durch die Löcher des MAPs die Schrauben lockern zu können.

Das Silikon selbst hat eine zu große Haftung, v.a. durch die Fläche des Schuhs. Deshalb wird es vor der Messung mit Speisestärke eingedeckt, was eine Bodenhaftung ähnlich der Plastiksohle des Originalschuhs von NAO zur Folge hat. Dies verhindert außerdem ungewollte Adhäsion.

3.3 Laufstegkonstruktion

Der NAO Roboter ist für den Einsatz auf geraden Bodenflächen im Innenbereich ausgelegt wobei er bei einem Bewegungsablauf ohne Anpassung an die Umwelt wie mit dem Befehl `moveTo()` durch Rutschen nicht immer die gleiche Strecke zurücklegt. Um wiederholbare Messreihen garantieren zu können ist eine Teststrecke von Nöten. Des Weiteren sind verschiedene, flache Untergründe für eine Sohlenentwicklung interessant. Außerdem kann auf das MAP nur Einfluss genommen werden, wenn ein magnetisches Feld angelegt wird. Deshalb wurde ein Laufsteg mit einem Hohlraum angefertigt, um unter der Fläche, auf der NAO läuft, Magneten anzubringen.

Der Laufsteg besteht aus einer $120 \times 66,4$ cm großen Pressholzplatte, die auf der Oberseite mit einem Aluminiumkonstrukt erweitert ist, welches die Einschubplatten von beiden Längsseiten und nach oben hin abschließt, Abb. 10 links. Auf den kurzen Seiten verriegeln jeweils zwei drehbare Keile den Einschub, so dass die Platten eingeschlossen werden, Abb. 9.



Abb. 9. Seitenansicht der Rampe mit einer Breite von 66,4 cm.

Auf der Unterseite sind an den Längsseiten zwei mit T-Nut versehene Aluminiumstangen angebracht, sowie eine zweiteilige Stangenkonstruktion, die eine Winkelverstellung mit Raste erlaubt, zu sehen in Abb. 10 rechts. Die einstellbaren Winkel betragen ca. 5° bis 17° , oder es wird für 0° vollständig eingeklappt.

Man hat NAO bereits schräge Flächen gehen lassen wie in [46]. Dies erfordert einen komplett anderen Gang und wäre über den zeitlichen Rahmen dieser Arbeit hinausgegangen. Die Neodymmagnete, die verwendet wurden, haben eine Haftkraft von ca. 16 kg, eine Maße von $40 \times 40 \times 4$ mm [47] und wurden an die Unterseite der Rampe geschraubt. Die ersten Versuche ergaben schließlich, dass das MAP nur bei einem Abstand ohne Einlageplatten reagierte. Deshalb wurden in den gesamten Messungen ohne diese Platten durchgeführt.

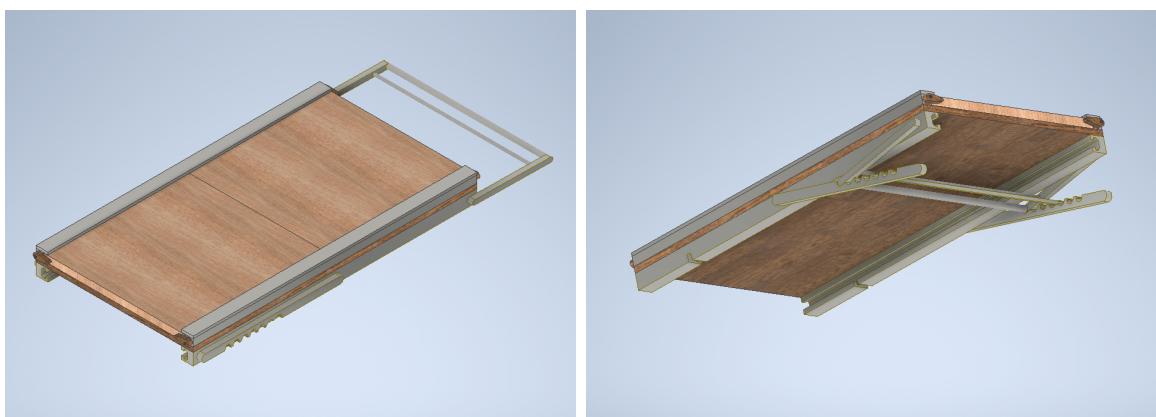


Abb. 10. Laufstegrampenkonstruktion mit zwei austauschbaren Platten und einer Winkelverstellung mit Raste. *Links:* Sicht von schräg oben mit eingeklappter Winkelverstellung. *Rechts:* Sicht von schräg unten mit niedrigster Winkeleinstellung.

4 Versuchsdurchführung



Abb. 11. Links: Der in Abb. 7 in Autodesk Inventor erstellte Schuh aus PETG befestigt an der Unterseite des Fußes von NAO. Rechts: Die aus der Gussform aus Abb. 8 entnommene Sohle mit 20 % MAP Anteil befestigt in dem aus PETG gedruckten Schuh.

Bereits in den vorherigen Kapiteln wurde auf die Durchführung dieses Versuchs eingegangen. Im Folgenden wird über den Ablauf und einige Problematiken gesprochen, welche sich während dem Herstellungs- und Laufprozess herauskristallisierten.

Wie bereits in Kapitel 3.2 beschrieben, wird zunächst das MAP hergestellt und anschließend in die Formen gegossen. Während der Vermengung wurde festgestellt, dass bei 60 % gem. MAP die Durchmischung nicht gewährleistet werden kann, solang der verwendete Becher über die Hälfte voll ist. Deshalb wurden in diesem Fall zwei identische Proben hergestellt, welche erst in der Form zusammengegossen wurden. Alle anderen Proben mussten nach der Durchmischung in zwei separate Becher umverteilt werden, da während der Entgasung das Gemisch an Volumen bis zu einer Druckabnahme von etwa 15 mbar zunimmt. Dies hat zur Folge, dass der Becher nur etwa halb voll sein darf, damit das MAP nicht „überkocht“.

Nach spätestens 24 Stunden ist das Silikon vollständig ausgehärtet und kann aus der Form gelöst werden. Aufgrund der Einhängevorrichtung musste die Gussform hierbei aufgebrochen werden, um die nur 2 mm dicken Stangen nicht abzubrechen oder aus dem MAP zu reißen. Zudem verkeilt sich das Silikon in den Unebenheiten des 3D Drucks, sodass selbst mithilfe von Silikonöl, welches vor dem Gießen in die Form gegeben werden kann, sich das MAP nur schwer lösen lässt.

Um nun die Sohlen an dem Roboter anzubringen, wird zuerst der aus PETG bestehende Schuh angebracht, zu sehen in Abb. 11 links. Im Unterschied zu dem Originalteil ist dieser größtenteils offen und besitzt kein Gegenstück des Magneten unter der zweiten Strebe, welcher bei der Montage hilft. Anschließend wird die MAP Sohle durch Verbiegen und Druck eingehängt und schließt dadurch den Schuh ab, siehe Abb. 11 rechts. Nun ist der Roboter bereit für das Testen des Gangs.

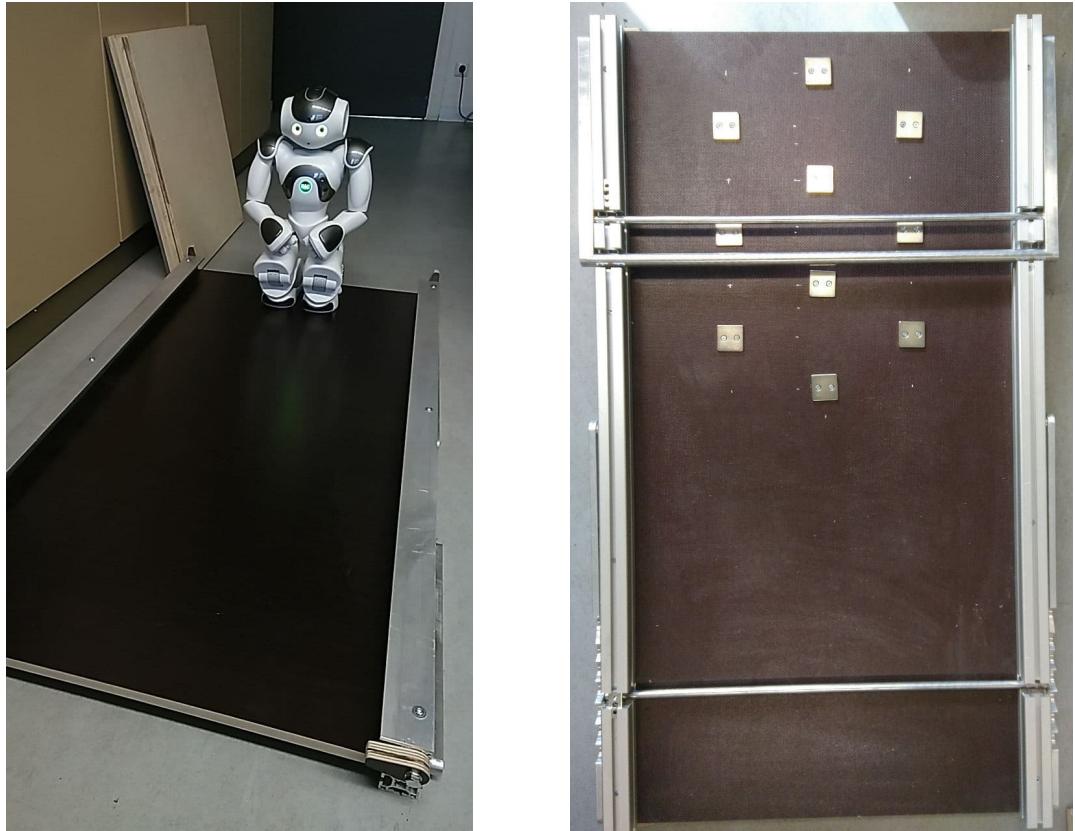


Abb. 12. *Links:* Der NAO Roboter steht im Ruhezustand an der Startposition auf der Rampe ohne Einlageplatten (an der Wand links neben NAO). *Rechts:* Rampenunterseite mit befestigten Neodymmagneten.

