



DIPLOMARBEIT

Herr M.Sc.
Felix Hillemeier

**Realisierung eines technischen Mock-up zur
Robotersteuerung mittels haptischen Feedbacks**

Mittweida, 2022

Fakultät Ingenieurwissenschaften

DIPLOMARBEIT

Realisierung eines technischen Mock-ups zur Robotersteuerung mittels haptischen Feedbacks

Autor:
Herr M.Sc.

Felix Hillemeier

Studiengang:
Industrial Engineering

Seminargruppe:
IE20w1-D

Erstprüfer:
Prof. Dr.-Ing. habil. Alexander Winkler

Zweitprüfer:
M.Sc. Christian Thormann

Einreichung:
Mittweida, 30.04.2022

Verteidigung/Bewertung:
Mittweida, 2022

Inhaltsverzeichnis

Inhaltsverzeichnis.....	II
Abbildungsverzeichnis.....	IV
Tabellenverzeichnis.....	VI
Abkürzungsverzeichnis.....	VII
Abstract - Zusammenfassung.....	IX
1 Einleitung	1
1.1 <i>Relevanz von Robotersteuerung.....</i>	<i>1</i>
1.2 <i>Zielsetzung und Forschungsfrage</i>	<i>2</i>
1.3 <i>Methodik und Abgrenzung der Thematik.....</i>	<i>3</i>
2 Theoretisches Fundament	4
2.1 <i>Roboter.....</i>	<i>4</i>
2.1.1 <i>Roboterarten.....</i>	<i>7</i>
2.1.2 <i>Roboterarm.....</i>	<i>8</i>
2.1.3 <i>Kinematik.....</i>	<i>10</i>
2.1.4 <i>Vorwärtstransformation</i>	<i>11</i>
2.1.5 <i>Rückwärtstransformation</i>	<i>12</i>
2.2 <i>Kollaborative Roboter (Cobots).....</i>	<i>13</i>
2.3 <i>Robotersteuerung</i>	<i>15</i>
2.3.1 <i>Feedback.....</i>	<i>20</i>
2.4 <i>Human Factors</i>	<i>22</i>
2.4.1 <i>Mensch-Maschine-Schnittstelle.....</i>	<i>23</i>
2.4.2 <i>Gamification.....</i>	<i>24</i>
3 Methodik und Umsetzung	26
3.1 <i>Technischer Mock-up.....</i>	<i>26</i>

3.2	<i>Sony DualSense Controller</i>	28
3.3	<i>Cobot von Universal Robots</i>	33
3.3.1	<i>Interfaces</i>	35
3.3.2	<i>Sensorik</i>	37
3.4	<i>Script</i>	38
3.4.1	<i>Python</i>	39
3.4.2	<i>Kommunikation zum Roboter</i>	40
3.4.3	<i>Kommunikation zum Controller</i>	42
3.4.4	<i>Grafische Benutzeroberfläche</i>	42
4	Ergebnisse	44
4.1	<i>Steuerungskonzept</i>	44
4.2	<i>Steuerungsprogramm mit haptischem Feedback</i>	47
4.2.1	<i>Daten auslesen</i>	49
4.2.2	<i>Datenberechnung und Datentransformation</i>	49
4.2.3	<i>Hauptfunktionen und Datenübergabe</i>	52
4.3	<i>Anwendung</i>	56
4.4	<i>Sicherheitsbetrachtung</i>	58
5	Diskussion	60
5.1	<i>Beantwortung der Forschungsfrage</i>	60
5.2	<i>Diskussion der Ergebnisse</i>	61
5.3	<i>Methodische Limitationen</i>	61
5.4	<i>Schlussbetrachtung</i>	62
	Attachments	V
	Literaturverzeichnis	V
	Selbstständigkeitserklärung	XII

Abbildungsverzeichnis

Abbildung 1: Publikationen und Installationen zu Robotern in Anlehnung an Dzedzickis et al. 2022	6
Abbildung 2: Roboterportfolio des Herstellers Fanuc	7
Abbildung 3: Marktanteil nach Roboterart nach Pott und Dietz (2019).....	8
Abbildung 4: Koordinatensysteme von Industrierobotern nach Zeichen und Fürst (2000).....	9
Abbildung 5: Koordinatensysteme und Achsdrehung für Eulerwinkel nach Husty (1997).....	10
Abbildung 6: Mensch-Roboter-Kollaboration nach dem Institut Chemnitzer Maschinen- und Anlagenbau e.V.	14
Abbildung 7: Arten der Roboterprogrammierung nach Weber (2017)	16
Abbildung 8: Handprogrammiergeräte oder Steuerungen unterschiedlicher Roboterhersteller nach eigener Recherche	17
Abbildung 9: Relevanz von Gamification nach eigener Darstellung	24
Abbildung 10: Skizze des Mock-ups nach eigener Darstellung	26
Abbildung 11: Technische Skizze des Sony DualSense Controllers	30
Abbildung 12: Elektronik und Innenleben des Sony DualSense Controllers	31
Abbildung 13: Schematischer Aufbau der Elektronik des Controllers.....	31
Abbildung 14: Aktuelles Produktportfolio von Universal Robots mit technischen Spezifikationen	34
Abbildung 15: Richtung der einwirkenden Kräfte und Drehmomente am Roboterflansch	37
Abbildung 16: Spezifikationen der Kraft- und Momentensensorik aus den Datenblättern der UR-Serie	38
Abbildung 17: Schematische Darstellung zur Funktion von dem Pythonmodul ‚ur_rtde‘	40
Abbildung 18: Steuerungskonzept des PS5 Controllers	46
Abbildung 19: Schematische Aufbau der Software zur haptischen Robotersteuerung nach eigener Darstellung.....	47

Abbildung 20: Sigmoidfunktion der drei Parameter des haptischen Feedbacks nach eigener Darstellung	51
Abbildung 21: GUI des Programms zum Steuern des Roboters	53
Abbildung 22: Programmbeispiel der Bewegungssteuerung mittels der ServoJ Funktion	54
Abbildung 23: while-Funktion der erstellten Software.....	55

Tabellenverzeichnis

Tabelle 1: Nutzwertanalyse der Eingabegeräte	29
Tabelle 2: Eingabegeräte mit haptischem Feedback (1/4)	V
Tabelle 3: Eingabegeräte mit haptischem Feedback (2/4)	VI
Tabelle 4: Eingabegeräte mit haptischem Feedback (3/4)	VII
Tabelle 5: Eingabegeräte mit haptischem Feedback (4/4)	VIII

Abkürzungsverzeichnis

API	Application Programming Interface
CNC	Computerized Numerical Control
DIN	Deutsches Institut für Normung
DLR	Deutsches Zentrum für Luft- und Raumfahrt
EN	Europäische Normen
GUI	graphical user interface
IEEE	Institute of Electrical and Electronics Engineers
IO	input/output
IP	Internet Protocol
ISO	Internationale Organisation für Normung
ISS	International Space Station
LAN	Local Area Network
MIT	Massachusetts Institute of Technology
NC	Numerical Control
OS	Operating System
RTDE	Real-Time Data Exchange
SCARA	Selective Compliance Assembly Robot Arm
SDK	Software Development Kit
Tcl	Tool Command Language

TCP	Tool Center Point
TCP/IP	Transmission Control Protocol / Internet Protocol
TK	Toolkit
UR	Universal Robot
USB	Universal Serial Bus
WLAN	Wireless Local Area Network

Abstract - Zusammenfassung

Ziel dieses Projekts ist es, einen kollaborativen Roboterarm mit sechs Freiheitsgraden, der ein haptisches Feedback an den Roboterbediener übermittelt, mittels Gamecontroller zu steuern. Der Cobot ist so konstruiert und ausgelegt, dass eine Zusammenarbeit mit dem Menschen ermöglicht wird. Der Roboter, der mit der entsprechenden Sensorik ausgestattet ist, kann auf ihn einwirkende Kräfte und Momente fühlen und in Gefahrensituationen einen Not-Halt vornehmen.

Die vom Roboter aufgenommenen Kräfte werden mithilfe des Controllers an den Roboterbediener übertragen, sodass dieser empfinden kann, was der Roboter durch seine Sensoren ertastet. Ziel ist, die Robotik intuitiver und erlebbarer zu machen. Mit Konzepten der Gamification soll die Hemmschwelle für das Ausprobieren der komplexen Roboterprogrammierung und -steuerung genommen werden. So können neue Anwendungsfälle erschlossen und Lösungen für die Problematik des Fachkräftemangels erarbeitet werden.

Es wurde im Laufe des Projektes ein Roboter-Steuerungskonzept für den DualSense™ Wireless-Controller der Firma Sony entworfen und umgesetzt. Mit Python wurde eine Software mit Benutzeroberfläche für den Roboterbediener programmiert, um die Steuerung zu realisieren und das Konzept des haptischen Feedbacks umzusetzen. Als wichtige Mensch-Roboter-Schnittstelle wurden Roboterdaten im Graphical User Interface (GUI) visualisiert und aufbereitet. Somit wurde ein technisches Mock-up realisiert, das als Proof of Concept dient. Die Machbarkeit eines Roboter-Steuerungssystems mit haptischem Feedback in kostengünstiger und jedem zugänglicher Weise wurde demonstriert. Dieses kann nun weiterentwickelt und genutzt werden.

1 Einleitung

Diese Abschlussarbeit hat die Zielsetzung, die Forschung im Themengebiet der Robotik anwendungsbezogen zu erweitern. In der inhaltlichen Abhandlung wird die Relevanz der Robotik und deren Steuerung in der Praxis mit aktuellen Trends der Mensch-Roboter-Kollaboration dargestellt und analysiert. Hierauf aufbauend wird ein technischer Mock-up kreiert, der die Bedienung von Robotern innovativ verändert. Die Arbeit gliedert sich in fünf Kapitel. In diesem ersten Kapitel, der Einleitung, wird die wissenschaftliche Relevanz des Themas herausgearbeitet und es werden die Zielsetzung und die Forschungsfrage der Arbeit erläutert. Anschließend wird in Kapitel 2 mit dem theoretischen Fundament die inhaltliche Basis gelegt. Hierzu werden technische Grundlagen und das theoretische Fundament (u. a. Nomenklatur, Roboter, Feedback und Steuerung) sowie relevante Konzepte erläutert. Im Fokus des dritten Kapitels steht die Methodik, also der Aufbau und die Umsetzung des Technologie-Mock-ups, mit den Entscheidungen und Entwicklungen im Verlauf des Projekts. Darauf aufbauend werden die finalen Ergebnisse im vierten Kapitel vorgestellt. Im letzten Kapitel, der Diskussion, werden die Forschungsfrage beantwortet, die Ergebnisse kritisch analysiert und in der Schlussbetrachtung die Implikationen sowie die Limitationen zukünftiger Forschung beleuchtet.

1.1 Relevanz von Robotersteuerung

Roboter mit Gefühlen – Industrieroboter sind Maschinen, die starr einprogrammierte Aufgaben ausführen. Sie haben im weitesten Sinne keine Empfindungen und, müssen dennoch teilweise feinmotorische Aufgaben mit zerbrechlichen Objekten ausführen. Doch wie kann es dem Roboterbediener ermöglicht werden, feinfühlig und präzise Programme zu erstellen? Dazu sollte er empfinden, was der Roboter (durch seine Sensoren) ertastet. Über haptisches Feedback in der Bedienung über einen Joystick bzw. ein Eingabegerät kann dies gelingen. Dem Roboterbediener soll es über die neuartige Steuerung möglich sein, in Echtzeit das zu spüren, was der Endeffektor an Kräften erfährt. Hierzu soll mit

einem Kraft-Momenten-Sensor oder über die Gelenkströme die Kraft (und Richtung der Kraft), die der Endeffektor erfährt, gemessen und dem Bediener mittels der Steuerung als haptisches Signal übermittelt werden. Das haptische Feedback soll den Roboterbediener dazu befähigen, den taktilen Reiz zu fühlen. Das soll die Teach-in-Programmierung in Echtzeit verbessern und neue Möglichkeiten der Anwendung realisierbar machen. Mithilfe der nun gegebenen Sensibilität wird es dem Roboterbediener ermöglicht, feinfühlig Aufgaben mit zerbrechlichen oder sensiblen Gegenständen auszuführen. Das bietet potenzielle Use-Cases in der Medizin bei Operationen, in den künstlerischen Bereichen, z. B. beim Töpfern, Malen, Zeichnen, Schreiben oder Kalligraphieren, beim Schleifen, Entgraten oder während filigraner Handlings- bzw. Manipulationsaufgaben. Es könnte auch ein bedeutender Aspekt der Mensch-Roboter-Kollaboration werden, da mittels der Steuerung eine Art Feingefühl auf den Roboter übertragen werden kann. Aufgrund der Neuartigkeit und fehlenden Marktreife sind klare wirtschaftliche Anwendungsfälle noch nicht abzusehen.

1.2 Zielsetzung und Forschungsfrage

Der Schwerpunkt der Abschlussarbeit ist die softwareseitige Realisierung eines technischen Mock-ups, der in die Bedingung eines Roboters haptisches Feedback inkludiert. Die Umsetzung erfolgt an einem kollaborativen Roboter der Marke Universal Robots und einem proprietären Joystick bzw. Controller. Mit dem Mock-up soll die Realisierung der Idee des haptischen Feedbacks bewiesen und ausgetestet werden. Die Möglichkeiten des ausgewählten Eingabegeräts sollen ausgeschöpft werden, um auch andere Feedbacksignale (audio/visuell) zu implementieren. Die Forschungsfrage nach den existierenden Möglichkeiten, einen 6-Achs-Roboterarm feinfühlig zu steuern, ist deshalb von großer Relevanz in der aktuellen Forschung der Robotik und wird im Rahmen der vorliegenden Arbeit behandelt.

1.3 Methodik und Abgrenzung der Thematik

Um die Forschungsfrage zu beantworten und dem Ziel der Arbeit gerecht zu werden, wird ein technischer Mock-up zur Demonstration erstellt. Ein Mock-up ist ein Prototyp, um ein Konzept oder eine Idee vorzustellen. Dieser dient als Proof of Concept einer Robotersteuerung mit haptischem Feedback für einen kollaborierenden 6-Achs-Roboterarm. Die Analyse beschränkt sich dabei auf die Erstellung eines realen Demonstrators und eines möglichen Bedienkonzepts. Auf der Grundlage des Mock-ups soll der Roboterbediener eine innovative Möglichkeit zur Handhabung des Manipulators haben. Ziel der Arbeit ist es, die Robotersteuerung neu zu denken und zu erweitern. Im theoretischen Fundament wird eine Literaturrecherche als Methodik angewandt. Relevante Literatur der aktuellen Forschung und Wissenschaft zu dem Thema, die als fundamentale Grundlage der Abschlussarbeit fungiert, wurde durch eine Online-recherche eruiert. Hierbei dienten die Begriffe ‚robotic control‘, ‚sensoric feedback‘, ‚haptical joystick‘ und ‚force control robotics‘ und deren deutsche Übersetzungen in den Onlinebibliotheken des IEEE (Institute of Electrical and Electronics Engineers), google.scholar.de, ScienceDirect und SpringerLink als Sucheinstellungen. Weiterhin wurde die Bibliothek der Hochschule Mittweida zur Literaturrecherche genutzt. Die Veröffentlichungen, die für diese Abhandlung relevant sind, setzen sich hauptsächlich mit der Steuerung von Robotern oder anderen Maschinen auseinander und greifen das Thema haptisches Feedback auf. Hierauf aufbauend werden auch die Entscheidungen zur Realisierung des technischen Mock-ups abgeleitet, um Wissenschaft und Praxis in einem anwendungsbezogenen Ergebnis zu vereinen.

2 Theoretisches Fundament

In diesem Kapitel werden die theoretischen Grundlagen anhand der wissenschaftlichen Literatur bestimmt und zentrale Fachbegriffe definiert und eingegrenzt. Es werden alle Themengebiete der Robotik, die zum Erstellen des Mock-ups relevant sind, nacheinander abgehandelt und die Implikationen der Umsetzung zur aktuellen Forschung hervorgehoben, angefangen mit allgemeinen Themen der Robotik (Kapitel 2.1 und 2.2) hin zu konkreten Forschungsg Grundlagen zu Steuerungskonzepten (Kapitel 2.3) von Robotern und der Ausgestaltung der Bedienung (Kapitel 2.4).