Zunächst wurde der Laufsteg ohne Magneten verwendet. Erst nach der erfolgreichen Messung aller Sohlen, d.h. mit höchstens einem Sturz pro Messreihe, wurden die Magneten wie in Abb. 12 rechts zu sehen, angebracht.

Zu Beginn jeder Messung wird NAO in hockender Position aufgestellt, Abb. 12 links. Dies ist seine Ruheposition, in der er gestartet wird und in die er bei dem Herunterfahren zurückkehrt. Der autonome Modus muss in Choregraphe (siehe Kapitel 2.2) bei Start ausgeschaltet werden, andernfalls würde NAO sich sofort aufstellen, sich leicht bewegen und bei Gesichtserkennung mit Spracherkennung beginnen und antworten.

Sowohl über Lan als auch Wlan lässt sich über SSH eine Verbindung zu NAO herstellen. Um das Gleichgewicht besser messen zu können wurde Wlan genutzt. Über die Shell wurde das Programm 1 (siehe Anhang) ausgeführt. Der Roboter steht zuerst auf, wobei seine Beine hierbei in einer schiebenden Bewegung zur Seite gedrückt werden. Dies hat zur Folge, dass bereits hier der Schuh zerstört wird, wenn die Sohlen nicht rutschen können. Anschließend wird die Messung begonnen während NAO den vorgefertigten Gang ausführt.

Für jede Messabfrage benötigt der Roboter eine Verarbeitungszeit, das bedeutet, dass die Dauer eines Abfragendurchlaufs, also der Abfrage aller erwünschten Werte, an die Anzahl der Ausgaben gekoppelt ist. In den im folgenden Kapitel ausgewerteten Abfragen wurden

insgesamt 24 unterschiedliche Sensorenausgaben eingeholt. Damit NAO hierbei etwa einen Meter Strecke zurücklegt, werden 220 Durchläufe dieser Abfragen durchgeführt (zu erkennen in der For-Schleife von Code 1). Schlussendlich nimmt NAO erneut die hockende Position ein und kann wieder an den Anfang des Laufstegs getragen werden.

Anfangs war die Stabilität mit MAP Sohlen nicht gewährleistet. Dies lag zum einen daran, dass das Silikon am Boden durch Adhäsion festklebte. Dadurch wurde entweder die Sohle aus der Verankerung gerissen, NAO verlor das Gleichgewicht oder die Schuhe brachen. Dies wurde durch die bereits genannte Speisestärke verbessert. Ein weiteres Problem waren die Zylinder, welche den Schuh durch Schrauben an der Oberseite befestigen. Diese brachen meist auf Höhe der Querstreben, wodurch NAO sehr instabil wurde. Durch eine Verstärkung dieser Stellen in Autodesk konnte auch diese Schwachstelle eliminiert werden.

Die Schräglagefunktion der Rampe wurde nicht genutzt. Grund hierfür ist eine sehr differenzierte Herangehensweise des Gangs, die erforderlich wäre, damit NAO laufen kann, siehe auch [46].

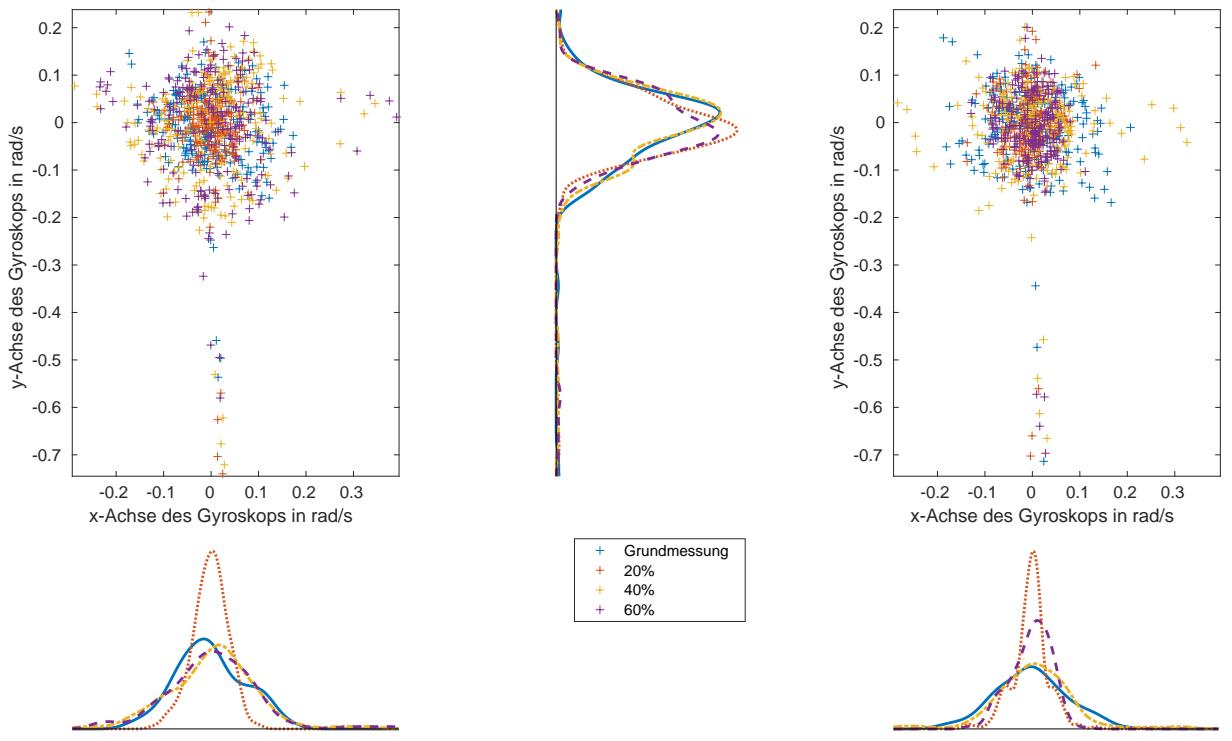


Abb. 13. Ausgabe des Gyroskops, x-Achse auf y-Achse, links ohne Magneten, rechts mit Magneten

5 Auswertung und Interpretation

In diesem Kapitel werden die aufgenommenen Messungen ausgewertet und deren Bedeutung graphisch analysiert. Wie in Kapitel 2.2 erwähnt, werden zweidimensionale Ausgaben, wie die x- und y-Ausgabe des Gyroskops, in einem Scatterhistogramm dargestellt, welches es ermöglicht, die Dichteverteilung einzusehen.