2.1 Roboter

Der Begriff ‚Roboter‘ entstammt dem slawischen Wort ‚robota‘ für Fronarbeit und erlangte Popularität in einem dystopischen Theaterstück zu Beginn des 20. Jahrhunderts, in dem künstliche Menschen gebaut werden, um den Menschen zu entlasten. Zu dieser Zeit wurden zerspanende Arbeiten noch mit handgeführten Werkzeugmaschinen ausgeübt. Ab der zweiten Hälfte des 20. Jahrhunderts wurden nunmehr NC-Maschinen (NC: Numerical Control) entwickelt, die im Laufe der Zeit durch CNC-Maschinen (CNC: Computerized Numerical Control) abgelöst wurden und die handgeführten Maschinen automatisierten. Dieser Entwicklung entstammte George Charles Devols Idee einer Maschine, die heute als ‚Industrieroboter‘ bezeichnet wird. Mit dem Patent ‚Programmed article transfer‘ (Devol, JR 1961) sicherte er seine Idee und baute mit dem ‚Unimate 1900‘ den ersten Industrieroboter, der für General Motors erhitzte Werkstücke aus der Druckgussmaschine entnahm. Später wurde der Roboter für das Punktschweißen von Automobilkarosserien in der Fertigungsstraße eingesetzt (Mareczek 2020a, S. 1–8). Im Jahr 1978 wurde mit dem ‚PUMA‘ (programmable universal machine for assembly) von Unimation die nächste Evolutionsstufe der Industrieroboter vorgestellt. Er war elektrisch angetrieben und wies mit seinen sechs Achsen große Ähnlichkeit zu heutigen Industrierobotern auf. (Haun 2013, S. 5) An diesem Grundkonzept des Industrieroboters hat sich nur wenig geän-

dert. Dennoch steckt in diesen Robotern eine Reihe von Technologien, die sich stetig weiterentwickeln: Antrieb, Steuerung, Sensoren, Mechanik und Informationstechnologie(Haun 2013, S. 9).

Die deutsche Normung definiert einen ‚Industrieroboter‘ als:

„Automatisch gesteuerter, frei programmierbarer Mehrzweck-Manipulator, der in drei oder mehr Achsen programmierbar ist und zur Verwendung in der Automatisierungstechnik entweder an einem festen Ort oder beweglich angeordnet sein kann.“ (DIN EN ISO 10218-1:2011)

Industrieroboter werden heutzutage für die unterschiedlichsten Tätigkeiten in der Industrie eingesetzt, unter anderem zur Durchführung von Aufgaben wie Pick&Place, Schweißen, Lackieren, Verpacken und Etikettieren, Bearbeiten, Montieren, Palettieren und die Produktinspektion (Ji und Wang 2019) (Nguyen und Melkote 2019). Viele Industrieroboter haben sechs oder mehr Achsen und können diverse Endeffektoren ansteuern. Der Endeffektor bestimmt die Arbeitstätigkeit und ist das Gerät am Ende des Roboterarms, das für die Interaktion zwischen dem Roboter und seiner Umgebung sorgt (Tlach et al. 2017). Deshalb haben sie in ihren Aufgabengebieten einen hohen Grad an Flexibilität und Freiheiten. Sie können für monotone, gefährliche, repetitive oder präzise Aufgaben eingesetzt werden.

Heutzutage existieren viele Arten von Maschinen, die als Roboter bezeichnet werden. Alle haben unterschiedliche Kinematiken, Gelenkanordnungen, Freiheitsgrade, sowie Anwendungsgebiete. Das Technologiegebiet ist trotz seiner über 60-jährigen Geschichte immer noch ein relevantes und innovatives Forschungsgebiet. Neue Technologien und abstrakte Anwendungsfelder werden stetig in Kombination mit Robotern getestet und eruiert. Die Rate der Automatisierung und die Anzahl der Robotereinsätze in der Industrie nimmt immer weiter zu. Aufgrund der geringer werdenden Preise für Roboterarme und neuen technologischen Möglichkeiten bieten sich weitere Potenziale für die Robotik. Sowohl die Installationen an Robotern als auch wissenschaftliche Publikationen in dem Gebiet steigen stetig an (siehe Abbildung 1)(Dzedzickis et al. 2022).

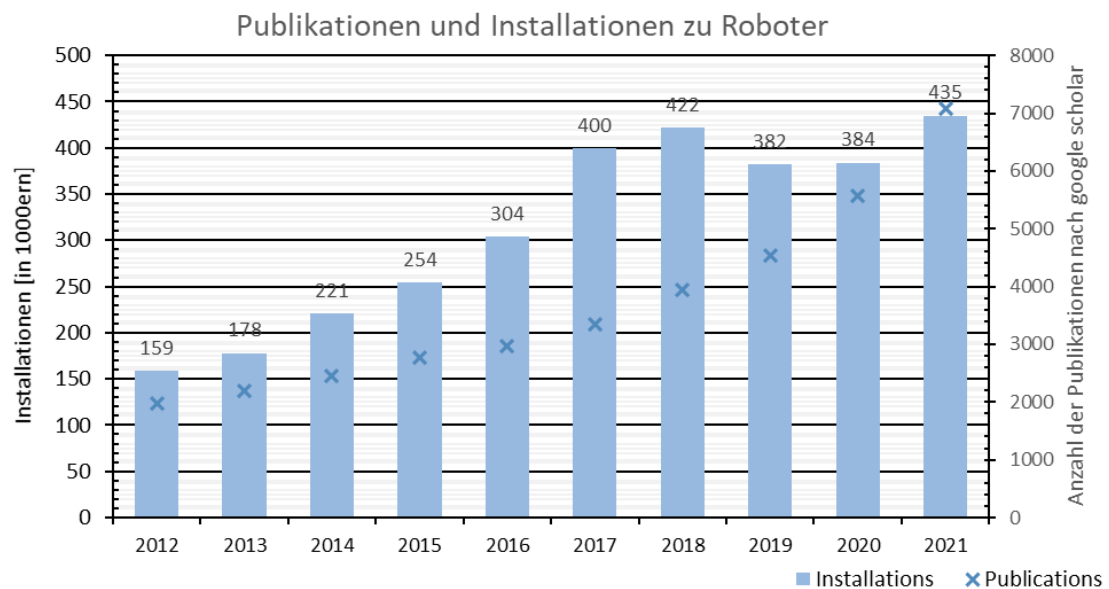


Abbildung 1: Publikationen und Installationen zu Robotern in Anlehnung an Dzedzickis et al. 2022

Mit dieser Abhandlung soll ein Beitrag geleistet werden, um das Forschungsgebiet dieser für die Menschheit höchst relevanten Technik zu erweitern. Das Thema Robotik hat auch nach all dieser Zeit noch viele Lücken in der Forschung, da ein ständiger technologischen Fortschritt in den unterschiedlichen Segmenten stattfindet. Die Abhandlung will durch einen angewandten Proof of Concept das Themenfeld der Robotik weiter untersuchen und neue Erkenntnisse ermöglichen.

2.1.1 Roboterarten

Aufgrund des sehr breit gefächerten Begriffs der technischen Maschine ‚Roboter‘ existiert eine Vielzahl an Ausgestaltungen, die von Roboterherstellern vertrieben werden (siehe Abbildung 2).



Abbildung 2: Roboterportfolio des Herstellers Fanuc¹

Zunächst kann in der Art der Ausgestaltung der Roboter zwischen seriellen und parallelen Robotern unterschieden werden. Bei den seriellen Robotern führt eine offene kinematische Kette von der Roboterbasis zum Endeffektor. Beispiele hierfür sind die typischen 6-Achs-Industrieroboter, Portalroboter, Palettierroboter oder SCARA-Roboter (Selective Compliance Assembly Robot Arm). Bei der Parallelkinematik führen mehrere kinetische Ketten zum Endeffektor, werden also parallel zueinander bewegt und bilden eine geschlossene kinematische Kette. Typische Vertreter dieser Roboterart sind Delta- oder Tripod-Roboter (Winkler 2020, 19 ff.).

Eine weitere geläufige Art ist der Serviceroboter, der zur direkten Unterstützung der Menschen eingesetzt wird und eine Abgrenzung zu den produzierenden Industriemanipulatoren darstellt. Heutige Servicerobotersysteme werden u. a. in den Bereichen Verteidigung, Rettung und Sicherheit sowie Landwirtschaft oder für Dienstleistungen eingesetzt. Ihr Erscheinungsbild und ihre Kinematik sind individuell gefasst und je nach Aufgabe anders orientiert, teilweise mit menschenähnlichen Zügen (humanoide Robotik) (Decker 2011). In Form von

¹ Vgl. https://automationspraxis.industrie.de/wp-content/uploads/F/A/FANUC_Roboter_Uebersicht_83262047-AE53-4ACD-A580-7FA65146FDD4.jpg (zuletzt aufgerufen am 25.02.2022)

Staubsaugrobotern haben solche Serviceroboter schon breiten Einzug in die Haushalte der Menschen gefunden (Forlizzi und DiSalvo 2006).

Weiterhin werden fahrerlose Transportsysteme oftmals als (autonome) mobile Roboter deklariert (Hertzberg et al. 2012). Maschinen können demnach eine Roboterart darstellen, ohne zwingend eine kinematische Kette zu besitzen. In diese Roboterart fallen auch Drohnen oder andere bewegbare Maschinen, deren Bewegungspfad frei programmierbar ist und automatisiert durchlaufen werden kann. Zusammenfassend wird ersichtlich, dass der Bereich Robotik ein äußerst breit gefächertes Forschungsgebiet ist, mit einer Vielzahl unterschiedlicher Maschinen, die alle unter dem gleichen Begriff subsumiert werden. Jedoch können der Aufbau, die Funktion und die Anwendungsgebiete bei jeder Roboterart stark divergieren.

2.1.2 Roboterarm

In dieser Forschungsarbeit steht der serielle 6-Achs-Industrieroboter im Vordergrund, da der Roboter des Mock-ups diese Kinematik aufweist und er der für die Industrie relevanteste Robotertyp ist (siehe Abbildung 3).

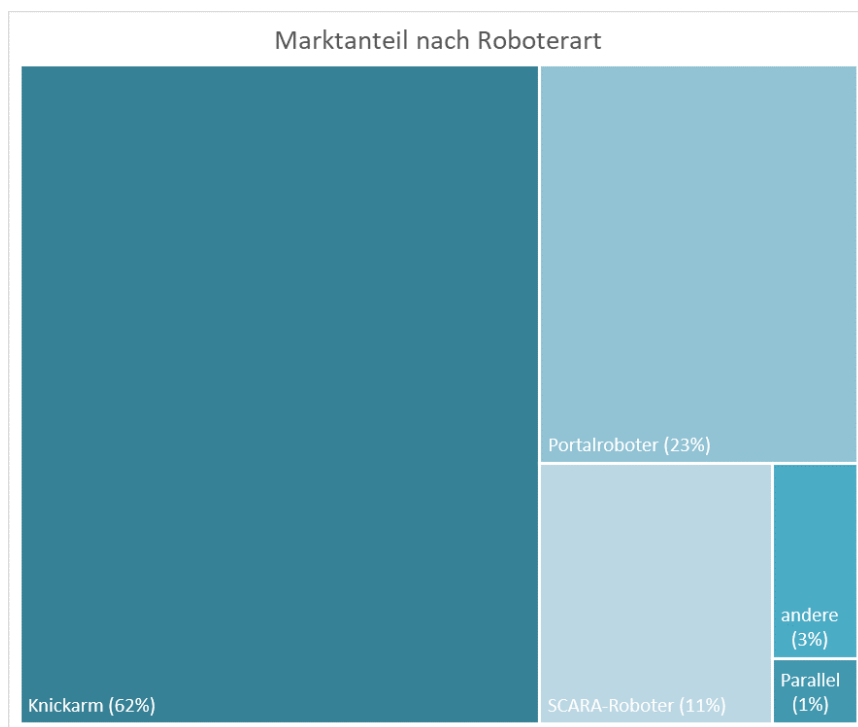


Abbildung 3: Marktanteil nach Roboterart nach Pott und Dietz (2019)

Im weiteren Verlauf der Arbeit wird die Bezeichnung ‚Roboter‘ oder ‚Industrieroboter‘ ausschließlich für diese Roboterart verwendet. Sie besteht aus sechs Gelenken, die z. B. mit Servomotoren betrieben werden. Die Gelenke haben einen Freiheitsgrad und sind mit starren Robotergliedern verbunden. In der realen technischen Umsetzung existiert hiervon eine Vielzahl von Abwandlungen für z. B. Spezialanwendungen oder bestimmte Arbeitsgebiete.

Die Steuerung des Roboters bzw. die Programmierung der Bewegungsabläufe erfolgt oft auf Basis von unterschiedlichen Koordinatensystemen. Oft wird der Endeffektor anhand eines Koordinatensystems gesteuert und verfahren. Es können auch direkt die Servomotoren der Achsen angesteuert werden, jedoch ist das für die Programmierung in der Regel nicht praktikabel. Jeder Roboterhersteller hat hierzu andere Konventionen, Bezeichnungen und Umsetzungen, jedoch werden häufig drei verschiedene Koordinatensysteme zur Orientierung des Roboters verwendet.

Das Weltkoordinatensystem ist das Koordinatensystem der Roboterumgebung. Dieses liegt oft auf dem nicht veränderlichen Roboter-Ursprungskoordinatensystem in der Basis des Roboters. Das Werkzeugkoordinatensystem befindet sich am Werkzeugende (auch TCP – Toolcenterpoint genannt) des Roboters, das Basiskoordinatensystem (oder auch Werkstückkoordinatensystem) bezieht sich auf ein externes Objekt (siehe Abbildung 4) (Zeichen und Fürst 2000, S. 124).

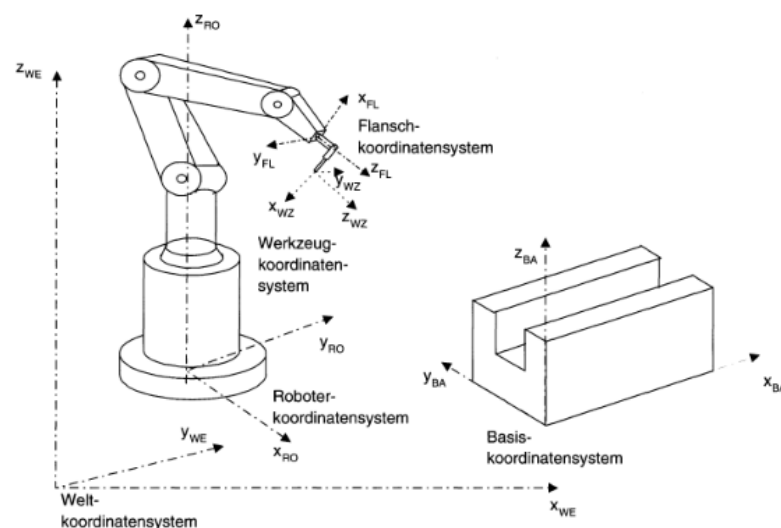


Abbildung 4: Koordinatensysteme von Industrierobotern nach Zeichen und Fürst (2000)

Mithilfe der drei Koordinatensystemarten werden die Roboterprogramme erstellt, indem die Bewegungen bzw. anzufahrenden Punktkoordinaten des Roboters immer auf eines der Bezugssysteme verweisen.

2.1.3 Kinematik

Die Kinematik des Roboters kann mit unterschiedlichen Ansätzen mathematisch dargestellt werden. Nach L. Euler kann die Raumkinematik als Rotation im dreidimensionalen euklidischen Raum als Abbildungsmatrix in Abhängigkeit dreier wesentlicher Parameter, der ‚EULER-Winkel‘, beschrieben werden. Der Orientierungsarbeitsraum wird erzeugt, indem der Roboter bzw. das Roboterglied aus seiner Ursprungslage heraus nacheinander um die Eulerwinkel in drei vorher definierten Koordinatenachsen gedreht wird (siehe Abbildung 5). Je nach Konvention kann die Drehung über unterschiedliche Achsen erfolgen und somit unterschiedliche Winkel erzeugen. Es existieren bei drei möglichen Drehachsen somit zwölf verschiedene Konfigurationen von Eulerwinkeln zur minimalen Darstellung der Orientierung (Husty et al. 1997, S. 335; Mareczek 2020b, S. 2–3).

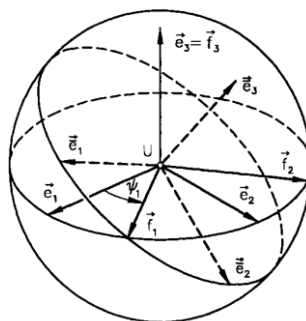


Abbildung 5: Koordinatensysteme und Achsdrehung für Eulerwinkel nach Husty (1997)

Eine weitere Konvention zur Beschreibung der Kinematik ist die homogene Transformationsmatrix. Die Translation und Rotation des Industrieroboters zur Berechnung der Stellung des Endeffektors müssen mehrere Male hintereinander angewandt werden, um von einem Startkoordinatensystem (oft das Welt- oder Roboterkoordinatensystem) über mehrere Zwischenkoordinatensysteme zu einem Werkzeugkoordinatensystem zu gelangen. Die lokalen Koordinatensysteme jedes Glieds zwischen Sockel und Endeffektor stellen die Zwischenkoordinatensysteme dar. Die Ketten von Transformationen können mit einer ge-

meinsamen Transformationsmatrix zusammengefasst werden (Mareczek 2020a, S. 55).

$$T_h = \begin{bmatrix} R_{3 \times 3} & \vec{p}_{3 \times 1} \\ \vec{f}_{1 \times 3} & w_{1 \times 1} \end{bmatrix}$$

Die homogene Transformationsmatrix (T_h) besteht aus der Rotationsmatrix ($R_{3 \times 3}$), einem Positionsvektor ($\vec{p}_{3 \times 1}$), einem Perspektivtransformationsvektor ($\vec{f}_{1 \times 3}$) und dem Skalierungsfaktor ($w_{1 \times 1}$) (Winkler 2020).

Sie ist jedoch nicht eindeutig, nicht kontinuierlich und sie enthält redundante Parameter in der Rotationsmatrix, weshalb im Jahr 1988 M. Horn in seiner Abhandlung eine alternative Darstellung der absoluten Orientierung mittels Quaternionen in der Robotik beschrieb. Mit Quaternionen sollen die Drehungen im dreidimensionalen Raum angemessen beschrieben werden, indem sie eine eindeutige Beschreibung der Orientierung im Raum mit geringer Redundanz und auch in Relation zu anderen Objekten ermöglichen (Horn et al. 1988). Der Quaternion Q wird wie folgt definiert und mithilfe komplexer Zahlen dargestellt:

$$Q = \epsilon_0 + \epsilon_1 i + \epsilon_2 j + \epsilon_3 k$$

wobei $\epsilon_0, \epsilon_1, \epsilon_2$ und ϵ_3 skalare Werte ($\in \mathbb{R}$) annehmen und i, j und k imaginäre Einheiten ($\in \mathbb{R}^3$) und Operatoren darstellen, die die folgenden Gleichungen erfüllen:

$$i^2 = j^2 = k^2 = -1$$

$$ij = k, \quad jk = i, \quad ki = j, \quad ji = -k, \quad kj = -i, \quad ik = -j$$

Mit diesen Gleichungen kann die Drehung im Raum beschrieben werden (Siciliano und Khatib 2008, S. 14).

2.1.4 Vorwärtstransformation

Die Vorwärtstransformation oder auch direkte Kinematik befasst sich mit dem Problem, wie mithilfe der Gelenkkoordinaten des Roboters die Position und Orientierung des Endeffektors ermittelt werden können (Winkler 2020).

Für serielle Roboter mit einer offenen kinematischen Kette stellt die Methode nach Denavit-Hartenberg eine allgemeingültige Konvention zur Berechnung der direkten Kinematik bereit. Aufgrund der starren Verbindung der einzelnen Roboterglieder lassen sich die einzelnen Koordinatensysteme der Gelenke durch Translation und Rotation ineinander überführen. Dabei hat jedes Manipulator-

segment aufgrund seiner Anordnung im Roboterarm ein körpereigenes Koordinatensystem, womit die Position und Orientierung des Endeffektors mittels Matrizenmultiplikationen der einzelnen Koordinatensysteme berechnet werden. Das Besondere der Methode ist, dass nur vier statt sechs Parameter erforderlich sind, um ein Bezugssystem relativ zu einem anderen zu bestimmen.

Um die Konvention praktisch durchzuführen, müssen die individuellen DH-Parameter (d_i : Gelenkabstand, θ_i : Gelenkwinkel, a_i : Armelementlänge, α_i : Verwindung) des Roboters abgemessen werden, und es sind vier Operationen erforderlich. Hieraus resultiert, dass eine Transformationsmatrix des Koordinatensystems S_{i-1} in das Koordinatensystem S_i zu berechnen ist. Es erfolgt eine Translation entlang der z_{i-1} -Achse um d_i und anschließend eine Drehung um die z_{i-1} -Achse um den Winkel θ_i . Danach folgt eine Verschiebung entlang der neuen x -Achse um a_i , um in einem letzten Schritt eine Rotation um die neue x -Achse um den Winkel α_i durchzuführen. (Denavit und Hartenberg 1955; Mareczek 2020a, 89 ff.; Siciliano und Khatib 2008, S. 23; Winkler 2020)

DH – Transformationsmatrix:

$$A_i = TRANS(z, d_i) * ROT(z, \theta_i) * TRANS(x, a_i) * ROT(x, \alpha_i)$$

Die Denavit-Hartenberg-Konvention wurde im Laufe der Zeit angepasst und abgewandelt, weshalb inzwischen zahlreiche ähnlich beschaffene Konventionen mit unterschiedlichen Koordinatensystemanordnungen oder Bezeichnungen existieren. Einschränkungen dieser Methode finden sich u. a. in kinematischen Ketten, die sich verzweigen, oder bei Gelenken mit mehr als einem Freiheitsgrad (Siciliano und Khatib 2008, S. 48).

2.1.5 Rückwärtstransformation

Die Rückwärtstransformation oder auch indirekte Kinematik befasst sich mit dem Problem, wie mithilfe der Position und Orientierung des Endeffektors die Gelenkkoordinaten des Roboters ermittelt werden können (Winkler 2020).

Moderne Manipulatoranwendungen benötigen hohe Genauigkeiten auf einzelnen Punkten oder entlang eines Pfads, die mit händischem Teachern nicht erreichbar sind. Dafür müssen die einzelnen Gelenkstellungen so berechnet werden, dass sich die genaue Endeffektorlage ergibt. Zur Berechnung der Gelenkstellungen existieren verschiedene Methoden, die oft auf nichtlinearen Gleichungssystemen beruhen und oft mehr als eine Lösung aufweisen. Mit numeri-

sche Lösungsverfahren lassen sich z. B. mit hoher Rechenleistung alle Arten von Kinematik berechnen. Die meisten 6-Achs-Roboterarme können jedoch auch mit schnelleren analytischen Lösungsverfahren berechnet werden. Treten unendlich viele Lösungen auf, wird von einer Singularität gesprochen (Marczek 2020a, 117 ff.). Das ist ein relevanter Bereich in der Roboterprogrammierung, da es dort zu hohen Winkelbeschleunigungen oder gar zu Fehlern kommen kann. Die vorgegebene Geschwindigkeit am Endeffektor (TCP-Geschwindigkeit) kann mit den mathematisch berechneten und dem Roboter möglichen Winkelgeschwindigkeiten der Robotergelenke (Joint-Geschwindigkeit) nicht erreicht werden. Der Roboter verliert hier einen Freiheitsgrad und ist in seiner Bewegung eingeschränkt, weshalb oft versucht wird, diese Bereiche in der Pfadplanung zu umgehen (Pott und Dietz 2019, S. 30).

2.2 Kollaborative Roboter (Cobots)

Der Begriff ‚Cobot‘ ist ein Kunstwort aus den Begriffen ‚kollaborativ‘ und ‚Roboter‘ und steht für die Zusammenarbeit eines Roboterarmes mit dem Menschen in einem Arbeitsbereich (Colgate et al. 1996). Roboter sind nicht mehr automatisch durch Schutzzäune, Laserscanner oder Begrenzungen direkt vom Menschen getrennt. Es gibt mit Cobots auch eine direkte Interaktion zwischen Menschen und Robotern (Linda Onnasch et al. 2016).

Roboter integrieren sich auch abseits der direkten Produktion immer mehr in die Lebens- und Arbeitswelt des Menschen. In der industriellen Fertigung längst unentbehrlich, erledigen die modernen Robotersysteme zunehmend auch Aufgaben in anderen Lebensbereichen und Anwendungsfeldern (Haun 2013, S. 6).

Die deutsche Normung definiert einen ‚Cobot‘ als:

„Roboter, der für das direkte Zusammenwirken mit dem Menschen innerhalb eines festgelegten Kollaborationsraums konstruiert ist.“

(DIN EN ISO 10218-2:2011)

Cobots sind eine neue Entwicklung, um die Produktion und wirtschaftliche Nutzbarkeit zu optimieren. Der Roboter soll in die Lage versetzt werden, seine Arbeitsumgebung zu überwachen, sicher mit Menschen und der Umgebung zu

interagieren, neue Fähigkeiten und Aufgaben zu erlernen und sein Verhalten an den jeweiligen Kontext anzupassen. Besonders im Paradigma der Industrie 4.0 ergänzt also der Roboter die Fähigkeiten des Menschen. Er lernt neue Aufgaben und passt sich selbst an, um Unsicherheiten, Planungsfehler und unvorhersehbare Ereignisse auszugleichen. Vor allem kollaborierende Roboter können viele Vorteile für die Produktionsumgebung mit sich bringen, da sie Sicherheitsrisiken verringern und flexible Lösungen in Bezug auf Nutzung und Programmierung bereitstellen (Roveda et al. 2021).

Es existieren verschiedene Ansätze, die Fähigkeiten von Menschen und Maschinen zu kombinieren. In unterschiedlichen Anwendungsszenarien wird untersucht, wie Effizienz und Produktivität gesteigert werden können. In der Arbeitsverteilung kann der Roboter diverse Tätigkeiten übernehmen und auf unterschiedliche Weise mit dem Menschen interagieren (Buxbaum 2020, 96 ff.).

Ein gängiges Modell ist das der Kollaboration, Kooperation und Koexistenz zwischen Roboter und Mensch (siehe Abbildung 6).

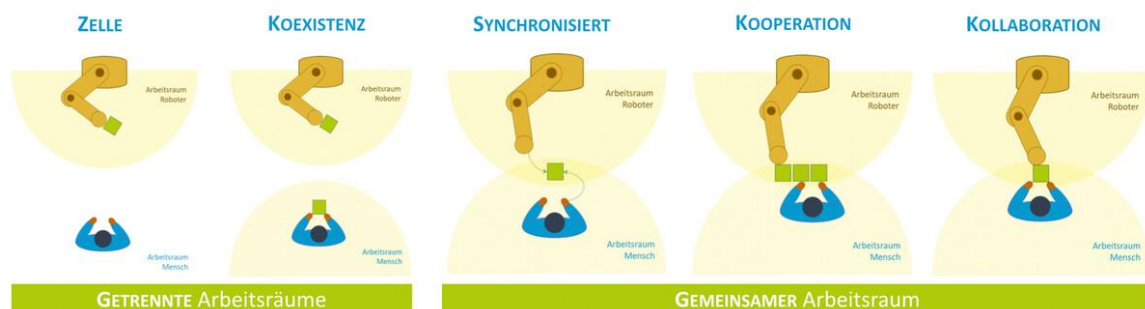


Abbildung 6: Mensch-Roboter-Kollaboration nach dem Institut Chemnitzer Maschinen- und Anlagenbau e. V.

Der Status quo in der industriellen Fertigung ist heutzutage die Produktion eines Roboters in einer separaten Zelle. Hin zur kompletten Kollaboration, in der Mensch und Roboter zusammen in einem Arbeitsbereich am selben Werkstück arbeiten, existiert eine stufenweise Taxonomie. Die Koexistenz beschreibt eine zeitlich und räumlich sehr begrenzte Interaktion ohne gemeinsamen Arbeitsraum. In dieser Interaktion gibt es keine gemeinsame Zielstellung, an der Mensch und Roboter arbeiten. In der synchronisierten Interaktion haben Mensch und Roboter einen sich überschneidenden Arbeitsraum, in diesem jedoch keine zeitgleichen Bearbeitungen. Bei der Kooperation überschneidet sich der Arbeitsraum, in dem parallel gearbeitet wird. Die Handlungen sind nicht

unmittelbar voneinander abhängig, da es eine klare Aufgabenteilung zwischen Menschen und Robotern gibt (Linda Onnasch et al. 2016).

Jedoch ist die konzeptionell beschriebene Nutzung von Cobots bislang weit entfernt von der Realität. Es existieren noch viele Hürden, die die Potenziale stark einschränken. Nach einer Umfrage in der finnischen Industrie sind besonders das fehlende Wissen um Anwendungsfälle, als auch die Fragen zur Sicherheit die Haupthemmnisse für den Einsatz im Unternehmen (Aaltonen und Salmi 2019).

Der erste Hersteller, der Cobots verkaufte, ist der dänische Hersteller ‚Universal Robot‘, der 2005 gegründet wurde. Nach drei Jahren Entwicklungsphase wurde 2008 der erste marktreife kollaborative Roboter verkauft. (Universal Robots) Inzwischen gibt es eine Vielzahl an Akteuren im Cobot-Markt, von weiteren Start-ups, die sich auf diese Marktnische spezialisiert haben, bis zu etablierten Roboterherstellern, die über die horizontale Produktdiversifikation Cobots in ihr Portfolio aufgenommen haben. Inzwischen existiert eine hohe Produktvielfalt im Bereich der Cobots, von klassischen 6-Achs-Roboterarmen mit Kraft-Momenten-Sensoren, über 7-Gelenk-Roboter (LBR iiwa von Kuka), 2-Arm Cobots mit 14-Achsen (YuMi IRB von ABB), Cobots mit integrierten Kamerasystemen und Objekterkennung (Sawyer von Rethink Robotics), bis hin zu Cobots mit hoher Traglast (CR-35iA von Fanuc). Insgesamt hat sich das Produkt des Cobots im Robotikmarkt etabliert und ist ein Treiber für Innovationen im Bereich der Robotik. Es weist historisch einen stetig steigenden Umsatz und Marktanteil aus (Ausnahme: 2020 aufgrund der Corona-Pandemie) und es wird mit einem weiteren Umsatzwachstum in den kommenden Jahren mit jährlich über 15 % gerechnet, sodass das Robotiksegment auch weiterhin relevant bleiben wird (Xiao 2021).

2.3 Robotersteuerung

Industrieroboter können auf unterschiedliche Arten gesteuert werden. Nahezu jeder Roboter besitzt eine eigene Robotersteuerung und eine eigene Programmiersprache, die die Programmierung des Roboters ermöglicht. Es können auf der Feldebene Aktoren und Sensoren angeschlossen werden, deren Wirkung in den Eingangssignalen an die Steuerung zurückgemeldet wird (Plenk 2019, 16 ff.). Die Robotersteuerung kann in der Regel auch mit einer externen speicher-

programmierbaren Steuerung (SPS), die die Steuerung übernimmt, über die gängigen Bussysteme verbunden werden. Typische Arten der Programmierung sind in der nachfolgenden Grafik dargestellt (siehe Abbildung 7) (Weber 2017).

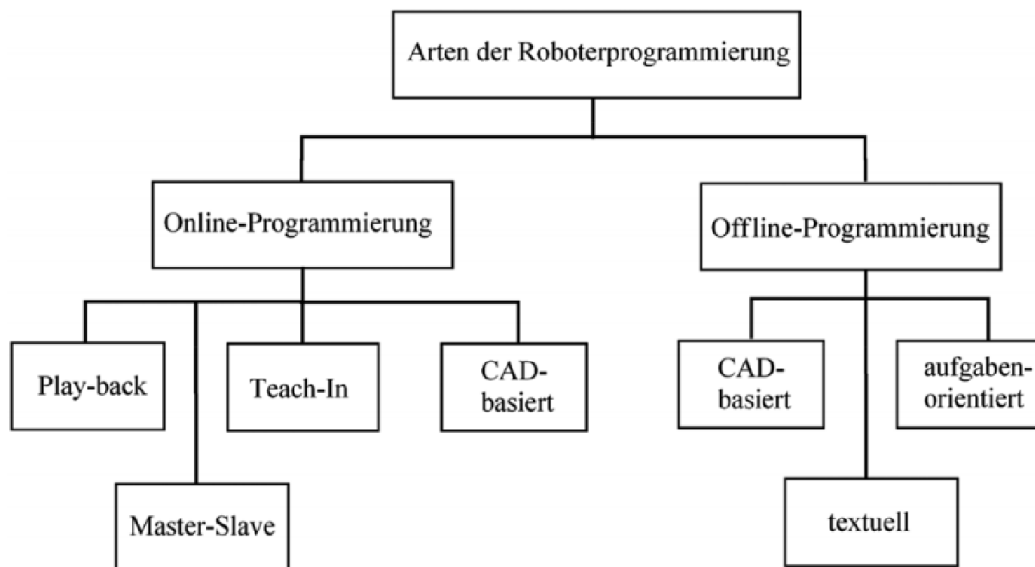


Abbildung 7: Arten der Roboterprogrammierung nach Weber (2017)

Unter ‚Offline-Programmierung‘ wird textuelles Programmieren bzw. das Programmieren eines Roboters in einer Simulationsumgebung verstanden. Dabei gibt es keine direkte Verbindung zum Roboter und es wird auch keine Interaktion in der realen Umgebung erzeugt. Hierbei wird ein Roboterprogramm unabhängig von der eigentlichen Roboteranlage erstellt. Die Onlineprogrammierung hingegen beschreibt die Programmierung direkt am realen Roboter. Der typische Life-Cycle einer Roboterinbetriebnahme ist die CAD-basierte Offlineprogrammierung, um an einem digitalen Zwilling den Arbeitsprozess des Roboters im Vorfeld zu simulieren. Anschließend können Feinheiten mit dem Teach-in-Verfahren in der realen Arbeitsumgebung angepasst werden. Für das Teach-in hat jeder Roboter ein Teach-Pendant (auch Handprogrammiergerät oder Programmierhandgerät genannt), das zur Onlineprogrammierung vor Ort dient (Pott und Dietz 2019, S. 76).