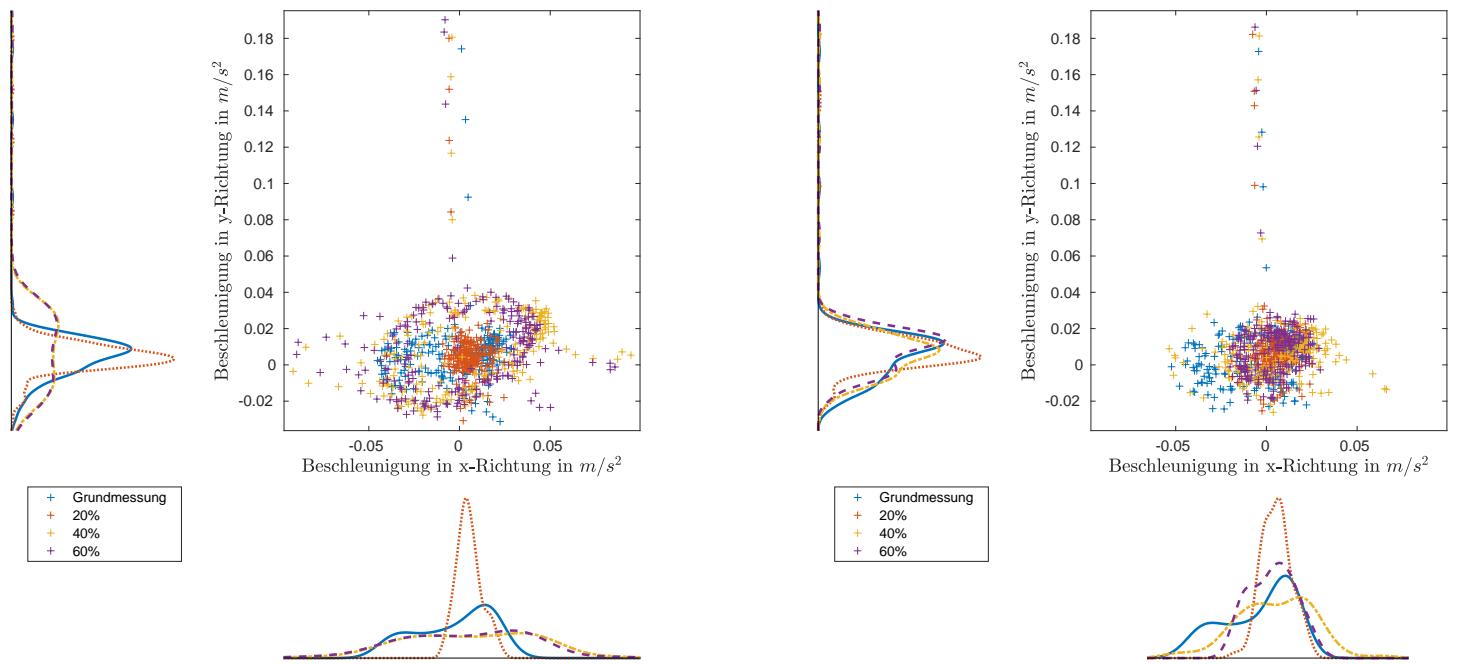


Abb. 14. Ausgabe des Beschleunigungssensors, x-Achse auf y-Achse aufgetragen, links ohne Magneten, rechts mit Magneten

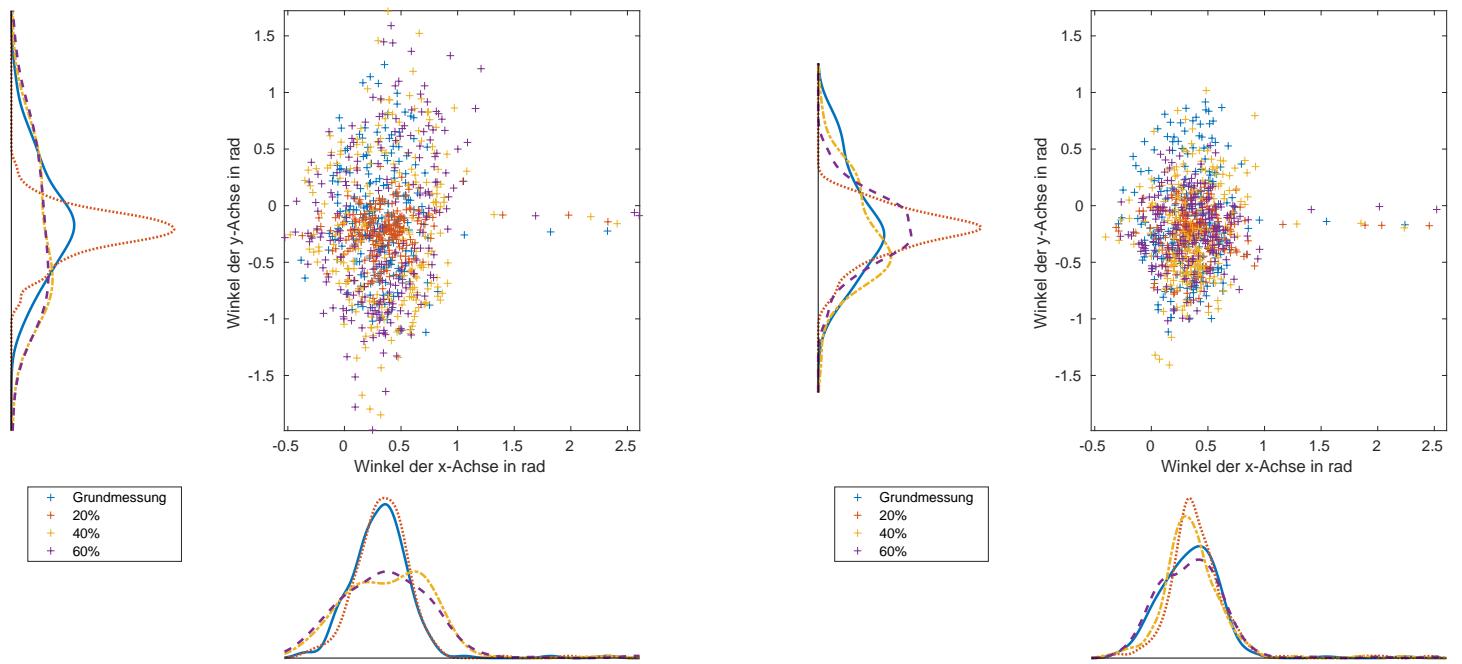


Abb. 15. Echtzeit errechneter Winkel, x-Achse auf y-Achse aufgetragen, links ohne Magneten, rechts mit Magneten

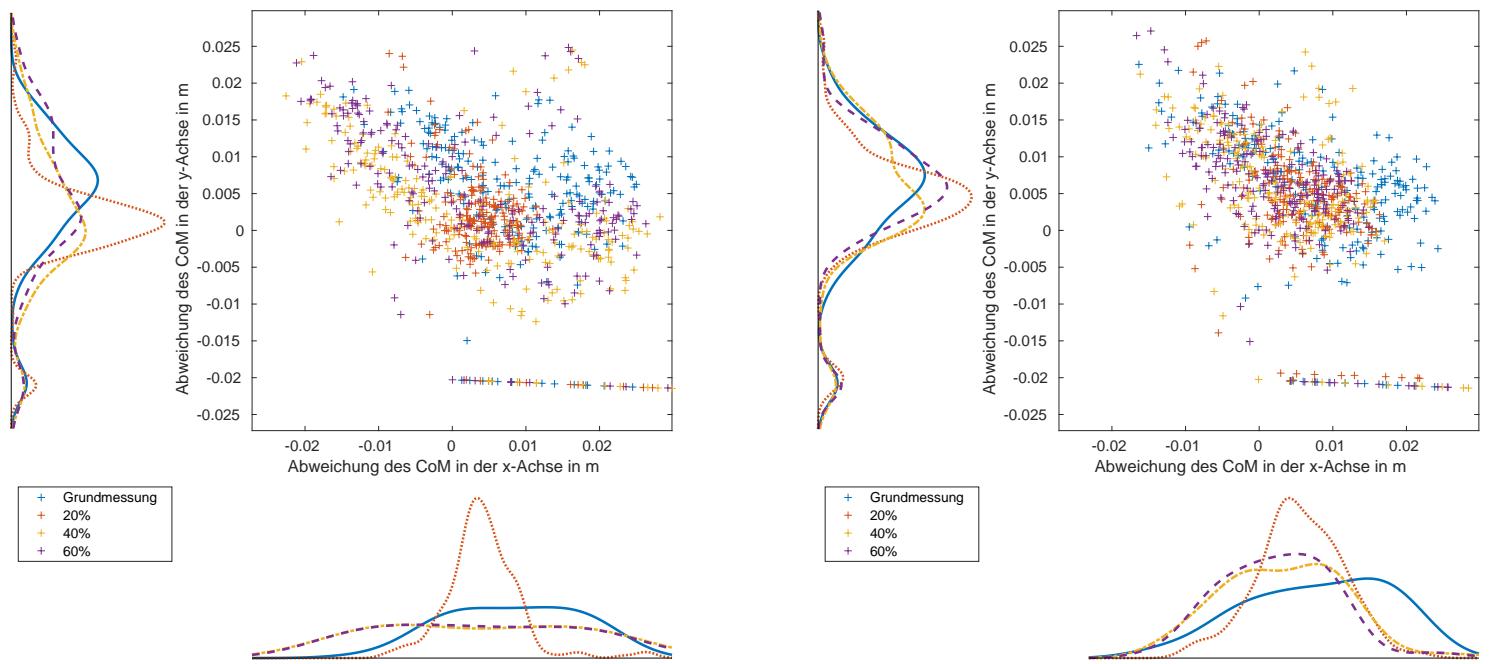


Abb. 16. Linker errechneter Massenschwerpunkt aufgenommen durch die FSR, x-Achse auf y-Achse aufgetragen, links ohne Magneten, rechts mit Magneten