Jeder Hersteller hat andere Handprogrammiergeräte die teilweise Ähnlichkeiten, aber auch deutliche Unterschiedlichkeiten aufweisen. Jeder Hersteller setzt das Steuerungskonzept des Roboters mit gewissen Eigenheiten und einer eigenen Mensch-Maschine-Schnittstelle (siehe Kapitel 2.4.1) um. In der Abbildung 8 sind Handprogrammiergeräte oder alternative Lösungen von unterschiedlichen Herstellern dargestellt.

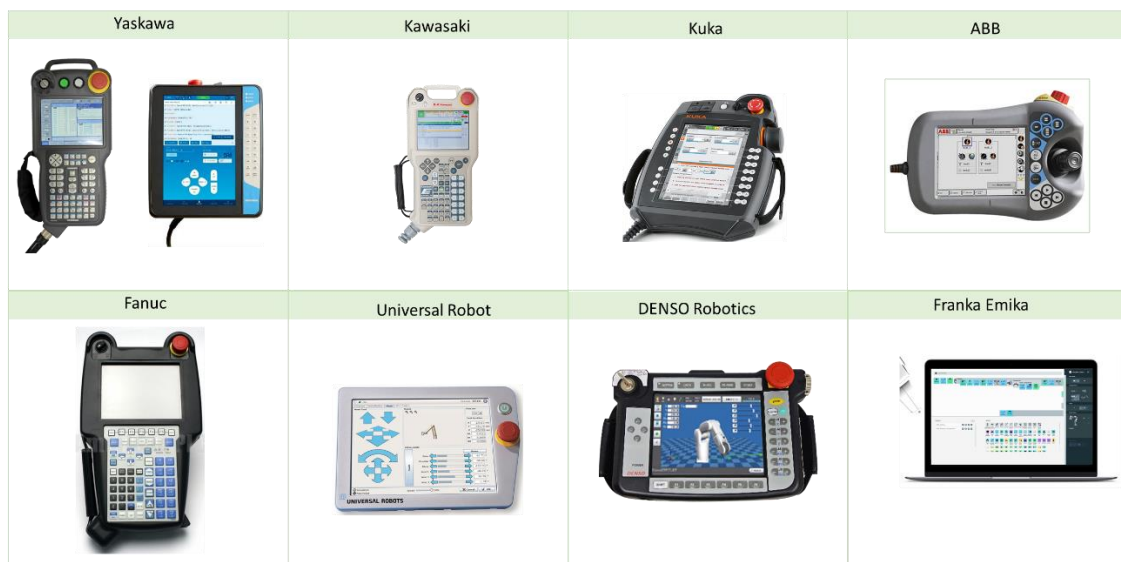


Abbildung 8: Handprogrammiergeräte oder Steuerungen unterschiedlicher Roboterhersteller nach eigener Recherche

Es wird ersichtlich, dass jeder Hersteller einen anderen Ansatz verfolgt, um die Steuerung des Roboters zu ermöglichen. Es kann angenommen werden, dass keine optimale Steuerung existiert, sondern unterschiedliche Ansätze mit Vor- und Nachteilen. Um die Achsen anzusteuern oder die Translation und Rotation der 6 Achsen durchzuführen, existieren verschiedene Konzepte. Yaskawa hat rechts und links jeweils sechs Tasten, um eine positive und negative Veränderung der 6 Freiheitsgrade per Knopfdruck zu ermöglichen. Kawasaki, DENSO, Kuka und Fanuc nutzen hierfür jeweils auf der linken Seite 12 Knöpfe. ABB hat einen Joystick mit drei Freiheitsgraden, über die Bewegung in x- und y-Achse sowie eine Drehbewegung können drei Freiheitsgrade angesteuert werden, wobei es einen weiteren Modus gibt, um die weiteren drei Freiheitsgrade des Roboters zu nutzen. Es können jedoch nie gleichzeitig alle sechs Achsen oder alle Freiheitsgrade der Translation und Rotation gesteuert werden. Das Handprogrammiergerät von Kuka hat eine 6D-Maus integriert, die genauso viele Freiheitsgrade, wie der Roboter besitzt. Somit können mit einem Controlstick am Handprogrammiergerät alle Freiheitsgrade des Roboters in einer kompakten Art und Weise gesteuert werden. Neuere Hersteller wie Universal Robots und Franka Emika setzen auf Touchscreen und wenig haptische Knöpfe und Tasten. Bei dem Universal Robots Teach-Pendant gibt es unterschiedliche digitale Knöpfe an unterschiedlichen Stellen, die zusammen mit einer graphischen Visualisierung genutzt werden können, um den Roboter zu steuern. Franka Emika verzichtet sogar ganz auf das Handprogrammiergerät und ersetzt dieses

durch einen Computer und eine dazugehörige Software, um die 7 Achsen des Cobots zu steuern.

Neben den klassischen Formen der Programmierung existiert eine Vielzahl von experimentellen Methoden, um einen Roboter zu steuern. Die Bandbreite reicht von wissenschaftlichen Mock-ups über innovative Produkte von Roboterherstellern bis hin zu Start-up-Ideen und speziellen Branchen Anwendungen. Sowohl softwareseitig als auch die Hardware betreffend gibt es viele Entwicklungen, um die Roboterprogrammierung und -steuerung zu unterstützen. Oft liegt der Beweggrund darin, die Programmierung einfacher zu gestalten, da es an Fachkräften zur Roboterprogrammierung mangelt. Jeder Roboterhersteller hat eine eigene Programmiersprache und es bedarf einer tiefgreifenden Schulung und Weiterbildung für die individuellen Systeme. Viele experimentelle Geräte zur Steuerung haben z. B. einen Joystick, eine 3-D- oder eine 6-D-Maus inkludiert, um eine intuitivere Steuerung zu ermöglichen.

Das Forschungsthema, einen Roboter mit einem Controller oder Joystick zu steuern, wurde bereits einige Male untersucht. So wurde 2016 in einem Forschungsbeitrag eine Software entwickelt, die Eingaben eines handelsüblichen Gamepad-Geräts liest, auswertet und an die Robotersteuerung weiterleitet, die die Bewegung eines Industrieroboters steuert. Vergleichsversuche zeigten eine schnellere Bearbeitung von Handlingaufgaben mit der intuitiveren und vielbekannten Gamepad-Steuerung (Wagner et al. 2016). Eine vergleichbare Studie kam 2019 zu einem ähnlichen Schluss (Rahman et al. 2019).

Hingegen ist die Forschung bezüglich des Vorhabens, in den Controller zur Robotersteuerung ein haptisches Feedback zu implementieren, noch lückenhaft. Es gab erst einige erste Versuche in diese Richtung. So begann das DLR (Deutsches Zentrum für Luft- und Raumfahrt) schon 2003 ein Projekt zum USB-Force-Feedback-Joystick. Der kraftreflektierende Joystick konnte durch die gemessenen Kontaktkräfte von Leichtbaurobotern dem Nutzer ein haptisches Feedback zu den Qualitätsunterschieden verschiedener Oberflächenbeschaffenheiten in einer Weltraumumgebung geben (DLR Institut für Robotik und Mechatronik). Weiterentwickelt wurde dieses Projekt 2011 mit dem raumfahrttauglichen Joystick (RJo), der einen Force-Feedback-Joystick mit zwei Freiheitsgraden enthält und den Astronauten auf der International Space Station (ISS) erlaubt, einen Roboter sensitiv auf der Erde zu steuern. Im Jahr 2015 wurden hierzu erste Experimente realisiert und es wurde die Telemanipulation des Ro-

boters mit haptischem Feedback ermöglicht. Der Forschungsschwerpunkt lag auf der weltraumtauglichen Gestaltung und Kommunikation (Riecke et al. 2016).

Um das Themengebiet für die Forschung zu erweitern, wird im Rahmen der vorliegenden Arbeit der Bereich mit einem eigenen praktischen Mock-up bearbeitet. Eine Betrachtung möglicher kommerzieller Lösungen zur Robotersteuerung dient als Grundlage zur Ideenfindung des Mock-ups und wird in Kapitel 3.2 vorgestellt.

In den Entwicklungen der Robotersteuerungen kann ein Trend in Richtung der Fokussierung auf den Menschen ausgemacht werden. Intuitive Mensch-Roboter-Interaktion war früher ein Thema, das sich ausschließlich auf die Servicerobotik bezog, aber heute findet es Einzug in viele Anwendungen, auch dort, wo klassische Formen der Robotersteuerung dominieren. Unabhängig von der Komplexität des Systems findet ein ständiger Informationsaustausch zwischen der Umgebung, dem Roboter und dem Menschen statt. Die Anweisungstypen, die ein Mensch erteilen kann, lassen sich grob in die klassische Programmierung, Sprachbefehle, visuelle Gestik, Berührung und die Verwendung aller Arten von Tasten bzw. Eingabegeräten unterteilen (Šekoranja et al. 2014). Auf der anderen Seite kann der Mensch aber auch seine Umgebung wahrnehmen. Diese kann mit den Modalitäten visuell, auditiv, olfaktorisch, gustatorisch und taktil beschrieben werden. Neben diesen fünf Sinnen können Druck, Kraft und Vibration durch die Haptik erkannt werden. Während der Mensch seine Sinne nutzen kann, muss ein Robotersystem durch Sensoren ergänzt werden. Diese Sinneseindrücke können in Eingabe- und Ausgabekanäle unterschieden werden, für die verschiedene Interaktionstechnologien existieren. Zu den Eingabetechnologien gehören Tastaturen, akustische, optische und haptische Komponenten sowie Eingabesysteme für virtuelle Umgebungen. Die Ausgabe von Informationen wird meist über Monitore oder audiovisuelle Signale realisiert. Interaktionstechnologien, die Eingabe und Ausgabe verbinden, sind z. B. Touchscreens, die in einigen Teach-Pendants zur Robotersteuerung heute schon implementiert sind. Fortgeschrittene Eingabe- und Programmierstechniken sind die Gesten- und die Sprachsteuerung, die in der Robotertechnik aktuell erforscht werden (Berg et al. 2019; Norberto Pires 2005). Außerdem bieten neue Ansätze eine berührungsbasierte Bewegung des Roboters, sodass ein Bediener den Roboter einfach mit der Hand in jede beliebige Richtung führen

kann. Jeder Ansatz bietet Vor- und Nachteile und kann als Grundlage dienen, um die Roboterprogrammierung zu vereinfachen (Šekoranja et al. 2014).

2.3.1 Feedback

Feedback ist eines der bedeutendsten Elemente des menschlichen Lernens und bietet somit ein hohes Potenzial bei berührungsbezogenen Mensch-Maschine-Schnittstellen. Von den in Kapitel 2.3 vorgestellten Arten der Interaktion sind visuelles, auditives und haptisches Feedback besonders relevant in der Robotik (Gao et al. 2021, S. 91).

Nach Loomis und Lederman gilt haptisches Feedback als eines der relevantesten Elemente für den Menschen, um etwas über seine Umgebung zu erfahren (Loomis und Lederman 1986). Diese Rückmeldung ist bei Manipulations- und Entdeckungsaufgaben des täglichen Lebens von entscheidender Bedeutung (Martínez et al. 2011). Das haptische Feedback in der Mensch-Maschine-Interaktion umfasst Elemente wie Vibrationen, Strukturveränderung, Scherung und elektrisches Feedback. Diese Elemente können nicht nur kontextbezogene, sondern auch klare, präzise und weniger störende Informationen übermitteln (Gao et al. 2021, S. 91). Dies gibt Grund zur Annahme, dass haptisches Feedback auch in der Robotik nutzbringend eingesetzt werden kann.

Das vibrotaktile Feedback scheint eine effektive Methode zu sein, um zwischen Texturmustern und verschiedenen Gegenständen zu unterscheiden, die durch die Frequenzänderungen ihrer Oberflächenmerkmale beim Reiben mit dem Finger identifiziert werden können (Martínez et al. 2011). Übertragen auf die Robotik kann der Endeffektor des Roboters mit eben jenen Fähigkeiten ausgestattet werden.

Die Umsetzung haptischen Feedbacks wurde schon in diversen Anwendungsgebieten wissenschaftlich untersucht, bis hin zur Implementierung in marktreife Produkte. Vibrationen wurden in der Praxis bereits erfolgreich eingesetzt, um Navigationsinformationen und Feedback zur Beschleunigung bereitzustellen. In einem Forschungsprojekt wurden beispielsweise Navigationsinformationen durch einen vibrierenden Hüftgurt übermittelt. Die Ergebnisse zeigen, dass vibrotaktile Hinweise sowohl für die Navigation als auch für das Situationsbewusstsein in Umgebungen mit mehreren Aufgaben nützlich sind (Pielot und Boll 2010).

Lieberman und Breazeal stellten einen Anzug für Arm und Oberkörper vor, der die Bewegung des Trägers durch vibrotaktilen Feedback verbessern bzw. steuern kann. Der Anzug besteht aus acht über den rechten Arm verteilten vibrotaktilen Aktoren, deren Frequenz und Amplitude unabhängig voneinander gesteuert werden. Auf diese Weise wird dem Nutzer ein Feedback zur aktuellen Position bzw. Stellung gegeben (Lieberman und Breazeal 2007).

Ein vibrierendes haptisches Armband für die Mensch-Roboter-Interaktion wurde in einem Forschungsbeitrag im Jahr 2012 getestet. Das Armband besteht aus drei vibrierenden Motoren, die den Benutzer mit Informationen über die Bewegungen der folgenden fahrerlosen Roboter versorgen (Scheggi et al. 2012).

Viele Anwendungen und Forschungsuntersuchungen befassten sich mit der robotergestützten Medizin und dem taktilen Feedback für Chirurgen und Mediziner. Kraft- bzw. Berührungsinformationen des Endeffektors an die Person, die den Roboter bedient, zu übertragen, wäre eine Möglichkeit, neue Potenziale bei bearbeitenden oder medizinischen Robotern zu eröffnen. Diese Signale können dem Bediener helfen, während der Operation sofortige Korrekturen vorzunehmen, z. B. das Schneiden, Trennen, Handhaben und Bewegen von Gewebe, Gefäßklemmungen, das Binden eines Knotens, das Erkennen der Gewebeart oder auch von Werkzeugkollisionen (Nawrat et al. 2016).

Unterschiedliche Studien konnten zeigen, dass vibrotaktile Stimulation ein brauchbarer Ersatz in der minimalinvasiven Chirurgie ist und die Leistung der Chirurgen verbessern kann. Das Feedback optimiert die Fähigkeit, ausgeübte Kräfte auf Gewebe zu kontrollieren und dessen Weichheit bei einer simulierten Gewebeuntersuchung zu unterscheiden (Okamura 2009; Schoonmaker und Cao 2007).

Auch das berühmte Robotersystem ‚da Vinci‘ für die angewandte Chirurgie wurde dahingehend untersucht. So wurde ein Sensor- und Betätigungssystem für das robotergestützte Chirurgiesystem mit einer vibrotaktilen Rückmeldung der Werkzeugkontaktbeschleunigungen entwickelt. In einer Studie äußerten die Chirurgen eine deutliche Präferenz für die Einbeziehung der vibrotaktilen Rückmeldung (Pacchierotti 2015, S. 10; McMahan et al. 2011). Das DLR entwickelte den oben beschriebenen Controller mit haptischer Rückkopplung für die minimalinvasive robotergestützte Chirurgie (MIRC) weiter, um einen Anwendungsfall zu der Technologie zu testen (Seibold 2013).

In Industrieanwendungen wird die Nutzung haptischen Feedbacks ebenfalls untersucht. So sollen haptische Rückmeldungen zu Griffkraft, Endeffektorkontakt und -druck sowie Beschleunigung die menschliche Leistung bei einer teleoperierten Pick-and-Place-Aufgabe verbessern. Untersucht wurde dies in einer Studie mit der Steuerung eines humanoiden PR2-Roboters mittels haptischen Geräts, das an der rechten Hand der Versuchsperson angebracht wurde. (Khurshid et al. 2017).

Bei der Bearbeitung von Werkstücken mit geringem Materialabtrag ging eine Studie zur Roboterbearbeitung von einer sensorgestützten, menschenähnlichen Bedienung aus, die eine verbesserte Ausführungsqualität erzielte. Das wurde durch Feedback und die somit enge Zusammenarbeit von Mensch und Industrieroboter mittels Sensoren und Softwaremodulen ermöglicht. Eine weiterführende Idee war Feedback auf der Basis von Krafterückkopplung beim Entgraten harten Materials, um auch dort Verbesserungen zu erzielen (Ji und Wang 2019).

2.4 Human Factors

Der Mensch nimmt eine immer dominantere Stellung in der Betrachtung neuer Fertigungstechnologien ein. Aus diesem Grund müssen Human Factors, also psychische, kognitive und soziale Einflussfaktoren, in Mensch-Maschine-Systemen betrachtet werden. Der Schwerpunkt der Human-Factors-Forschung liegt auf der Gestaltung der menschlichen Schnittstelle, der sogenannten Mensch-Maschine-Schnittstelle, und auf den Interaktionen zwischen dem (Roboter-)System und dem Bediener. Ziel von Human Factors ist es, die Bediener zu befähigen, ihre Aufgaben effektiv in einer positiven Umgebung zu erfüllen und damit einen positiven Beitrag zur Wertschöpfung zu leisten. Die Grundlage dafür ist die Minimierung von Bedienerfehlern durch ein sinnvoll gestaltetes Interface (Love 2007).

2.4.1 Mensch-Maschine-Schnittstelle

Die Interaktion zwischen Menschen und Robotern wird meist durch Benutzerschnittstellen realisiert. Verschiedene Interaktionsprinzipien werden in der DIN EN ISO 9241-110 definiert und es werden Gestaltungsempfehlungen für die Interaktion von Menschen mit Maschinen gegeben (DIN EN ISO 9241-110:2020-10). Hierbei wird der Informationsaustausch zwischen Menschen und Maschinen beschrieben. Mögliche Medien für die Interaktion mit Robotern sind u. a. visuelle Anzeigen, Hand- und Gesichtsgesten, Sprache, Audiosignale für Warnungen und physische Interaktion sowie Haptik. Mit der Entwicklung von Cobots erweitert sich die Interaktionstechnologie, die bei früheren Industrierobotern stark begrenzt war. Mit der Zunahme der Mensch-Roboter-Zusammenarbeit nimmt auch die Bedeutung der Mensch-Roboter-Interaktion zu. Aktuelle relevante Forschungsgebiete sind hier die Gesten-, Sprach- und Gesichtserkennung und -steuerung (Berg et al. 2019).

Beispielsweise entwickelte und evaluierte Pires (2005) die Spracherkennung in einem industriellen Kontext, sowohl bei direkter Zusammenarbeit als auch über die Fernsteuerung von Robotiksystemen (Norberto Pires 2005). Einige Konzepte wurden bereits in Kapitel 2.3 vorgestellt, da die Art der Robotersteuerung eine typische Mensch-Maschine-Schnittstelle darstellt. Jedoch inkludiert die Schnittstelle nicht nur die reine Steuerung, sondern befasst sich mit dem gesamten Arbeitsplatz des Roboterbedieners. Hierzu zählen alle Signale, Displays, Steuergeräte und Informationen, die zwischen Maschine und Bediener ausgetauscht werden, um eine effektive Arbeit zu ermöglichen. Menschen sind eine potenzielle Fehlerquelle, in vielen Fertigungsprozessen aber unentbehrlich. Daraus ergibt sich die Herausforderung, die Mensch-Maschine-Schnittstellen so zu konzipieren, dass Fehler vermieden werden. Die Anwenderprogramme und Steuerungselemente müssen auf eine Weise gestaltet sein, dass sie potenziell schwerwiegende Fehlbedienungen oder Missgeschicke auffangen können. Hierzu muss die Psyche des Menschen berücksichtigt werden. Der Bediener muss die Aufgaben wahrnehmen, indem er sie mit Aufmerksamkeit und Erkenntnis für die Tätigkeit ausführt. Zudem wird Kognition benötigt, die eine Situation schafft, in der der Arbeitsprozess interpretiert und bewertet werden kann. Das kann dadurch erreicht werden, dass alle relevanten Informationen leicht verfügbar und sinnvoll dargestellt werden. Weiterhin sollen die Mensch-

Maschine-Schnittstellen so umgesetzt sein, dass Aktionen einfach und verständlich durchgeführt werden können. Die Aufgaben und ihre Auswirkungen sollen aus der Benutzeroberfläche ersichtlich werden und intuitiv durchzuführen sein (Love 2007).

2.4.2 Gamification

Ende des Jahres 2010 etablierte sich der Begriff ‚Gamification‘ und gewann seitdem zunehmend an Bedeutung in Forschung und Praxis. In der Wissenschaft wird das Konzept der Gamification seit 2011 untersucht und hat eine immer größer werdende Relevanz.

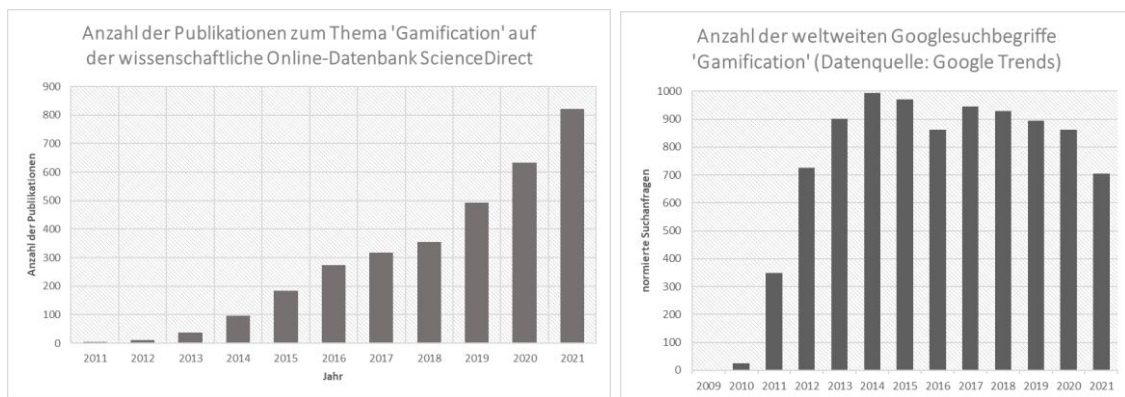


Abbildung 9: Relevanz von Gamification nach eigener Darstellung

Das Konzept der Gamification ist, Spiel-Design-Elemente aus Computer- oder Konsolenspielen in einem spielfremden Kontext einzusetzen. Ähnlich beschaffene Konzepte gab es schon lange vor diesem Begriff, z. B. Treuepunkte, Bonusmeilen im Vielfliegerprogramm oder Noten und Abschlüsse im Schulsystem. Inzwischen werden jedoch gezielt die Spielelemente und -techniken analysiert, die psychologische Effekte auslösen, sodass die Spieler motiviert und konzentriert bleiben. Die Spielelemente werden dann im Marketing, in Geschäftsmodellen oder in der täglichen Arbeit eingesetzt, um diese Effekte in Bezug auf Kunden und Angestellte zu nutzen (Sailer 2016, 6 ff.).

Die aktuellen Erkenntnisse zur Gamification durch die Forschung sind nach der systematischen Übersichtsstudie von Krath et al. 2021 wie folgt zusammengefasst. Gamification kann die Relevanz von Zielen und Errungenschaften transparent darstellen. Nutzer werden durch geführte Pfade zu zielgerichteten, überschaubaren und einfachen Aktivitäten bzw. Aufgaben geführt und erhalten un-

mittelbares Feedback. Gute Leistungen können durch geeignete Gamification-Methoden positiv verstärkt werden. In vielen Gamification-Mechaniken wird es den Nutzern ermöglicht, individuelle Ziele zu verfolgen und zwischen verschiedenen Fortschrittspfaden zu wählen, während die gamifizierten Systeme Aufgaben und Komplexität an die Fähigkeiten des Nutzers anpassen können. Das wirkt motivationsfördernd, da der Arbeiter seine Aufgaben autonom wählen kann und eine größere Entscheidungsbefugnis im Rahmen seiner Tätigkeiten erhält. Soziale Gamification-Elemente können soziale Vergleiche ermöglichen und die Nutzer miteinander verbinden, damit sie sich gegenseitig unterstützen und auf ein gemeinsames Ziel hinarbeiten können (Krath et al. 2021).

Für den Technologiedemonstrator soll das Game-Element des Controllers zur Steuerung eines Roboters verwendet werden. Ziel ist es, ein schon bekanntes Element zu nutzen, mit dem viele Menschen vertraut sind, um mentale Hürden zu überwinden. Das soll fachfremden Personen einen einfachen Einstieg und erfahrenen Roboterprogrammierern einen einfachen Umstieg in die Robotik ermöglichen.

3 Methodik und Umsetzung

In diesem Kapitel werden die Umsetzung des technischen Mock-ups und die hierzu grundlegenden Technologien beschrieben. Als Mittelpunkt der Arbeit wird die Auswahl der Teilkomponenten erläutert und die Entscheidung begründet. Weiterhin werden Alternativen aufgezeigt, um ein transparentes und wissenschaftliches Gesamtbild zu gewährleisten.

3.1 Technischer Mock-up

Zu Beginn des Projekts wurde eine Grobplanung erstellt und eine Machbarkeitsprüfung vorgenommen. Es erfolgte die Festlegung auf einen Roboter und einen Controller; weiterhin wurden die Schnittstellen und Möglichkeiten des Roboters betrachtet. In Abbildung 10 ist die theoretische Überlegung zur Umsetzung schematisch dargestellt. Diese wurde nach der Machbarkeitsstudie erstellt und dient als konkreter Umsetzungsplan für das Mock-up.

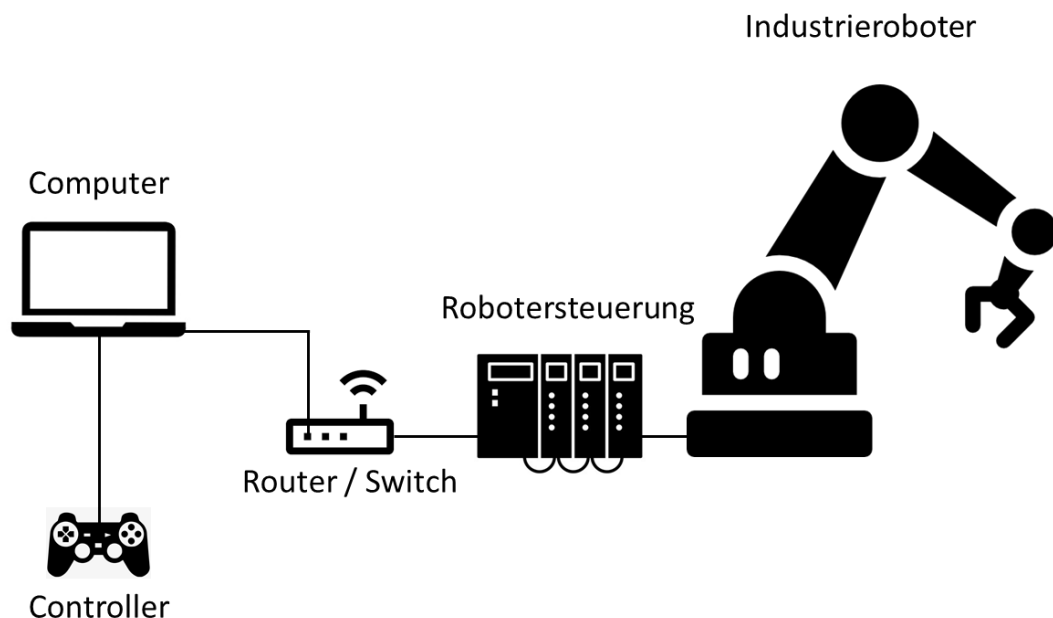


Abbildung 10: Skizze des Mock-ups nach eigener Darstellung

Nun wurde eine Umgebung gesucht, um die externe Software zu erstellen, die die Funktionalitäten und GUI ermöglicht. Hier wurde Python ausgewählt und es

wurden unterschiedliche Module und Vorgehen eruiert den Roboter von Universal Robot anzusteuern und auszulesen.

Die benötigten Programmteile und Funktionalitäten wurden in einzelne Module zerlegt und als Funktionen separat entwickelt und getestet. Die Einzelteile waren die Kommunikation zum Roboter, das Auslesen der Eingabewerte des Controllers, das Steuern des Roboters und das haptische Feedback. Final wurde noch eine einfache Benutzeroberfläche (Graphical User Interface - GUI), um Prozessdaten für den Bediener zu visualisieren.

Um die Funktionen zu testen auch ohne sich mit dem Roboter direkt zu verbinden, wurde alles in der Offlineprogrammierung an einem digitalen Zwilling getestet. Universal Robot stellt für dieser Zwecke eine Virtualisierung der Robotersteuerung mittels Offline-Simulator bereit.² Über die Virtualisierungssoftware VirtualBox wurde eine Linuxumgebung geschaffen, auf der die Simulationssoftware URSim 5.9.4 die Robotersteuerung der eSerie und CB3 Serie nachstellen kann. Mit den passenden Netzwerkeinstellungen kann die eigens erstellte Software für den Mock-up von dem primären Windowssystem auf die virtualisierte Steuerung zugreifen und getestet werden.

Der Softwareentwicklungsprozess gestaltete sich wie folgt: Die Funktionen wurden in individuelle Module unterteilt, die virtuell am digitalen Zwilling getestet wurden und anschließend über die Onlineprogrammierung am realen Roboter. Die daraus resultierenden Ergebnisse wurden in die nächste Iterationsstufe überführt und das Vorgehen wurde wiederholt. Aus Projektsicht ließen sich nahezu unbegrenzt viele Funktionen und Tools in den Mock-up einbauen, um das Nutzererlebnis zu verbessern. Hier wurde aufgrund der Zeit- und Ressourcenrestrektion darauf geachtet, nur für das Forschungsergebnis relevante Features umzusetzen. Das Ziel war es mit der Software die eingangs gestellte Forschungsfrage (siehe Kapitel 1.2) zu beantworten.

² Vgl. <https://www.universal-robots.com/download/software-e-series/simulator-non-linux/offline-simulator-e-series-ur-sim-for-non-linux-594/> (zuletzt aufgerufen am 27.02.2022))

3.2 Sony DualSense Controller

Aufgrund der Zeitrestriktion für die vorliegende Arbeit wurde auf die Konstruktion eines eigenen Eingabegerätes verzichtet und ein proprietärer Joystick/Controller verwendet. Hierfür wurde zuerst eine umfangreiche Online-Marktrecherche nach möglichen Lösungen durchgeführt, hauptsächlich über ‚google.de‘ und ‚reddit.com‘, unter anderem mit den Suchbegriffen ‚feedback controller‘, ‚sensoric feedback Joystick‘ und ‚haptic machine controll‘ sowie deren deutsche Übersetzung. Es wurden unterschiedliche Hersteller (Spohn & Burkhardt GmbH & Co. KG, MEGATRON Elektronik GmbH & Co. KG, Althen Sensors & Controls NL) von Industriejoysticks/Eingabegeräten angeschrieben und angefragt. Dabei wurden nicht nur explizit Geräte zur Robotersteuerung gesucht, sondern allgemein Controller/Joysticks/Eingabegeräte aus Gebieten, die eine Lösung für die Problemstellung darstellen könnten. Die Ergebnisse wurden tabellarisch zusammengefasst und anhand der technischen Spezifikationen kategorisiert. Diese sind in der Tabelle 2 bis Tabelle 5 dargestellt, wobei hier nur käuflich erhältliche Modelle inkludiert wurden³.

Um ein passendes Device zu eruieren, wurden die Ergebnisse methodisch mit Hilfe einer Nutzwertanalyse analysiert. Diese ist ein Instrument zur systematischen Entscheidungsfindung (Zangemeister 2014, S. 44).