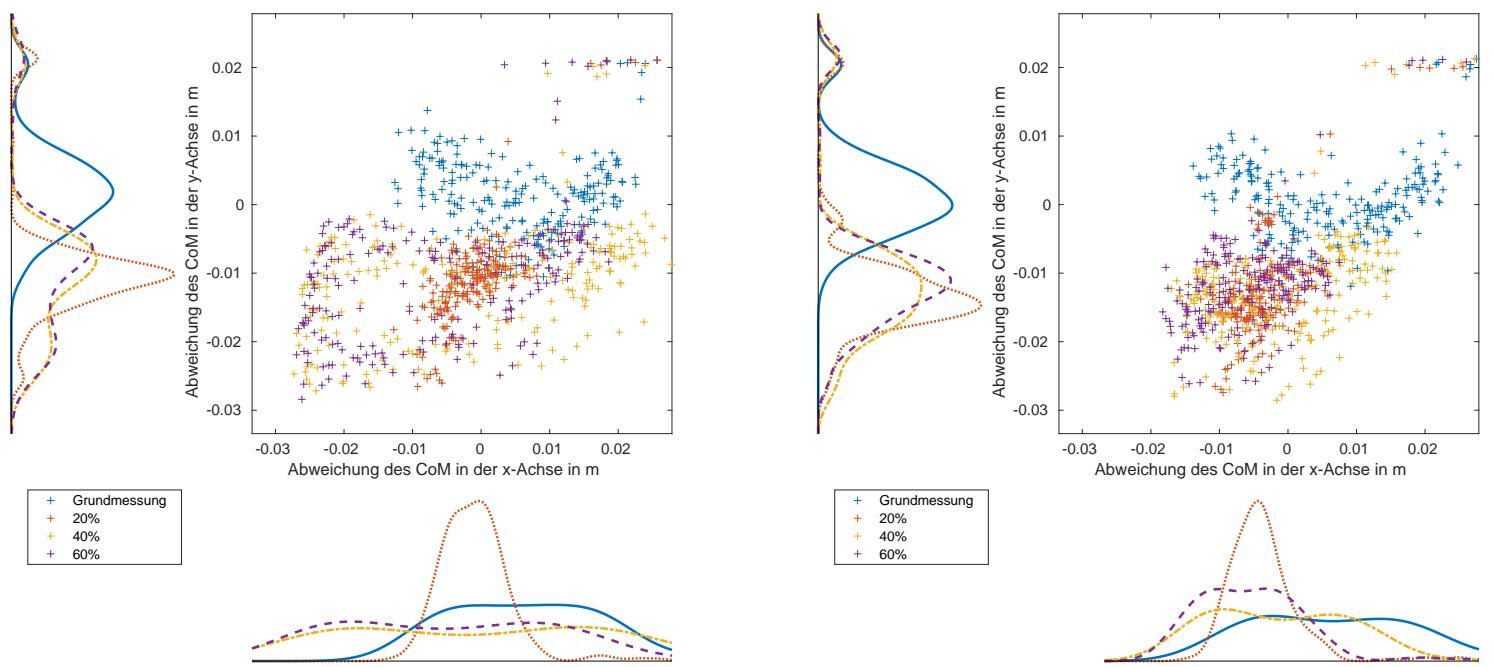


Abb. 17. Rechter errechneter Massenschwerpunkt aufgenommen durch die FSR, x-Achse auf y-Achse aufgetragen, links ohne Magneten, rechts mit Magneten

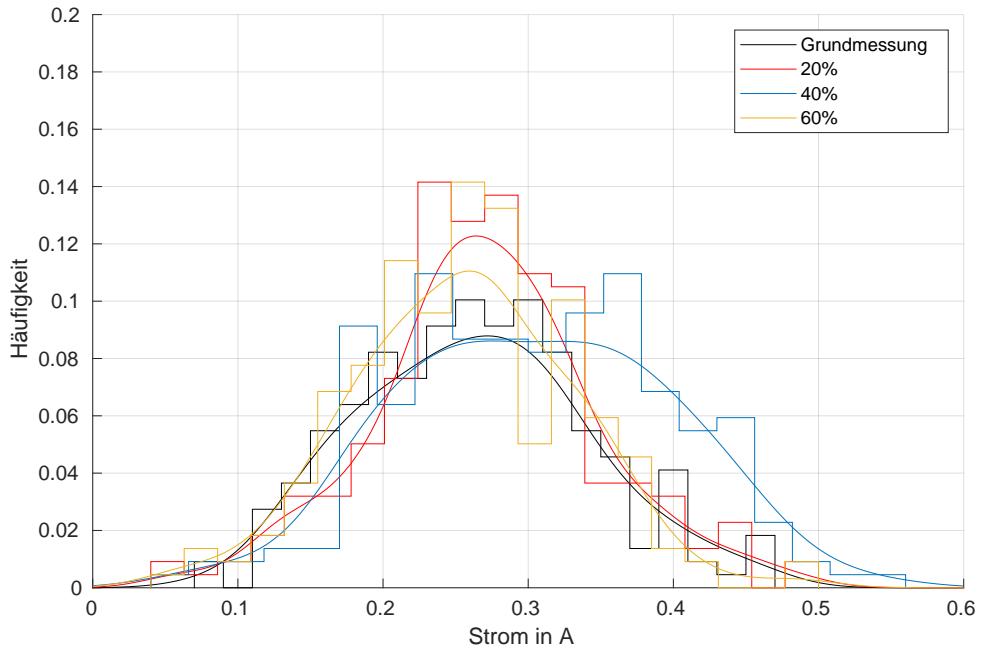
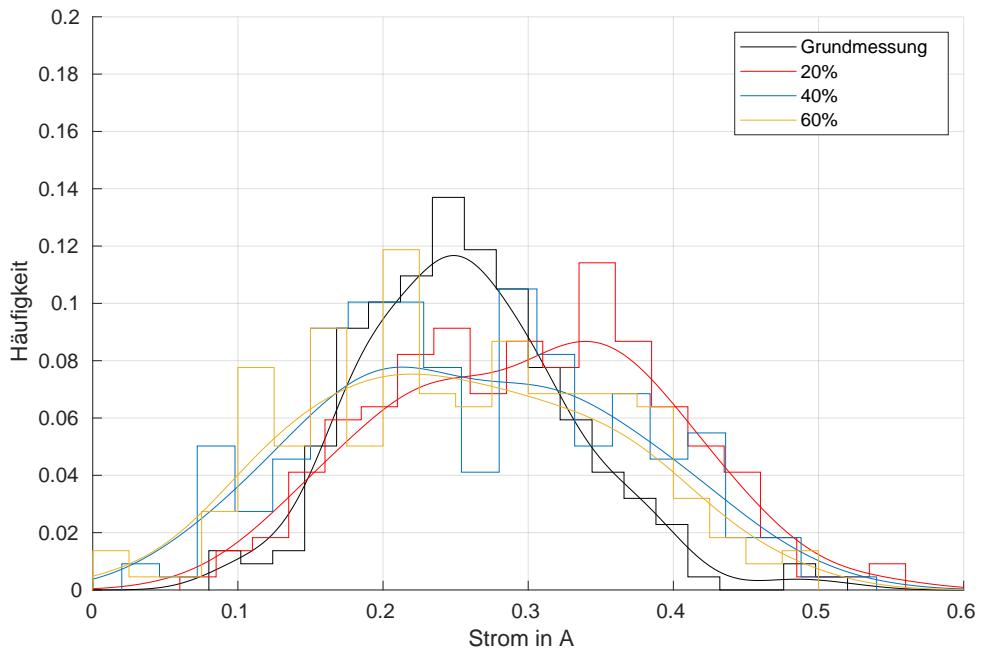


Abb. 18. AnklePitch gemessener Strom im linken Fuß, Strom in Ampère aufgetragen auf die Häufigkeit. Das Histogramm wurde ergänzt durch eine Wahrscheinlichkeitsdichtefunktion. Der obere Graph sind die Aufnahmen ohne Magneten, der untere mit Magneten.

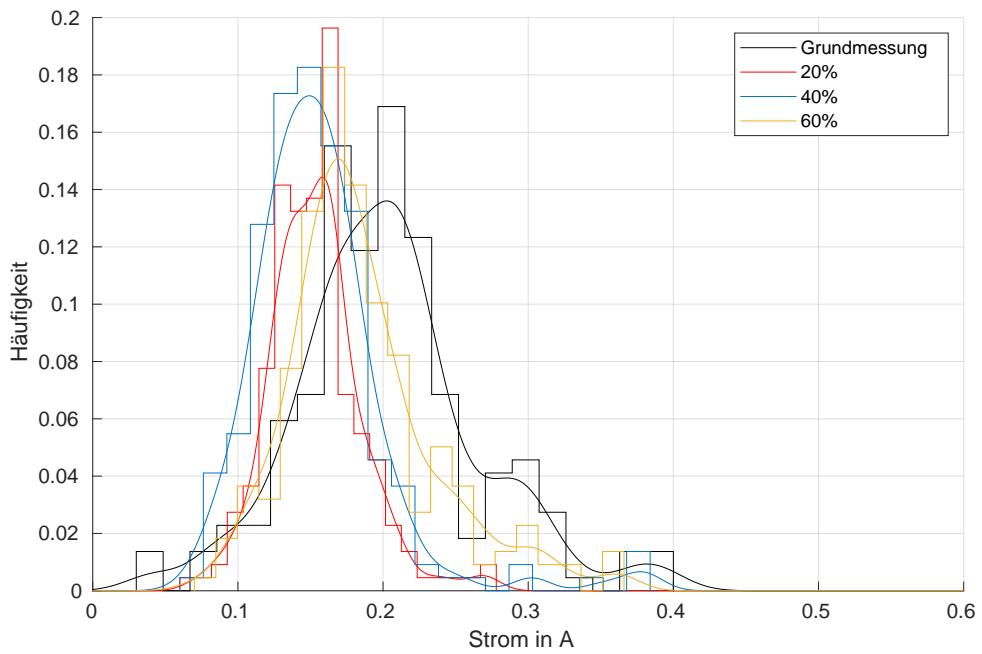
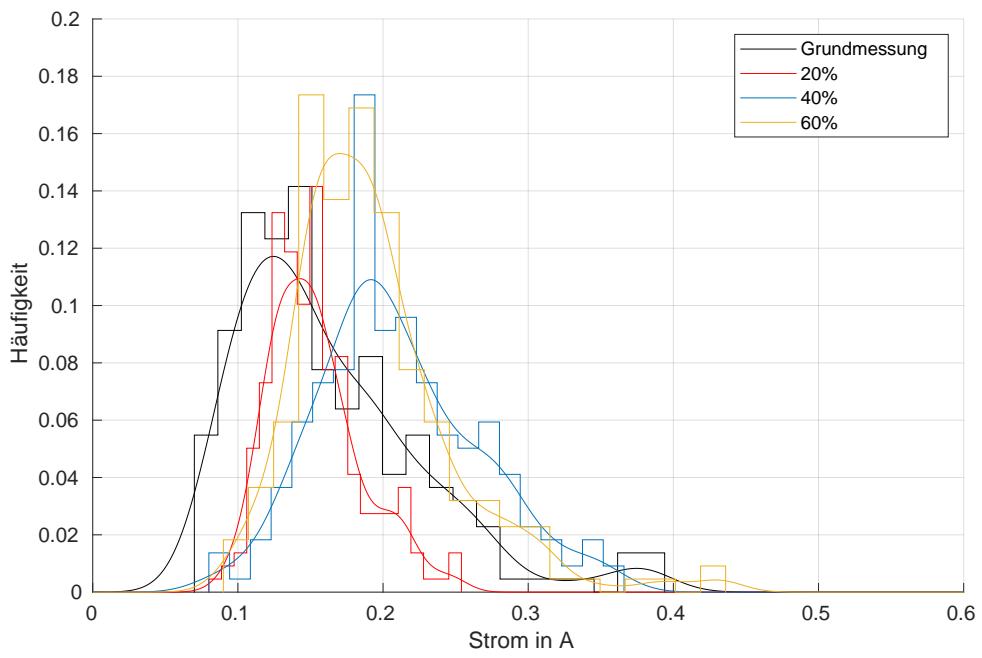


Abb. 19. AnklePitch gemessener Strom im rechter Fuß, Strom in Ampère aufgetragen auf die Häufigkeit. Das Histogramm wurde ergänzt durch eine Wahrscheinlichkeitsdichtefunktion. Der obere Graph sind die Aufnahmen ohne Magneten, der untere mit Magneten.

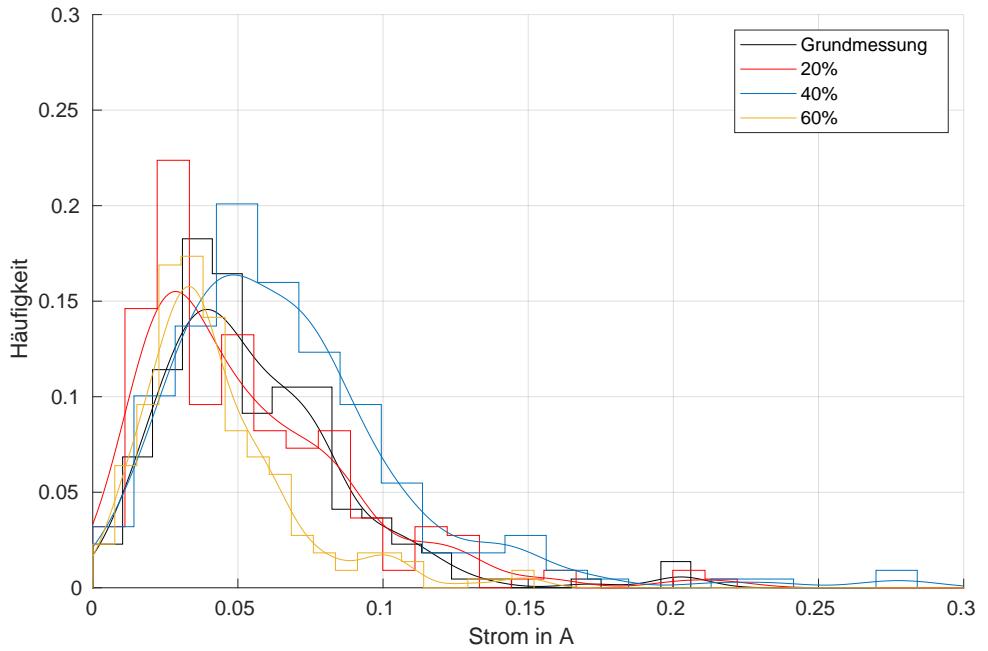
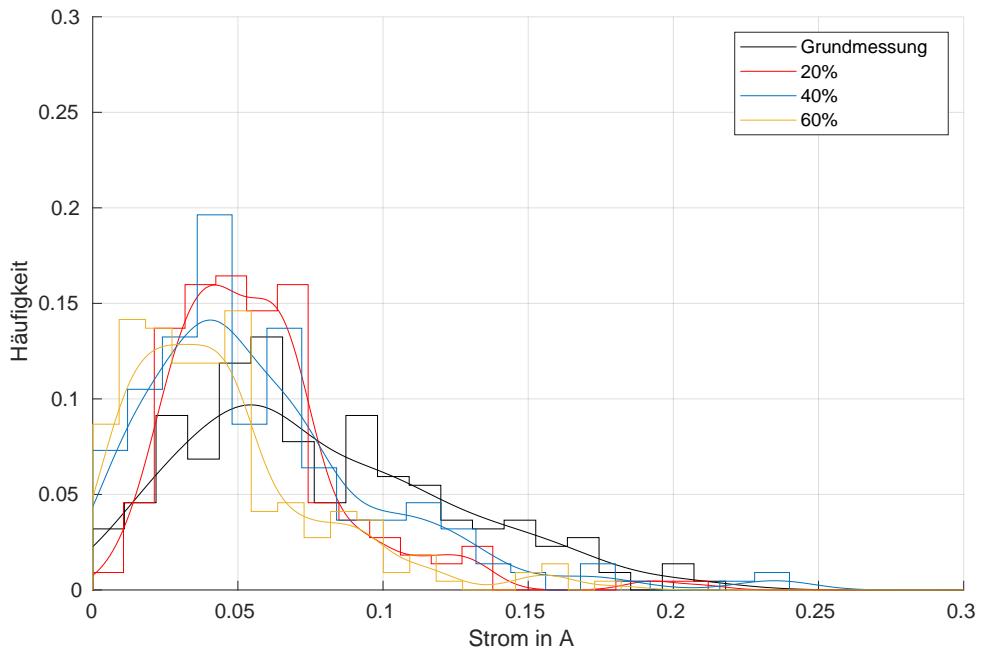


Abb. 20. AnkleRoll gemessener Strom im linken Fuß, Strom in Ampère aufgetragen auf die Häufigkeit. Das Histogramm wurde ergänzt durch eine Wahrscheinlichkeitsdichtefunktion. Der obere Graph sind die Aufnahmen ohne Magneten, der untere mit Magneten.