Im Rahmen der Nutzwertanalyse wurden die fünf verschiedenen Bewertungskriterien Preis, Feedback, Verfügbarkeit, Bedienung und Machbarkeit definiert. Das Kriterium ‚Preis‘ dient der Bewertung der entstehenden Kosten zur Umsetzung des Mock-ups mit gegebenem Device und könnte durch die Budgetrestriktion des Lehrstuhls zum K.-o.-Kriterium werden. Das ‚Feedback‘ gibt Aufschluss über die unterschiedlichen Möglichkeiten, haptisches oder audiovisuelles Feedback durch das Eingabegerät darzustellen. Die ‚Verfügbarkeit‘ setzt sich aus den Faktoren Lieferzeit und Reaktion des Händlers auf die E-Mail-Anfrage zusammen. Die ‚Bedienung‘ gibt einen Hinweis, ob sich das Produkt überhaupt für die Steuerung eines Roboters eignet. Als letztes Kriterium umreißt die ‚Machbarkeit‘ die Sinnhaftigkeit der Lösung sowie die gegebenen Möglichkeiten, um mit dem Eingabegerät die Forschungsfrage zu beantworten.

³ Bei einem Teil der eruierten Geräte handelte es sich um Ergebnisse von Forschungsvorhaben oder sie wurden in wissenschaftlichen Publikationen beschrieben, standen jedoch nicht zum Verkauf.

Nach der umfassenden Recherche zu allen Anforderungsbereichen der Eingabegeräte wurde jedes Kriterium qualitativ mit 1 (sehr schlecht) bis 5 (sehr gut) bzw. 0 (K.-o-Kriterium) bewertet. Es wurden alle Bereiche gleich gewichtet und ein Durchschnittswert (arithmetisches Mittel) berechnet.

Da es sich um eine Ordinalskala handelt, ist eine Mittelwertberechnung laut Theorie methodisch nicht korrekt, jedoch gibt diese Art der Nutzwertberechnung für den vorliegenden Anwendungsfall ein sinnvolles und verwertbares Ergebnis. Zudem ist, trotz der statistischen Verzerrung, das hier angewandte Vorgehen eine gängige Praxis für die Nutzwertanalyse (Zangemeister 2014, 44 ff.).

Tabelle 1: Nutzwertanalyse der Eingabegeräte

Produkte	Preis	Feedback	Verfügbarkeit	Bedienung	Machbarkeit	Mittelwert
Sony DualSense Controller	5	3	5	5	3	4,2
Haptx Gloves	0	5	3	5	5	3,6
Touch von 3D Sxstems	3	3	1	4	5	3,2
Phantom Premium	2	4	1	4	5	3,2
Reactive Grip Motion Controller	3	5	1	4	3	3,2
Stratos Explore	3	3	5	3	5	3,8
CLS-E Force-Feedback Joystick	3	3	5	4	3	3,6
EMYCSA – RealSimulator	4	3	5	5	3	4
Sensodrive - Force-Feedback Joystick	2	4	2	4	4	3,2
Spacemouse	5	1	5	4	1	3,2
90JAM-Joystick	2	3	2	5	3	3
HE1-32-R-A-5-BK	5	1	5	4	1	3,2
Kategorien	Beschreibung				Skalenniveau	Verbalisierung
Preis:	Kosten der Beschaffung.				1	sehr schlecht
Feedback:	Art des sensorischen und haptischen Feedbacksystems.				2	schlecht
Verfügbarkeit:	Lieferzeit und Reaktion auf E-Mail-Anfragen des Herstellers.				3	mittel
Bedienung:	Art der Bedienung und Sinnhaftigkeit zur Robotersteuerung.				4	gut
Machbarkeit:	Möglichkeit der Umsetzung und Einschätzung zur Realisierung.				5	sehr gut

Die Nutzwertanalyse hat als Ergebnis den proprietären Joystick/Controller DualSense von Sony. Mit einer Dimension von rund 160 x 66 x 106 mm und einem Gewicht von 280 g gehört er zu den kleineren und leichteren Bedienelementen der Gruppe (siehe Abbildung 11).

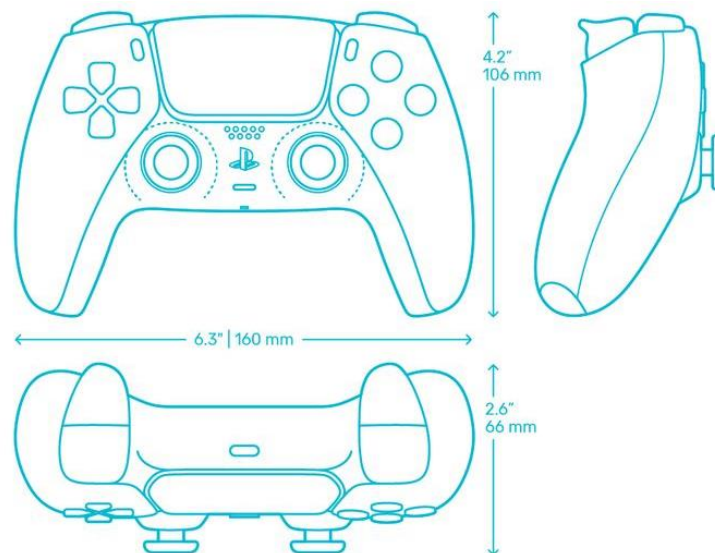


Abbildung 11: Technische Skizze des Sony DualSense Controllers⁴

Das Gerät weist eine Vielzahl (18) von Buttons sowie zwei Joysticks mit zwei Freiheitsgraden auf. Bei den hinteren Tasten handelt sich um adaptive Trigger, bei denen individuelle Widerstände eingestellt werden können. Als haptisches Feedback setzt der Controller auf Vibration in unterschiedlichen Frequenzen. Hierzu sind zwei Vibratoren verbaut, jeweils einer an der rechten und linken Seite. Des Weiteren sind ein Mikrofon, Lautsprecher und ein Akkumulator integriert. Als Schnittstellen fungieren ein 3,5-mm-Klinkenstecker als Audioausgang sowie ein USB-C-Eingang zum Laden und zur Datenübertragung, die zudem über Bluetooth realisiert werden kann. Der Controller hat einen eingebauten Lithium-Ionen-Akku mit einer Akkukapazität von 1560 mAh. Die Spannung beträgt 3,65 Volt. Abbildung 12 zeigt das Innenleben des Controllers.

⁴ Bildquelle: <https://i.pinimg.com/736x/09/55/86/095586e212abe07a15182789d3ad2e60.jpg> (zuletzt aufgerufen am 25.03.2022)



Abbildung 12: Elektronik und Innenleben des Sony DualSense Controllers⁵

Für den Mock-up sind die haptischen Elemente und die Ansteuerung der unterschiedlichen Tasten entscheidend. Insbesondere sind die beiden Vibratoren an der Haltevorrichtung des Controllers relevant (rechts und links unten). Es handelt sich jeweils um einen Linear Resonant Actuator, bei dem eine Spule eine lineare Bewegung von Magneten auslöst, die an elastischen Teilen wie Federn oder Silikonmembranen aufgehängt sind. Beide Aktuatoren sind von der Foster Electric Company; sie erzeugen Vibrationen in unterschiedlicher Frequenz und Intensität.

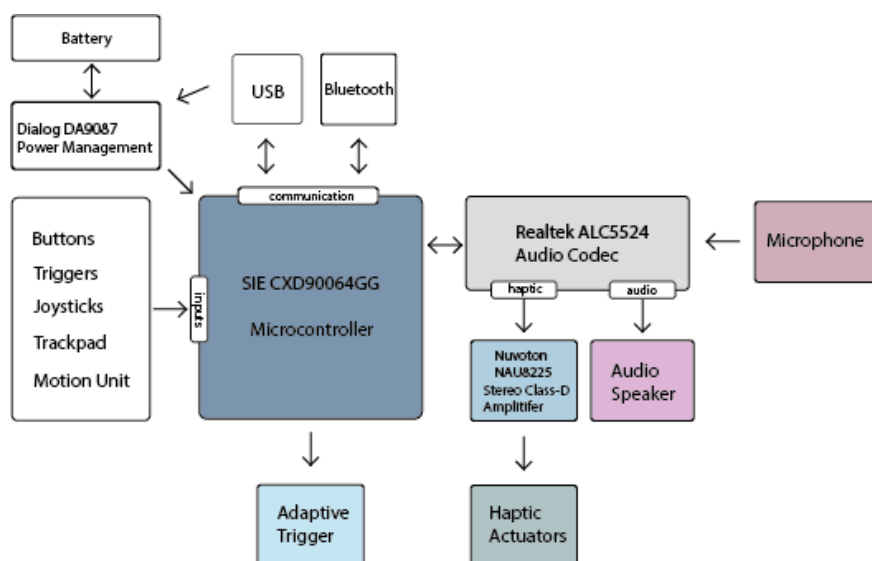


Abbildung 13: Schematischer Aufbau der Elektronik des Controllers⁶

⁵ <https://www.youtube.com/watch?v=4rNITneXnCo> (zuletzt aufgerufen am 09.02.2022)

In Abbildung 13 ist schematisch dargestellt, wie die aufgeführten Komponenten des Controllers angeordnet sind und so eine Steuerung mit haptischem Feedback ermöglichen.

Neben der Nutzung des haptischen Feedbacks wären auch audiovisuelle Signale mit dem Eingabegerät realisierbar. Über eine Farbcodierung könnten Informationen durch das Beleuchtungssystem oder über die integrierten Lautsprecher Audiosignale vermittelt werden. Das erweitert die haptische Komponente um eine zusätzliche Ebene und ermöglicht eine Nutzung der Konzepte der Human Factors. Jedoch wird agil im Projektverlauf entschieden, ob eine Implementierung dieser Features für das Mock-up und das Bedienkonzept notwendig sind.

Durch die Tatsache, dass es sich bei dem Eingabegerät um ein bekanntes Produktdesign der Consumerelektronik (Gamecontroller) handelt, ergeben sich hieraus mehrere Vorteile. Durch die Gamification der Robotersteuerung wird die Hemmschwelle für Nicht-Fachkräfte verringert und das Feld der Roboterprogrammierung/-steuerung wird einer breiteren Masse zugänglich gemacht. Das kann für die Berufsorientierung und/oder Bildung für Schüler und Studenten genutzt werden, um diesen einen einfachen Einstieg in das Thema zu bieten. Darüber hinaus ist zu vermuten, dass die Hürden der Umgewöhnung oder das Anlernen gering ausfallen, da es sich um ein den meisten Personen bekanntes Eingabegerät handelt.


⁶ https://uploads-ssl.webflow.com/5eb03713da929d4bea15c26b/60f82aafbd2361204c2cccd5_ps5-dualsense-system-sony-controller-schematisation.png (zuletzt aufgerufen am 23.03.2022)

3.3 Cobot von Universal Robots

Die Umsetzung des technischen Mock-ups sollte mit einem Industrieroboter erfolgen. Aufgrund der industriellen Relevanz und der thematischen Verknüpfung wurde festgelegt, eine alternative Steuerung für einen 6-Achs-Roboterarm umzusetzen. Die Umsetzung könnte mithilfe eines Industrieroboterarms mit integriertem Kraft-Momenten-Sensor realisiert werden, oder mithilfe eines Cobots, der bereits über die erforderliche Sensorik verfügt. Aufgrund der in Kapitel 2.2 vorgestellten Potenziale der kollaborativen Zusammenarbeit und der Mensch-Maschine-Schnittstelle von Cobots sowie deren stetig steigender Relevanz im Robotiksegment wurde entschieden, einen kollaborativen Roboterarm auszuwählen.

Das Unternehmen Universal Robots ist als Innovator des Cobot-Marktes weit verbreitet und hat seit 2016 mit UR+ den Plattform-Gedanken in seiner Robotersoftware adaptiert. Damit stellt das Unternehmen Schnittstellen, ein Software Development Kit (SDK) sowie eine Virtualisierung des Robotercontrollers frei zur Verfügung. Diese offene Programmierumgebung ist ursächlich dafür, dass sich Cobots von Universal Robots besonders im wissenschaftlichen Bereich für Versuche, Mock-ups und zu Lehrzwecken etabliert haben. Aus diesem Grund (und wegen der Verfügbarkeit im Technikum) wurde die Umsetzung mit einem Cobot von Universal Robots realisiert. Bei den Cobots von Universal Robots handelt es sich um einen typischen seriellen 6-Achs-Roboterarm mit einer offenen kinematischen Kette und sechs Freiheitsgraden.

Universal Robots hat seine Produktionslinie im Jahr 2012 mit der sogenannten CB-Serie begonnen, diese bietet drei Variationen mit 3 kg, 5 kg oder 10 kg Traglast und unterschiedlichen Reichweiten. Nach der 3ten Produktiteration (CB3) wurde 2018 mit der e-Series eine neue Generation von Cobots vermarktet (siehe Abbildung 14).







	UR3e	UR5e	UR10e	UR16e
 Reach	500 mm / 19.7 ins	850 mm / 33.5 ins	1300 mm / 51.2 ins	900 mm / 35.4 ins
 Payload	3 kg / 6.6lbs	5 kg / 11 lbs	12.5 kg / 27.55 lbs	16 kg / 35.3 lbs
 Footprint	Ø 128 mm	Ø 149 mm	Ø 190 mm	Ø 190 mm
 Weight	11.2 kg / 24.7 lbs	20.6 kg / 45.4 lbs	33.5 kg / 73.9 lbs	33.1 kg / 73 lbs

Abbildung 14: Aktuelles Produktportfolio von Universal Robots mit technischen Spezifikationen⁷

Mit dem integrierten Kraft-Momenten-Sensor am Werkzeugflansch ermöglicht die e-Series eine genauere sensitive Erfassung von Krafteinwirkungen und ist prädestiniert für die Umsetzung des Mock-ups. Zudem wurde die Signalfrequenz von 125 Hz auf 500 Hz erhöht, was eine gleichmäßigere Steuerung ermöglicht, da Bewegungssignale schneller übertragen werden können. Prinzipiell wird eine universelle Programmierung für die CB3- und e-Serie angestrebt, jedoch muss das haptische Feedback an die Roboterart angepasst werden, weil die Robotersteuerung aufgrund der unterschiedlichen Kraftmessung unterschiedliche Werte für die Kraftfunktionen ausgibt. Der Mock-up wird primär für den UR3e erstellt, aber auch für den UR5e und UR3 CB getestet.

Um den Cobot zu programmieren, gibt es unterschiedliche Möglichkeiten. Das Teach-Panel (Handprogrammiergerät) hat eine grafische Benutzeroberfläche, in der Programme über vorgefertigte Module zusammengestellt werden können. Diese werden als grafische Blöcke zur Deklaration von Variablen, logischen Ausdrücken, Schleifen, Roboterbewegungen oder Anwendungen dargestellt. Diese werden nacheinander vom Roboter ausgeführt und ermöglichen eine Programmierung, ohne sich Gedanken über die Syntax machen zu müssen. Mit der Plattform UR+ lassen sich unkompliziert sogenannte URCaps programmieren und damit vorgefertigte Roboteranwendungen veröffentlichen. Hersteller von Hardware und Zubehör (Endeffektoren, Sensoren, Sicherheitseinrichtungen, Kameras u. a.) können fertige Funktionen anbieten, die herkömmlich über

⁷ <https://www.universal-robots.com/products/> (zuletzt aufgerufen am 12.03.2022)

das Schalten und Auslesen von Ein- und Ausgängen umständlich selbst programmiert werden mussten.

Dieses Konzept wurde von verschiedenen Cobotherstellern adaptiert, weswegen die Roboterprogrammierung von Cobots oft als intuitiv oder einfach bzw. als in kürzester Zeit erlernbar bezeichnet wird. Die vorgefertigten Blöcke schränken den Anwender jedoch auch ein, weshalb mit einer textuellen Programmierung Skripte über die eigens von Universal Robots bereitgestellte Programmiersprache Polyscope erstellt werden können. Während die speicherprogrammierbaren Steuerungen nach IEC 61131 genormten und in der Regel gleichen Konventionen in der Programmierung unterliegen, hat jeder Roboterhersteller eine eigene Programmiersprache mit individuellen Funktionen, Syntax, Umfang und Möglichkeiten. In diesem Mock-up wird ein externes Script in Python genutzt, das auf die Funktionen der Polyscope-Programmierung zurückgreift und Funktionen der Programmierungsumgebung an den Roboter-Controller überträgt oder abfragt.

3.3.1 Interfaces

Die Roboter von Universal Robots besitzen eine Vielzahl von standardisierten Schnittstellen, die zur Übermittlung von Daten genutzt werden können. Hieraus ergab sich die erste Fragestellung, welche Schnittstellen für die Umsetzung des Demonstrators am sinnvollsten genutzt werden konnten.

Es ist möglich, eine TCP/IP-Socket-Communication über die frei verfügbaren Ports zu erstellen und das TCP/IP-Protocol zu nutzen. Der Roboter ist hierbei der Client und der PC im Netzwerk fungiert als Server. Über die individuelle Schnittstelle können in der robotereigenen Programmiersprache URScript Befehle übertragen werden.