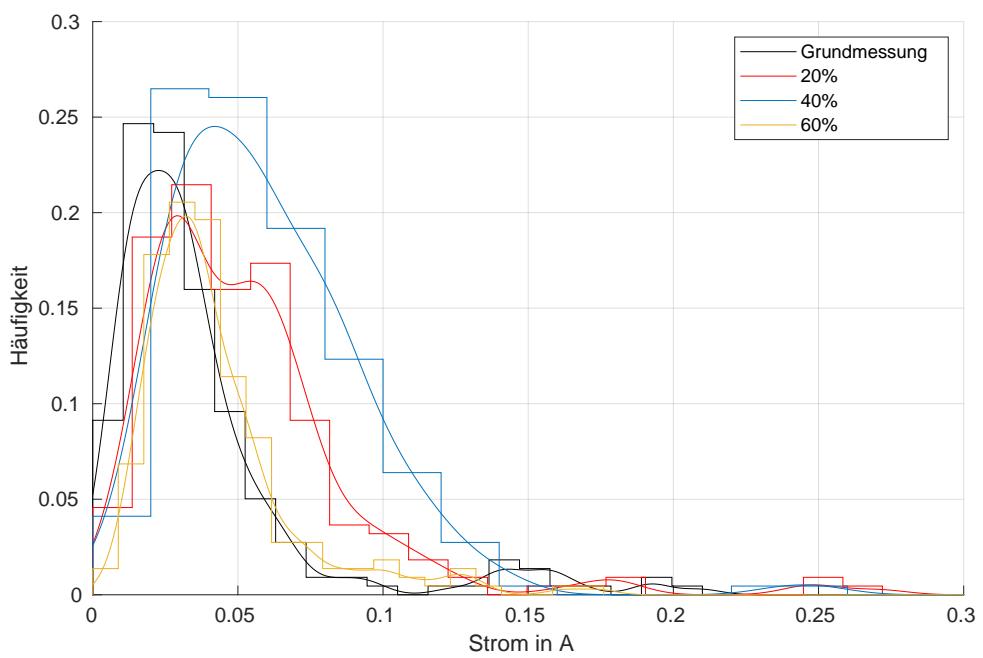
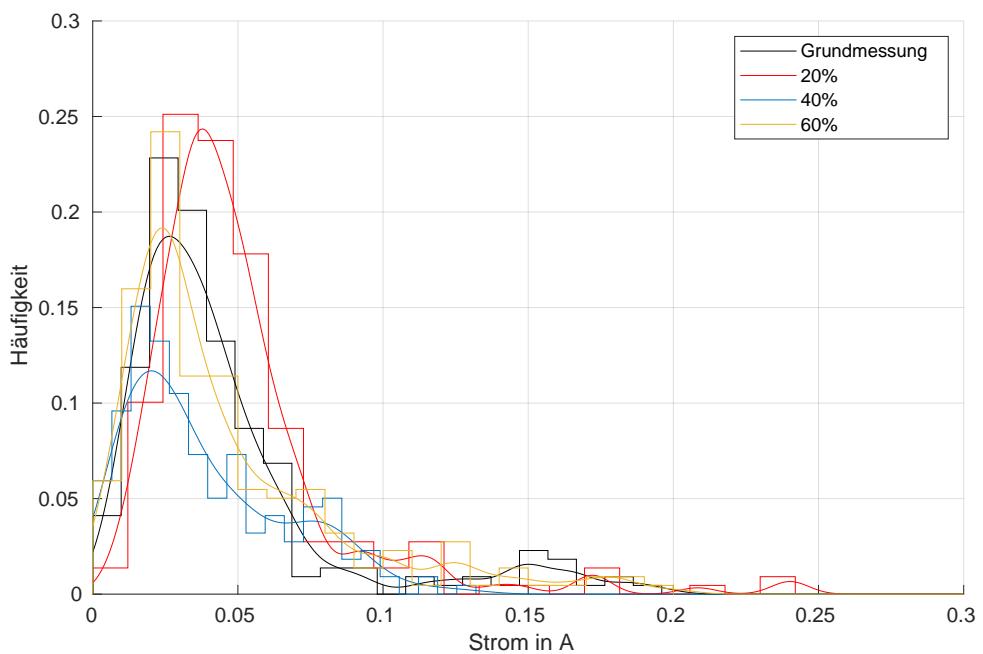


Abb. 21. AnkleRoll gemessener Strom im rechter Fuß, Strom in Ampère aufgetragen auf die Häufigkeit. Das Histogramm wurde ergänzt durch eine Wahrscheinlichkeitsdichtefunktion. Der obere Graph sind die Aufnahmen ohne Magneten, der untere mit Magneten.

6 Fazit und Ausblick

7 Anhang

Code 1. Pythonprogramm für Messaufnahmen

```
1 # ! /usr/bin/env python
2 # -*- encoding: UTF-8 -*-
3
4 """Example: Use getData Method to Use FSR Sensors"""
5
6 import qi
7 import argparse
8 import sys
9 import time
10 import csv
11 import re
12 import shutil
13 from tempfile import mkstemp
14 import os
15
16 def sed(pattern, replace, source, dest=None, count=0):
17     """Reads a source file and writes the destination file.
18
19     In each line, replaces pattern with replace.
20
21     Args:
22         pattern (str): pattern to match (can be re.pattern)
23         replace (str): replacement str
24         source (str): input filename
25         count (int): number of occurrences to replace
26         dest (str): destination filename, if not given,
27                     source will be over written.
28
29         fin = open(source, 'r')
30         num_replaced = count
31
32         if dest:
33             fout = open(dest, 'w')
34         else:
35             fd, name = mkstemp()
36             fout = open(name, 'w')
37
38         for line in fin:
39             out = re.sub(pattern, replace, line)
40             fout.write(out)
41
```

```

42         if out != line:
43             num_replaced += 1
44             if count and num_replaced > count:
45                 break
46     try:
47         fout.writelines(fin.readlines())
48     except Exception as E:
49         raise E
50
51     fin.close()
52     fout.close()
53
54     if not dest:
55         shutil.move(name, source)
56
57
58 def zeilen_aufteilen(file):
59     sed(',platzhalter,' , '\n' , file)
60     sed(',platzhalter' , '' , file)
61
62
63 def recordData(memory_service):
64     """ Get pressure sensor data from ALMemory
65     Returns a matrix of values
66
67     """
68     print "Recording data..."
69     data = list()
70     for range_counter in range(1, 220):
71         #Gyroscope
72         GyrX = memory_service.getData("Device/SubDeviceList
73             /InertialSensor/GyroscopeX/Sensor/Value")
74         GyrY = memory_service.getData("Device/SubDeviceList
75             /InertialSensor/GyroscopeY/Sensor/Value")
76         AngleX = memory_service.getData("Device/
77             SubDeviceList/InertialSensor/AngleX/Sensor/Value"
78             )
79         AngleY = memory_service.getData("Device/
80             SubDeviceList/InertialSensor/AngleY/Sensor/Value"
81             )
82         AccX = memory_service.getData("Device/SubDeviceList
83             /InertialSensor/AccelerometerX/Sensor/Value")
84         AccY = memory_service.getData("Device/SubDeviceList
85             /InertialSensor/AccelerometerY/Sensor/Value")
86         data.append(GyrX)
87         data.append(GyrY)

```

```

80     data.append(AngleX)
81     data.append(AngleY)
82     data.append(AccX)
83     data.append(AccY)
84
85     # Adding the summary of the FSR
86     LFsrTw = memory_service.getData("Device/
87         SubDeviceList/LFoot/FSR/TotalWeight/Sensor/Value"
88         )
89     RFsrTw = memory_service.getData("Device/
90         SubDeviceList/RFoot/FSR/TotalWeight/Sensor/Value"
91         )
92
93     LFcopX = memory_service.getData("Device/
94         SubDeviceList/LFoot/FSR/CenterOfPressure/X/Sensor
95         /Value")
96     LFcopY = memory_service.getData("Device/
97         SubDeviceList/LFoot/FSR/CenterOfPressure/Y/Sensor
98         /Value")
99     RFCopX = memory_service.getData("Device/
100        SubDeviceList/RFoot/FSR/CenterOfPressure/X/Sensor
101        /Value")
102     RFCopY = memory_service.getData("Device/
103        SubDeviceList/RFoot/FSR/CenterOfPressure/Y/Sensor
104        /Value")
105     data.append(LFsrTw)
106     data.append(RFsrTw)
107     data.append(LFcopX)
108     data.append(LFcopY)
109     data.append(RFcopX)
110     data.append(RFcopY)
111
112     # LeftAnkleRoll
113     PosAct = memory_service.getData("Device/
114         SubDeviceList/LAnkleRoll/Position/Actuator/Value"
115         )
116     PosSens = memory_service.getData("Device/
117         SubDeviceList/LAnkleRoll/Position/Sensor/Value")
118     ElectrSens = memory_service.getData("Device/
119         SubDeviceList/LAnkleRoll/ElectricCurrent/Sensor/
120         Value")
121     data.append(PosAct)
122     data.append(PosSens)
123     data.append(ElectrSens)
124
125     # RightAnkleRoll