Darüber hinaus existieren vordefinierte standardisierte Schnittstellen auf gewissen Ports in der Remote Control des UR-Cobots über TCP/IP. Diese unterscheiden sich in der Datenübertragungsrate, den übermittelten Datenpaketen und den zugewiesenen Ports. Das Primary und Secondary Interface bieten Roboterzustandsdaten und können URScript-Befehle mit einer Aktualisierungsrate von 10 Hz übersenden (Port 30001, 30011, 30002, 30012). Das Real-time-Interface ist von der Funktion identisch mit dem Primary und Secondary Inter-

face, jedoch ermöglicht es eine Übertragungsrate von bis zu 500 Hz (125 Hz bei der CB-Series) und somit Echtzeitfähigkeit (Port 30003,30013).

Die RTDE(Real-Time Data Exchange)-Schnittstelle bietet eine Möglichkeit, externe Anwendungen über eine Standard-TCP/IP-Verbindung mit der UR-Steuerung in Echtzeit zu synchronisieren (Port 30004). Bei der Verbindung mit der RTDE-Schnittstelle ist der Client (externer PC) für die Einrichtung der zu synchronisierenden Variablen verantwortlich. Es kann eine beliebige Kombination von Eingangs- und Ausgangsregistern angegeben werden, die der Client schreiben bzw. lesen muss. Der Echtzeit-Datenaustausch (RTDE) erzeugt standardmäßig Ausgabemeldungen mit 125 Hz (bis zu 500 Hz in der eSerie). Allerdings hat die Echtzeitschleife im Controller eine höhere Priorität als die RTDE-Schnittstelle. Wenn der Controller nicht über genügend Rechenressourcen verfügt, wird er daher eine Reihe von Ausgabepaketen auslassen. Die übersprungenen Pakete werden nicht nachgereicht, sondern der Regler sendet immer die aktuellen Daten. Eingangspakete werden in dem Kontrollzyklus verarbeitet, in dem sie empfangen werden, sodass die Belastung des Controllers je nach der Anzahl der empfangenen Pakete variiert. Für Eingangsmeldungen wird keine konstante Aktualisierungsrate benötigt; die Eingänge behalten ihren zuletzt empfangenen Wert und werden ebenfalls mit maximal 125 (500) Hz übertragen. Bei der Ausführung an einem externen PC muss die Netzwerklatenz und Rechenleistung des Prozesses betrachtet werden, um die maximale Übertragungsrate sicherzustellen.

Aufgrund der Echtzeitfähigkeit und Vielfalt der übertragenen und auszulesenden Daten wurde entschieden, dass RTDE-Interface von Universal Robots zur Umsetzung des Demonstrators zu nutzen. Die 10-Hz-Übertragungsrate des Primary und Secondary Interface könnte Verzögerungen in der Steuerung verursachen, die aus Bediener Sicht jedoch eine direkte Kontrolle ermöglichen sollte. Zudem bietet das RTDE-Interface mehr Optionen, um Daten und Befehle ein- bzw. auszulesen.

3.3.2 Sensorik

Das Schlüsselement, um das Mock-up zu realisieren, ist die Kraftmessung am Endeffektor des Roboters. Hierfür wird ein Sensor benötigt. Ein Sensor wird auf der untersten Stufe der Pyramide der Automatisierungstechnik eingeordnet und stellt ein technisches Bauteil dar, das aus einem Prozess zeitabhängige physikalische oder elektrochemische Größen erfasst und in ein elektrisches Signal umsetzt (Hesse und Schnell 2014). Kraftsensoren bestehen in der Regel aus einem Federkörper, der durch die zu messende Kraft elastisch und linear deformiert wird, sowie einer Vorrichtung zur Messung dieser Deformation. Diese Verformung wird in ein elektrisches Signal umgewandelt und an die Robotersteuerung übertragen. Die Messgenauigkeit wird durch ihre Montage und die Dynamik der Aufnahmestruktur im Robotersystem beeinflusst (Nguyen und Melkote 2019).

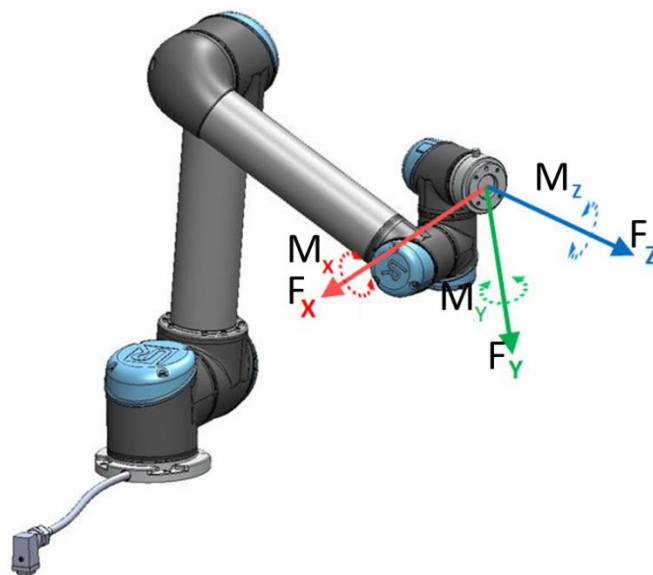


Abbildung 15: Richtung der einwirkenden Kräfte und Drehmomente am Roboterflansch⁸

Mehrkomponenten-Sensoren messen Kräfte (F) und/oder Momente (M), die in mehr als einer Raumrichtung auftreten – wie z. B. Messungen die wirkenden Kräfte in x-, y- und z-Richtung. Eine weitere physikalische Größe, die erfasst werden kann, sind die Drehmomente, die um die Achsen der Roboterendflansches wirken. Werden zusätzlich zu den Kräften auch Drehmomente ermittelt,

⁸ Vgl. https://aws1.discourse-cdn.com/business7/uploads/universal_robots/original/1X/1b7c74657a3b5377efcd28533b5ec8741f3f6220.jpg (zuletzt aufgerufen am 19.03.2022)

kann ein typischer Robotersensor am Endeffektor bis zu sechs Komponenten (drei Kräfte und drei Drehmomente) messen. In der Robotik existiert eine große Bandbreite an Kraft-Momenten-Sensoren, die zwischen dem Endeffektor und dem Roboterendflansch montiert werden können, um die auf den Roboter wirkenden Kräfte zu messen. Der Universal Robots Cobot in der e-Serie hat diesen schon am Roboterflansch eingebaut; dabei können die in die drei Dimensionen wirkenden Kräfte sowie die dazugehörigen Drehmomente ausgelesen werden. Diese Messbereiche und Spezifikationen können dem Datenblatt des jeweiligen Robotertyps entnommen werden (siehe Abbildung 16).

Spezifikationen der Kraft- und Momentensensoren des UR5e

F/T Sensor	Kraft, x-y-z	Moment, x-y-z
Messbereich	50,0 N	10,0 Nm
Auflösung	3,5 N	0,2 Nm
Genauigkeit	4,0 N	0,3 Nm

Spezifikationen der Kraft- und Momentensensoren des UR3e

F/T Sensor	Kraft, x-y-z	Moment, x-y-z
Messbereich	30,0 N	10,0 Nm
Auflösung	2,0 N	0,1 Nm
Genauigkeit	3,5 N	0,1 Nm

Abbildung 16: Spezifikationen der Kraft- und Momentensensorik aus den Datenblättern der UR-Serie

3.4 Script

Das Script zur Robotersteuerung verarbeitet die ausgegebenen Daten von Roboter und Controller und schickt Bewegungsbefehle zum Roboterarm, während es parallel dazu ein haptisches Feedback im Controller auslöst. Das Script ist die Grundlage der Mensch-Maschine-Schnittstelle und soll Ideen und Konzepte des Human Factors berücksichtigen, um die Robotersteuerung des Cobots von Universal Robots neu zu gestalten.

3.4.1 Python

Python ist eine kostenlose dynamische objektorientierte Programmiersprache. Sie verfügt über eine große Anzahl (kostenloser) wissenschaftlicher Toolboxes und Module mit vorgefertigten Funktionen. Sie ist so konzipiert, dass sie leicht zu programmieren und zu lesen ist. Sie wurde Anfang der 1990er Jahre von Guido van Rossum entwickelt und hat seither in einer Vielzahl von Bereichen, von der Webentwicklung über die Systemadministration bis hin zu Wissenschaft und Technik, große Popularität erlangt. Durch vollumfängliche kostenlose Distributionen, die komplette wissenschaftliche und technische Entwicklungssoftware für numerische Berechnungen, Datenanalyse und Datenvisualisierung auf der Basis von Python bereitstellen, erlangte sie eine signifikante Bedeutung für die Wissenschaft. Die Programmiersprache ist Open Source und hat sich in einigen Bereichen (z. B. Machine Learning, Bilderkennung, Data Science, Forschung) zu einer der erfolgreichsten Programmiersprachen entwickelt (Haslwanter 2021). Python bietet unterschiedliche Möglichkeiten mit ausführlicher Dokumentation, um die Schnittstellen von Universal Robots zu nutzen. Zudem können auch USB-Controller ausgelesen werden. Für den DualSense Controller existiert ein vorgefertigtes Modul.

Aus diesen Gründen fiel die Wahl auf Python als Programmiersprache, um das wissenschaftliche Mock-up in der Zeitrestriktion zu erstellen. Da das Script zum Demonstrator ein frei zugängliches Open-Source-Projekt werden soll, wird der PEP 8 – Style Guide for Python Code (<https://peps.python.org/pep-0008/>) zugrunde gelegt. Hierbei handelt sich um eine allgemeine Konvention, mit dem Ziel, die Lesbarkeit und Nachvollziehbarkeit des Codes zu verbessern und diesen mit einer klaren Konsistenz zu versehen.

3.4.2 Kommunikation zum Roboter

Das Pythonmodul ‚ur_rtde‘ ist aus einem Forschungsprojekt des dänischen Instituts Maersk Mc-Kinney Moller entstanden. Im Projekt ‚SDU Robotics‘ befasst sich das Institut mit Forschung in den Bereichen kognitive und angewandte Robotik. Mit der Version 1.5.0 wurde ein Framework für Python und C++ mit einer Vielfalt an Funktionen für den Universal Robots geschaffen (siehe Abbildung 17) (SDU Robotics).

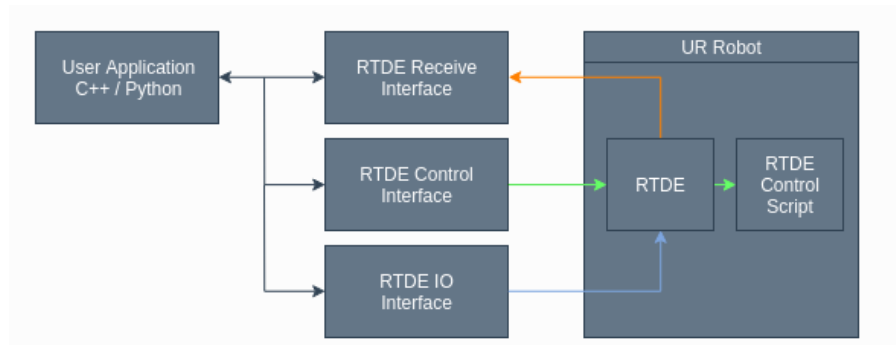


Abbildung 17: Schematische Darstellung zur Funktion von dem Pythonmodul ‚ur_rtde‘⁹

⁹ Vgl. https://sdurobotics.gitlab.io/ur_rtde/introduction/introduction.html (zuletzt aufgerufen am 19.03.2022)

Das Framework besteht aus drei Interfaces mit unterschiedlichen Funktionen, die auf die RTDE-Schnittstelle der kollaborierenden Roboter von Universal Robots zugreifen. Das ‚RTDE Control Interface‘ wird zur Steuerung des Roboters und für Hilfsfunktionen verwendet. Hierfür muss ein Steuerskript auf dem Roboter laufen, das automatisch hochgeladen wird (Grüner Pfeil). Das ‚RTDE Receive Interface‘ wird für den Empfang von Daten vom Roboter verwendet (Orange-farbener Pfeil). Das ‚RTDE IO Interface‘ dient zur Einstellung digitaler/analoger Inputs und Outputs (IOs) und zur Einstellung des Geschwindigkeitsreglers des Roboters (Blauer Pfeil) (SDU Robotics).

Die Aufteilung in die drei Schnittstellen ermöglicht eine Parallelisierung der unterschiedlichen Funktionen. Während Daten ausgelesen werden, ist es möglich, den Roboter zu bewegen und gleichzeitig Aus-/Eingänge zu schalten. In der API-Library (https://sdurobotics.gitlab.io/ur_rtde/api/api.html) sind die API-Klassen mit ihren Funktionen beschrieben. Das Framework ist mit allen Universal-Robots-Cobots der e-Serie und der CB3-Serie ab Softwareversion 3.3. kompatibel (SDU Robotics).

Alternativ zum Framework kann auch eine eigene Python-Schnittstelle zum RTDE-Interface erstellt werden. Hierzu bietet Universal Robots eine ausführliche Dokumentation mit vorgefertigten Demonstrationsprogrammen an, die entsprechend den eigenen Anforderungen angepasst werden können. Zu Beginn dieser Abhandlung wurde diese Möglichkeit evaluiert und die Programmierung damit gestartet. Die Strategie wurde jedoch nach einiger Zeit verworfen, da sie sich trotz der durch Universal Robots bereitgestellten fertigen Python-Testprogramme als umfangreicher und komplexer erwies als die Verwendung des ‚ur_rtde‘-Moduls. Da der Schwerpunkt der Abhandlung nicht in der Schnittstellenprogrammierung, sondern in der Erstellung eines Mock-ups liegt, um ein Proof of Concept zu ermöglichen, wurde – insbesondere aufgrund der Zeitrestriktion – daher das ‚ur_rtde‘-Framework genutzt.

3.4.3 Kommunikation zum Controller

Der Sony DualSense Controller kann über mehrere standardisierte Schnittstellen Daten übertragen. Eine Möglichkeit der Kommunikation ist Bluetooth, dem Industriestandard für die Datenübertragung zwischen Geräten über kurze Distanz per Funktechnik. Eine weitere drahtlose Verbindung kann über den WLAN(Wireless Local Area Network)-Standard der Normen IEEE-802.11 erzeugt werden, oder über ein Kabel mit dem Standard Universal Serial Bus (USB) für eine bit-serielle Datenübertragung.

Da sich der Roboterbediener bei der Online-Programmierung des Roboters in der Regel in der Nähe des Roboters befindet und der Controller über den USB-Anschluss geladen wird, fiel die Entscheidung, die Kommunikation über die USB-Schnittstelle zu realisieren.

Hierzu existiert das Python-Modul ‚pydualsense‘, das vom User flok (Florian K.) auf der Online-Plattform zur Versionsverwaltung für Software-Entwicklungsprojekte GitHub (<https://github.com/flok/pydualsense>) entwickelt wurde. Als Open-Source-Projekt ist die Software und deren Quelltext über eine MIT-Lizenz frei verwendbar. Das Modul ermöglicht es, Daten aus dem Sony DualSense Controller auszulesen und Funktionen anzusteuern. Dazu nutzt es die ‚hidapi‘-Library, die eine einfache und universelle Kommunikation von USB-Geräten für Windows, Linux und Mac unterstützt (<https://github.com/libusb/hidapi>). Mithilfe dieser Library lässt sich der Datenstrom auslesen und eruieren, welche Funktionen die ein- und ausgehenden Bytes haben. Das Modul ‚pydualsense‘ bietet eine umfangreiche Grundlage mit einer Vielzahl von Funktionen, um den Sony DualSense Controller zu bedienen. Aktuell befindet sich das Modul noch in der Entwicklung, jedoch sind alle für das technische Mock-up benötigten Funktionalitäten schon enthalten.

3.4.4 Grafische Benutzeroberfläche

Die grafische Benutzeroberfläche (GUI: graphical user interface) ist ein zentrales Element der Mensch-Maschine-Schnittstelle. Für die Funktionalität des Mock-ups und der wissenschaftlichen Fragestellung nur sekundär von Bedeutung, besitzt die grafische Benutzeroberfläche für den Anwender eine hohe Relevanz. Sie bereitet Informationen visuell auf und bietet eine Vielzahl an Mög-

lichkeiten, Human Factors – also psychische, kognitive und soziale Einflussfaktoren – im Mensch-Maschine-Systemen zu inkludieren (vergleiche Kapitel 2.4). Von Bedienelementen zu farblichen Signalen und zu Elementen der Gamification bietet die GUI eine Grundlage der Kommunikation zwischen Mensch und Maschine. Für das technische Mock-up wird eine GUI erstellt, die eine Erklärung zur Steuerung und zu den Funktionen des Controllers darstellt und die Systemwerte des Roboters visualisiert.