```

```

109     PosAct = memory_service.getData("Device/
110         SubDeviceList/RAnkleRoll/Position/Actuator/Value"
111             )
110     PosSens = memory_service.getData("Device/
111         SubDeviceList/RAnkleRoll/Position/Sensor/Value")
111     ElectrSens = memory_service.getData("Device/
112         SubDeviceList/RAnkleRoll/ElectricCurrent/Sensor/
113             Value")
112     data.append(PosAct)
113     data.append(PosSens)
114     data.append(ElectrSens)
115
116     # LeftAnklePitch
117     PosAct = memory_service.getData("Device/
118         SubDeviceList/LAnklePitch/Position/Actuator/Value"
119             )
118     PosSens = memory_service.getData("Device/
119         SubDeviceList/LAnklePitch/Position/Sensor/Value")
119     ElectrSens = memory_service.getData("Device/
120         SubDeviceList/LAnklePitch/ElectricCurrent/Sensor/
121             Value")
120     data.append(PosAct)
121     data.append(PosSens)
122     data.append(ElectrSens)
123
124     # RightAnklePitch
125     PosAct = memory_service.getData("Device/
126         SubDeviceList/RAnklePitch/Position/Actuator/Value"
127             )
126     PosSens = memory_service.getData("Device/
127         SubDeviceList/RAnklePitch/Position/Sensor/Value")
127     ElectrSens = memory_service.getData("Device/
128         SubDeviceList/RAnklePitch/ElectricCurrent/Sensor/
129             Value")
128     data.append(PosAct)
129     data.append(PosSens)
130     data.append(ElectrSens)
131
132     data.append('platzhalter')
133     time.sleep(0.05)
134     return data
135
136
137 def count_files():
138     counter = 1

```

```

139     # str.zfill schreibt vor, wie lang die Zahl mit Nullen
140     # davor sein soll. also zfill(3) ist 3 Zahlen lang.
141     filename = 'measurement' + str(counter).zfill(3) + '.'
142             csv'
143
144     # Wenn das file nicht exisiert, erstelle measurement001
145             .csv
146     while os.path.exists(filename):
147         counter = counter + 1
148         filename = 'measurement' + str(counter).zfill(3) +
149             '.csv'
150         create_file(filename)
151     return filename
152
153
154
155 def main(session):
156     """
157         This example uses the getData method to use FSR sensors
158
159         .
160     """
161     # Get the ALProxy ALMemory and ALMotion
162     from naoqi import ALProxy
163     memory_service = session.service("ALMemory")
164     motion = ALProxy("ALMotion", "nao.local", 9559)
165
166
167     # wake up nao
168     motion.wakeUp()
169
170     motion.moveInit()
171     motion.post.moveTo(0.90, -0.2, -0.4, [[
172             "MaxStepFrequency", 0.0]])
173
174     data = recordData(memory_service)
175     filename = count_files()
176
177     output = os.path.abspath(filename)
178     with open(output, "wb") as file:
179         writer = csv.writer(file, delimiter=',')
180         writer.writerow(data)
181         zeilen_aufteilen(output)
182         print "Results written to", output

```

```

179     # go back to crouch position and sleep
180     motion.rest()
181
182
183 if __name__ == "__main__":
184     parser = argparse.ArgumentParser()
185     parser.add_argument("--ip", type=str, default="",
186                         127.0.0.1",
187                         help="Robot IP address. On robot or Local Naoqi: use
188                         '127.0.0.1'.")
189     parser.add_argument("--port", type=int, default=9559,
190                         help="Naoqi port number")
191
192     args = parser.parse_args()
193     session = qi.Session()
194     try:
195         session.connect("tcp://" + args.ip + ":" + str(args
196                         .port))
197     except RuntimeError:
198         print ("Can't connect to Naoqi at ip \\" + args.ip
199               + "\\" on port " + str(args.port) + ".\n"
200         "Please check your script arguments. Run with -h
201             option for help.")
202     sys.exit(1)
203     main(session)

```

Der Programmcode 1 kann in mehrere Funktionen aufgeteilt betrachtet werden. Die Funktion `sed` ist aus [48] entnommen und funktioniert wie die gleichnamige Funktion der Linux-Bash. Sie wird benötigt, um nach jedem Durchgang der Messschleife in `recordData` eine neue Zeile in die CSV Datei zu schreiben. Dies geschieht durch die Funktion `zeilen_aufteilen`. `recordData` wurde aus den Beispielen der NAO Dokumentation [9, /Other tutorials/Python SDK - Tutorials/Python SDK - Examples/Sensors] entnommen und angepasst, sodass am Ende jeder Zeile von `data` ein Platzhalter eingefügt wird und alle gewünschten Sensoren abgegriffen werden. Die Funktion `count_files` sorgt dafür, dass keine vorhandenen Messungen überschrieben werden und jede Messdatei eine fortlaufende Nummerierung erhält.

In der `main` Funktion werden `ALMemory` und `ALMotion` geladen, und der Gang einschließlich des Abgreifens der Sensorwerte ausgeführt. Die Ausgabe der Messwerte während dem Gang ist nur möglich durch den Präfix `post` vor `moveTo`.

Abschließend dient die letzte `if`-Abfrage zur Verbindung mit NAO, allerdings nur, wenn dieses Pythonprogramm selbst auf dem NAO liegt. Die Methode `post` sowie die Aufnahme der Sensoren während dem Lauf der Methode `moveTo` funktionieren nur lokal, deshalb ist es in diesem Fall nicht möglich, das Programm von dem eigenen Rechner aus zu starten. Mit anderen Methoden wäre eine Programmaufrufung über eine Wlan Verbindung durch-

aus möglich. Um Programme direkt auf dem NAO zu starten, wird eine ssh-Verbindung hergestellt und darüber dann python ausgeführt.

Literatur

- [1] Jean-Paul Pelteret und Paul Steinmann. *Magneto-Active Polymers - Fabrication, characterisation, modelling and simulation at the micro- and macro-scale*. De Gruyter, 2020. DOI: <https://doi.org/10.1515/9783110418576>.
- [2] SoftBank Robotics. *NAO - Documentation*. URL: <https://developer.softbankrobotics.com/nao6/nao-documentation/nao-developer-guide> (besucht am 21.12.2020).
- [3] URL: <https://spectrum.ieee.org/automaton/robotics/humanoids/aldebaran-robotics-founder-and-ceo-steps-down-softbank-appoints-new-leader> (besucht am 04.01.2021).
- [4] URL: <https://www.softbankrobotics.com/emea/en/pepper> (besucht am 04.01.2021).
- [5] Syamimi Shamsuddin u. a. "Humanoid Robot NAO Interacting with Autistic Children of Moderately Impaired Intelligence to Augment Communication Skills". In: *Procedia Engineering* 41 (2012). International Symposium on Robotics and Intelligent Sensors 2012 (IRIS 2012), S. 1533–1538. ISSN: 1877-7058. DOI: <https://doi.org/10.1016/j.proeng.2012.07.346>.
- [6] *NAO (NAOqi-2.1) NAO - documentation*. URL: <https://developer.softbankrobotics.com/nao-naoqi-2-1/nao-documentation/nao-technical-guide/> (besucht am 04.01.2021).
- [7] P. Shahverdi, M. J. Ansari und M. T. Masouleh. "Balance Strategy for Human Imitation by a NAO Humanoid Robot". In: *2017 5th RSI International Conference on Robotics and Mechatronics (ICRoM)*. 2017, S. 138–143. DOI: <10.1109/ICRoM.2017.8466225>.
- [8] A. M. Shayan u. a. "Design and Development of a Pressure-Sensitive Shoe Platform for Nao H25". In: *2019 7th International Conference on Robotics and Mechatronics (ICRoM)*. 2019, S. 223–228. DOI: <10.1109/ICRoM48714.2019.9071802>.
- [9] SoftBank Robotics. *NAOqi - Developer guide*. Accessed: 2020-12-21. URL: <https://developer.softbankrobotics.com/nao6/naoqi-developer-guide> (besucht am 21.12.2020).
- [10] Version 2020.3. URL: <https://www.autodesk.de/products/inventor/overview?plc=INVPROSA&term=1-YEAR&support=ADVANCED&quantity=1#internal-link-what-is-inventor> (besucht am 31.01.2021).
- [11] URL: <https://knowledge.autodesk.com/support/inventor/learn-explore/caas/CloudHelp/cloudhelp/2019/ENU/Inventor-Help/files/GUID-D74F47F3-FE22-44EF-85BE-7C6B1F56DCF9-htm.html> (besucht am 31.01.2021).
- [12] URL: <https://www.filamentworld.de/> (besucht am 12.03.2020).
- [13] Version 4.8.0. URL: <https://ultimaker.com/de/software/ultimaker-cura> (besucht am 31.01.2021).
- [14] Version R2020 Update 3. URL: <https://de.mathworks.com/discovery/what-is-matlab.html> (besucht am 31.01.2021).

- [15] URL: <https://de.mathworks.com/help/stats/exploratory-data-analysis.html> (besucht am 31.01.2021).
- [16] Ludwig Fahrmeir u.a. *Statistik. Der Weg zur Datenanalyse*. 8., überarb. und erg. Auflage. Springer Spektrum, Berlin, Heidelberg 2016. ISBN: 978-3-662-50371-3.
- [17] URL: <https://de.mathworks.com/help/stats/kernel-distribution.html> (besucht am 18.03.2020).
- [18] URL: <https://de.mathworks.com/help/stats/fitdist.html> (besucht am 18.03.2020).
- [19] Z. Rigbi und L. Jilken. "The response of an elastomer filled with soft ferrite to mechanical and magnetic influences". In: *J. Magn. Magn. Mater* 37(3) (Juli 1983), 267–276. DOI: [10.1016/0304-8853\(83\)90055-0](https://doi.org/10.1016/0304-8853(83)90055-0).
- [20] Z. Rigbi und J. E. Mark. "Effects of a magnetic field applied during the curing of a polymer loaded with magnetic filler". In: *J. Polym. Sci., Polym. Phys. Ed.* 23(6) (Juni 1985), 1267–1269. DOI: [10.1002/pol.1985.180230618](https://doi.org/10.1002/pol.1985.180230618).
- [21] J. M. Ginder. "Encyclopedia of Applied Physics". In: Bd. 16. Wiley-VCH Verlag GmbH & Co KGaA, New York, New York, 1996. Kap. Rheology controlled by magnetic fields, 487–503. DOI: [10.1002/3527600434.eap402](https://doi.org/10.1002/3527600434.eap402).
- [22] James E Martin u.a. "Magnetostriction of field-structured magnetoelastomers". In: *Physical review. E, Statistical, nonlinear, and soft matter physics* 74.5 Pt 1 (2006), S. 051507. ISSN: 1539-3755. DOI: [10.1103/physreve.74.051507](https://doi.org/10.1103/physreve.74.051507).
- [23] G. Filipcsei Z. Varga und M. Zrínyi. "Smart composites with controlled anisotropy". In: *Polymer* 46(18) (Aug. 2005), 7779–7787. DOI: [10.1016/j.polymer.2005.03.10](https://doi.org/10.1016/j.polymer.2005.03.10).
- [24] G. Filipcsei Z. Varga und M. Zrínyi. "Magnetic field sensitive functional elastomers with tuneable elastic modulus". In: *Polymer* 47(1) (Jan. 2006), 227–233. DOI: [10.1016/j.polymer.2005.10.139](https://doi.org/10.1016/j.polymer.2005.10.139).
- [25] John M. Ginder, William F. Schlotter und Mark E. Nichols. "Magnetochemical elastomers in tunable vibration absorbers". In: *Smart Structures and Materials 2001: Damping and Isolation*. Hrsg. von Daniel J. Inman. Bd. 4331. International Society for Optics and Photonics. SPIE, 2001, S. 103 –110. DOI: [10.1117/12.432694](https://doi.org/10.1117/12.432694).
- [26] H.X. Deng und X.L. Gong. "Adaptive Tuned Vibration Absorber based on Magnetorheological Elastomer". In: *Journal of Intelligent Material Systems and Structures* 18.12 (2007), S. 1205–1210. DOI: [10.1177/1045389X07083128](https://doi.org/10.1177/1045389X07083128).
- [27] N Hoang u.a. "Development of a torsional dynamic absorber using a magnetorheological elastomer for vibration reduction of a powertrain test rig". In: *Journal of Intelligent Material Systems and Structures* 24.16 (2013), S. 2036–2044. DOI: [10.1177/1045389X13489361](https://doi.org/10.1177/1045389X13489361).
- [28] M Kallio u.a. "Dynamic compression testing of a tunable spring element consisting of a magnetorheological elastomer". In: *Smart Materials and Structures* 16.2 (2007), S. 506–514. DOI: [10.1088/0964-1726/16/2/032](https://doi.org/10.1088/0964-1726/16/2/032).

- [29] John M. Ginder u. a. "Controllable-stiffness components based on magnetorheological elastomers". In: *Smart Structures and Materials 2000: Smart Structures and Integrated Systems*. Hrsg. von Norman M. Wereley. Bd. 3985. International Society for Optics und Photonics. SPIE, 2000, S. 418 –425. DOI: [10.1117/12.388844](https://doi.org/10.1117/12.388844).
- [30] Holger Böse, Raman Rabindranath und Johannes Ehrlich. "Soft magnetorheological elastomers as new actuators for valves". In: *Journal of Intelligent Material Systems and Structures* 23.9 (2012), S. 989–994. DOI: [10.1177/1045389X11433498](https://doi.org/10.1177/1045389X11433498).
- [31] Marc-André Keip, Paul Steinmann und Jörg Schröder. "Two-scale computational homogenization of electro-elasticity at finite strains". In: *Computer Methods in Applied Mechanics and Engineering* 278 (2014), S. 62–79. ISSN: 0045-7825. DOI: <https://doi.org/10.1016/j.cma.2014.04.020>.
- [32] G Y Zhou und Q Wang. "Design of a smart piezoelectric actuator based on a magnetorheological elastomer". In: *Smart Materials and Structures* 14.4 (2005), S. 504–510. DOI: [10.1088/0964-1726/14/4/007](https://doi.org/10.1088/0964-1726/14/4/007).
- [33] G Y Zhou, K C Lin und Q Wang. "Finite element studies on field-dependent rigidities of sandwich beams with magnetorheological elastomer cores". In: *Smart Materials and Structures* 15.3 (2006), S. 787–791. DOI: [10.1088/0964-1726/15/3/014](https://doi.org/10.1088/0964-1726/15/3/014).
- [34] Meng G. Zhang Wm. et al Wei Kx. "Experimental investigation on vibration characteristics of sandwich beams with magnetorheological elastomers cores". In: *J. Cent. South Univ. Technol.* 15 (2008), 239–242. DOI: [10.1007/s11771-008-0354-7](https://doi.org/10.1007/s11771-008-0354-7).
- [35] Famin Qiu und Bradley J. Nelson. "Magnetic Helical Micro- and Nanorobots: Toward Their Biomedical Applications". In: *Engineering* 1.1 (2015), S. 021–026. ISSN: 2095-8099. DOI: <https://doi.org/10.15302/J-ENG-2015005>.
- [36] Tiantian Xu u. a. "Magnetic Actuation Based Motion Control for Microrobots: An Overview". In: *Micromachines* 6 (Sep. 2015), S. 1346–1364. DOI: [10.3390/mi6091346](https://doi.org/10.3390/mi6091346).
- [37] Guo Zhan Lum u. a. "Shape-programmable magnetic soft matter". In: *Proceedings of the National Academy of Sciences* (2016). ISSN: 0027-8424. DOI: [10.1073/pnas.1608193113](https://doi.org/10.1073/pnas.1608193113).
- [38] Wenqi Hu u. a. "Small-scale soft-bodied robot with multimodal locomotion". In: *Nature* 554.7690 (2018), S. 81–85. DOI: [10.1038/nature25443](https://doi.org/10.1038/nature25443).
- [39] Roland Fuhrer u. a. "Soft Iron/Silicon Composite Tubes for Magnetic Peristaltic Pumping: Frequency-Dependent Pressure and Volume Flow". In: *Advanced Functional Materials* 23.31 (2013), S. 3845–3849. DOI: <https://doi.org/10.1002/adfm.201203572>.
- [40] Yancheng Li u. a. "A highly adjustable magnetorheological elastomer base isolator for applications of real-time adaptive control". In: *Smart Materials and Structures* 22.9 (2013), S. 095020. DOI: [10.1088/0964-1726/22/9/095020](https://doi.org/10.1088/0964-1726/22/9/095020).
- [41] Song Qi u. a. "3D printed shape-programmable magneto-active soft matter for biomimetic applications". In: *Composites Science and Technology* 188 (2020), S. 107973. ISSN: 0266-3538. DOI: <https://doi.org/10.1016/j.compscitech.2019.107973>.

- [42] Dirk Sindersberger u. a. "Printing of hybrid magneto active polymers with 6 degrees of freedom". In: *Materials Today Communications* 15 (2018), S. 269–274. ISSN: 2352-4928. DOI: <https://doi.org/10.1016/j.mtcomm.2018.02.032>.
- [43] URL: https://electronics-electric.bASF.com/global/en/electronics/products/carbonyl_iron_powder.html (besucht am 22.03.2020).
- [44] Sergei Gorodkin, R James und William Kordonski. "Magnetic Properties of Carbonyl Iron Particles in Magnetorheological Fluids". In: *Journal of Physics: Conference Series* 149 (März 2009), S. 012051. DOI: [10.1088/1742-6596/149/1/012051](https://doi.org/10.1088/1742-6596/149/1/012051).
- [45] Taixiang Liu und Yangguang Xu. "MagnetoRheological Elastomers: Materials and Applications". In: März 2019. ISBN: 978-1-78984-716-1. DOI: [10.5772/intechopen.85083](https://doi.org/10.5772/intechopen.85083).
- [46] C. Lutz u. a. "NAO walking down a ramp autonomously". In: *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2012, S. 5169–5170. DOI: [10.1109/IROS.2012.6386268](https://doi.org/10.1109/IROS.2012.6386268).
- [47] URL: https://www.supermagnete.de/magnete-zum-anschrauben/quadermagnet-40-x-40-x-4mm_CS-Q-40-40-04-N (besucht am 06.01.2021).
- [48] URL: <https://stackoverflow.com/questions/12714415/python-equivalent-to-sed> (besucht am 08.03.2020).