Um die GUI zu realisieren, wird das Python-Modul ‚tkinter‘ (<https://docs.python.org/3/library/tkinter.html>) genutzt, das eine Sprachanbindung für das GUI-Toolkit Tcl/Tk bereitstellt. Tcl/Tk ist keine einheitliche Bibliothek, sondern besteht aus mehreren verschiedenen Modulen, die jeweils eigene Funktionen und eine eigene offizielle Dokumentation haben. Mit ‚tkinter‘ können diese genutzt werden, um eine grafische Benutzeroberfläche für die meisten Unix-Plattformen (einschließlich macOS) sowie Windows-Systeme zu erstellen.

4 Ergebnisse

Im vierten Kapitel wird das technische Mock-up vorgestellt und konkret gezeigt, wie die in Kapitel 3 erläuterten technischen Entscheidungen und Vorgehensweisen umgesetzt wurden. Der Schwerpunkt liegt dabei auf der erstellten Software, deren Funktionen und Algorithmen.

4.1 Steuerungskonzept

Die Bewegungssteuerung des Roboters erfolgt über eine roboteruntypische Steuerung mittels Gamecontroller. Prinzipiell hat der Roboter (vergleiche Kapitel 2.1.2) sechs Freiheitsgrade und somit Bewegungsrichtungen in Form von Translation und Rotation oder Achsen, die direkt angesteuert werden können. Da jeder Freiheitsgrad in positiver und negativer Richtung verändert werden kann, werden 12 verschiedene Tasten bzw. Steuerungselemente benötigt, um den Roboter vollumfänglich zu bewegen. In Kapitel 2.3 wurden unterschiedliche Konzepte der diversen Roboterhersteller vorgestellt. Ideen der Steuerungskonzepte sollen nun auf den PS5-Controller übertragen werden. Hierzu wurden unterschiedliche Varianten getestet.

Als initiale und offensichtliche Entscheidung wurden die beiden Joysticks jeweils mit zwei Freiheitsgraden genutzt, um hiermit vier Freiheitsgrade des Roboters anzusteuern. Diese Steuerung ist ergonomisch positioniert und so ausgestaltet, dass die Daumen dort abgelegt werden. Die Bewegung von Charakteren oder Änderungen der Perspektive mit Joysticks sind aus zahlreichen Spielen bekannt. Eine Adaption auf die Robotersteuerung ist eine intuitive Lösung der Gamification, um ein Spielelement auf die Arbeitswelt zu übertragen. Die beiden Controllerjoysticks lassen sich jedoch nur mit je zwei Freiheitsgraden kreisförmig um einen Mittelpunkt bewegen, demnach fehlen noch zwei weitere Freiheitsgrade, die vom Roboter gesteuert werden müssen.

Das erste Konzept lehnte sich an die Idee der ABB-Steuerung an. Demnach wurden mit den beiden Joysticks drei Freiheitsgrade angesteuert; mit dem Betätigen einer Taste konnten weitere drei Freiheitsgrade angesteuert werden. So

war es z. B. möglich, Translation und Rotation zu unterscheiden. Im ersten Modus wurden die ersten drei Achsen bzw. die Translationen durchgeführt, im zweiten Modus wurden die weiteren drei Achsen bzw. die Rotationen des Roboters gesteuert. Somit war der rechte Joystick in der x- und y-Achse belegt und der linke in der x-Achse. Dieses Konzept wurde jedoch verworfen, da jeder Freiheitsgrad des Joysticks vier unterschiedliche Funktionen haben konnte. Dadurch erwies sich das Umschalten nicht als intuitiv, sondern wurde kompliziert. Das Konzept wäre für Controller geeignet, die nicht über ausreichend Knöpfe, Tasten oder Joysticks verfügen. Ansonsten sollten Mehrfachbelegungen für die gleiche Taste vermieden werden. Ein Vorteil dieser Umsetzung wäre gewesen, die Auslenkung der Joysticks zu nutzen, um die Geschwindigkeit des Roboters zu bestimmen. Tests ergaben jedoch, dass die Steuerung des Roboters nicht praktikabel war.

Aus diesem Konzept entwickelte sich der nächste Gedanke, jede Taste zur Bewegungssteuerung nur zweifach zu belegen, d. h. in Achsbewegung und TCP-Bewegung. Dieses Konzept wird von der Mehrheit der vorgestellten Hersteller mit Ausnahme von ABB genutzt. Auf diese Weise können beide Joysticks vollumfänglich und zusätzlich die hinteren R1-, L1- und R2- und L2-Tasten genutzt werden, um alle sechs Freiheitsgrade abzubilden. Für diese Art der Umsetzung wurde geprüft, ob eine Taste betätigt wurde und anschließend wurde der Roboter mit konstanter Geschwindigkeit verfahren. Diese sollte mit den Pfeiltasten erhöht oder verringert werden können. Diese Umsetzung ermöglichte eine sinnvolle Steuerung des Roboters mit intuitiver Umsetzung nach Methoden der Gamification. Der Controller wurde ähnlich wie in Spielen eingesetzt und gehalten. Jedoch wurde das Konzept nach einigen Versuchen überarbeitet, weil das haptische Feedback der L- und R2-Triggertasten nicht sinnvoll genutzt werden konnte.

Um das haptische Feedback anwenden zu können, wurde das Layout der Bewegungsbefehle abgeändert. Statt der hinteren L1-, L2-, R1- und R2-Tasten wurden die Pfeiltasten auf der linken Seite des Controllers genutzt. Die Geschwindigkeit wurde nicht mehr über die Pfeiltasten, sondern über R2 gesteuert, sodass hier das haptische Feedback eingebaut werden konnte. Über das Betätigen der X-Taste auf der rechten Seite wurde dann zwischen Achs- und TCP-Bewegung umgeschaltet. Nach kurzen Anwendungsversuchen am Roboter wurde das Konzept über eine Änderung verfeinert und so zur finalen Umset-

zung vorbereitet. Statt mit Tastendruck in unterschiedliche Modi für Achs- und TCP-Bewegung umschalten zu müssen, wurde die L2-Taste für die Achsbewegung und die R2-Triggertaste für die TCP-Bewegung verwendet. Das Betätigen dieser Tasten legt die Geschwindigkeit der jeweiligen Bewegungsform fest und ermöglicht anschließend ein Bewegen mittels Joystick und Pfeiltasten (siehe Abbildung 18)



Abbildung 18: Steuerungskonzept des PS5 Controllers¹⁰

¹⁰ Vgl. <https://i.pinimg.com/736x/09/55/86/095586e212abe07a15182789d3ad2e60.jpg> (zuletzt aufgerufen am 19.03.2022)

4.2 Steuerungsprogramm mit haptischem Feedback

Der Kern des technischen Mock-ups ist die Software, mit der die eingehenden Informationen von Roboter und Controller verarbeitet werden können. Die Daten werden anschließend aufbereitet und verarbeitet, um durch Funktionen wieder an die Geräte übergeben zu werden. In der Abbildung 19 ist der Datenfluss schematisch dargestellt.

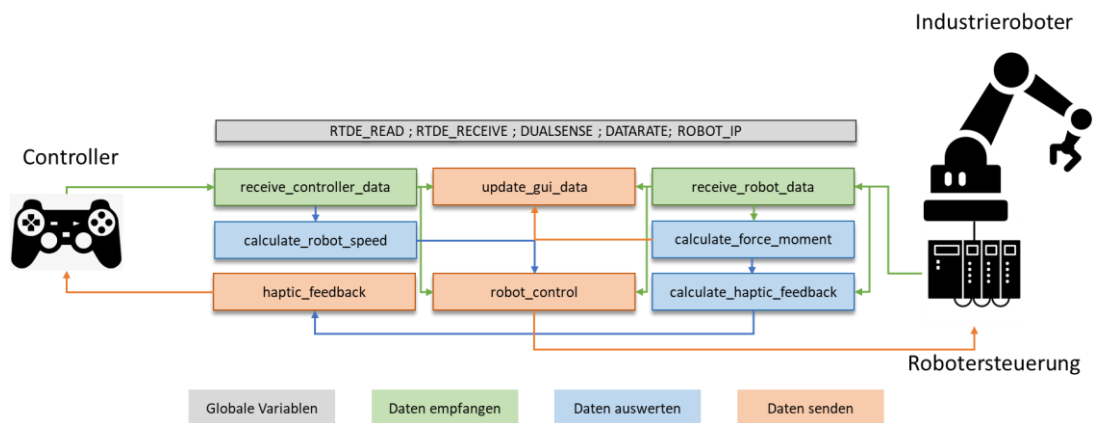


Abbildung 19: Schematische Aufbau der Software zur haptischen Robotersteuerung nach eigener Darstellung

Die beiden Funktionen `receive_controller_data` und `receive_robot_data` lesen die Daten vom Controller und Roboter aus (grün). Einige dieser Daten werden direkt in der GUI dargestellt oder zur Steuerung des Roboters genutzt, andere müssen erst weiterverarbeitet werden. Die drei Funktionen `calculate_robot_speed`, `calculate_force_moment` und `calculate_haptic_feedback` sind Hilfsfunktionen, die dazu genutzt werden, die Daten umzurechnen und für die Hauptfunktionen nutzbar zu machen (blau). Die eigentliche Steuerung und das haptische Feedback werden mit den Funktionen `haptic_feedback` und `robot_control` realisiert. Zudem wird die GUI mit der Funktion `update_gui_data` aktualisiert.

Initial müssen im Programm einmal die globalen Variablen festgelegt werden. Mit den Befehlen `RTDE_READ` und `RTDE_RECEIVE` wird die Verbindung mit dem Roboter initialisiert und mit dem Befehl `DUALSENSE` die des Controllers. Im String `ROBOT_IP` ist die IP des Roboters im Netzwerk hinterlegt, der zwingend für die Verbindung benötigt wird. Die globale Variable `DATARATE` legt die Frequenz fest, in der die Daten übertragen werden. Die CB3-Serie ist auf 125 Hz limitiert, die e-Series des URs ermöglicht 500 Hz, wobei die Leistung

des Computers, auf dem die Software läuft, den limitierenden Faktor darstellen kann. Aus diesem Grund wird in der GUI zudem die echte Datenrate des Prozesses wiedergegeben. Die Versuche werden wegen der Kompatibilität und der Leistung der Computersysteme mit 125 Hz durchgeführt.

In den ersten Zeilen des Programms werden die Python-Bibliotheken geladen, damit diese später genutzt werden können. Die bedeutenden größeren Frameworks wurden im Kapitel 3.4 vorgestellt. Des Weiteren wurden die Module `math` und `numpy` zur mathematischen Berechnung, `time` für Systemzeiten und `PIL` für Bilddarstellungen in der GUI importiert und verwendet.

Sobald das Programm aufgerufen und die GUI sowie die initialen Verbindungen realisiert wurden, wird eine unendliche While-Schleife gestartet, die alle Funktionen in der gewählten Frequenz ausführt. Es werden kontinuierlich die Daten übertragen, verarbeitet, dargestellt und das haptische Feedback sowie die Steuerung des Roboters realisiert, bis die Software beendet wird.

4.2.1 Daten auslesen

Zum Auslesen von Daten existieren zwei Funktionen, je eine für das jeweilige Gerät. Für beide muss initial eine Verbindung errichtet und an die Funktionen übergeben werden. Die Funktion `receive_roboter_data(rtde_read)` nutzt die Variable `rtde_read`, die zuvor die Verbindung zum Roboter aufbaut. Anschließend werden die Gelenkwinkel, TCP-Position, die Kraft am TCP, die TCP-Geschwindigkeit, das Moment, sowie der Sicherheitsstatus ausgelesen und in einer Liste ausgegeben. Listen sind in Python integrierte Variablen, die mehrere veränderliche Items geordnet speichern. Diese können einen beliebigen Datentyp ausweisen.

Die Funktion `receive_controller_data(dualsense)` nutzt die Variable `dualsense`, die zuvor die Verbindung zum Controller aufbaut. Anschließend werden die Pfeiltasten als boolesche Variable, die beiden Joysticks als Datentyp Integer im Intervall `[-127,128]` und die hinteren Trigger als Integer im Intervall `[0, 255]` ausgelesen.

Diese werden ebenfalls als Liste ausgegeben. Die übrigen Tasten könnten als `bool` ausgelesen werden; sie sind jeweils aufgrund von Performancegründen auskommentiert.

4.2.2 Datenberechnung und Datentransformation

Nicht alle Rohdaten des Controllers und Roboters können direkt genutzt werden, weswegen drei Funktionen zur Weiterverarbeitung von Daten inkludiert wurden. Diese Funktionen wandeln die Rohdaten um, sodass sie in anderen Funktionen verwendet werden können.

Die Funktion `calculate_roboter_speed` berechnet die Inkremente, mit der sich die Position des Roboters in der TCP oder Gelenkbewegung verändert. Die Geschwindigkeit wird durch das Betätigen der hinteren Triggertasten (R2 und L2) bestimmt. Diese senden an die Funktion `receive_controller_data` Werte zwischen 0 und 255. Die Geschwindigkeit kann durch die L2- und R2-Taste linear vom Bediener angepasst werden und ergibt bei durchgedrückten Triggertasten eine maximale Geschwindigkeit von 1 m/s in TCP Bewegung und 60 °/s in Gelenkgeschwindigkeit. Die Funktion ist reziprok abhängig von der Datenrate, da sich die Geschwindigkeit bei höherer Datenrate bei konstantem Inkrement be-

schleunigen würde. Der Grund ist, dass neue Bewegungsbefehle mit einer weiteren Position schneller übermittelt werden würden. Die beiden Skalierungsfaktoren (`scalefactor_tcp` und `scalefactor_joint`) bestimmen die Größe der übermittelten Bewegungsinkremente und sind auf die Maximalgeschwindigkeiten abgestimmt.

Die Funktion `calculate_force_moment` speichert die Beträge der Kräfte und Momente, die auf den TCP wirken, in einer Liste, sowohl einzeln in x-, y- und z-Richtung als auch als Summe.

Die Funktion `calculate_haptic_feedback` ist die Grundlage des haptischen Feedbacks, das der Nutzer beim Bedienen des Roboters erlebt. Hierfür stehen im Controller zwei Vibratoren sowie haptisches Feedback in den Triggertasten zur Verfügung. Sowohl die Triggertasten als auch die Vibratoren erlauben einen Input im Intervall von [0,255] als Integerwert, um die Intensität des Feedbacks zu bestimmen. Drei unterschiedliche Parameter bestimmen das Feedback des Bedieners, damit dieser spüren kann, was der Roboter ertastet. Die Roboter Geschwindigkeit und die am TCP wirkenden Kräfte und Momente sorgen für die Haptik im Controller. Um die Werte so zu transformieren, dass sie in einem Intervall von 0 bis 255 auftreten, wird eine logistische Funktion angewandt.

$$\text{Logistische Funktion: } f(x) = \frac{a * S}{a + (S - a) * e^{-S * k * x}}$$

Dabei ist a die untere Schranke und S die obere Schranke. Im vorliegenden Fall ist S somit als 255 definiert, dass das maximale Feedback ermöglicht. Die untere Schranke wird als 0,1 festgelegt. Aufgrund der Umwandlung in einen Integerwert würde die untere Schranke als 0 interpretiert werden und zu keinerlei haptischem Feedback führen. Aufgrund der Eigenschaften der logistischen oder auch Sigmoidfunktion genannten mathematischen Graphen eignet sich die Funktion für das Transformieren in einen Wert für das Feedback. Es handelt sich um eine beschränkte und differenzierbare reelle Funktion, die in diesem Fall auf den maximalen Eingabewert beschränkt wird, um Fehlfunktionen zu vermeiden. Die Funktion verläuft bis zum Wendepunkt stetig steigend (erste Ableitung > 0), besitzt jedoch nach dem einzigen Wendepunkt eine negative erste Ableitung der Funktion. Aufgrund der langsamen Steigung am Anfang existiert ein größerer Wertebereich, in der die Funktion als Ergebnis nahe 0 ist. Das ist sinnvoll, da erst ein Feedback auftreten soll, sobald größere Kräfte auf den Roboterarm wirken – und nicht bereits bei jeder kleineren Bewegung. Auf

der anderen Seite gibt es zudem einen größeren Bereich nahe der oberen Schranke, was dafür sorgt, dass ein starkes haptisches Feedback schon in der Nähe der maximalen Krafteinwirkungen auftritt, und nicht erst, nachdem die Grenzwerte erreicht wurden.

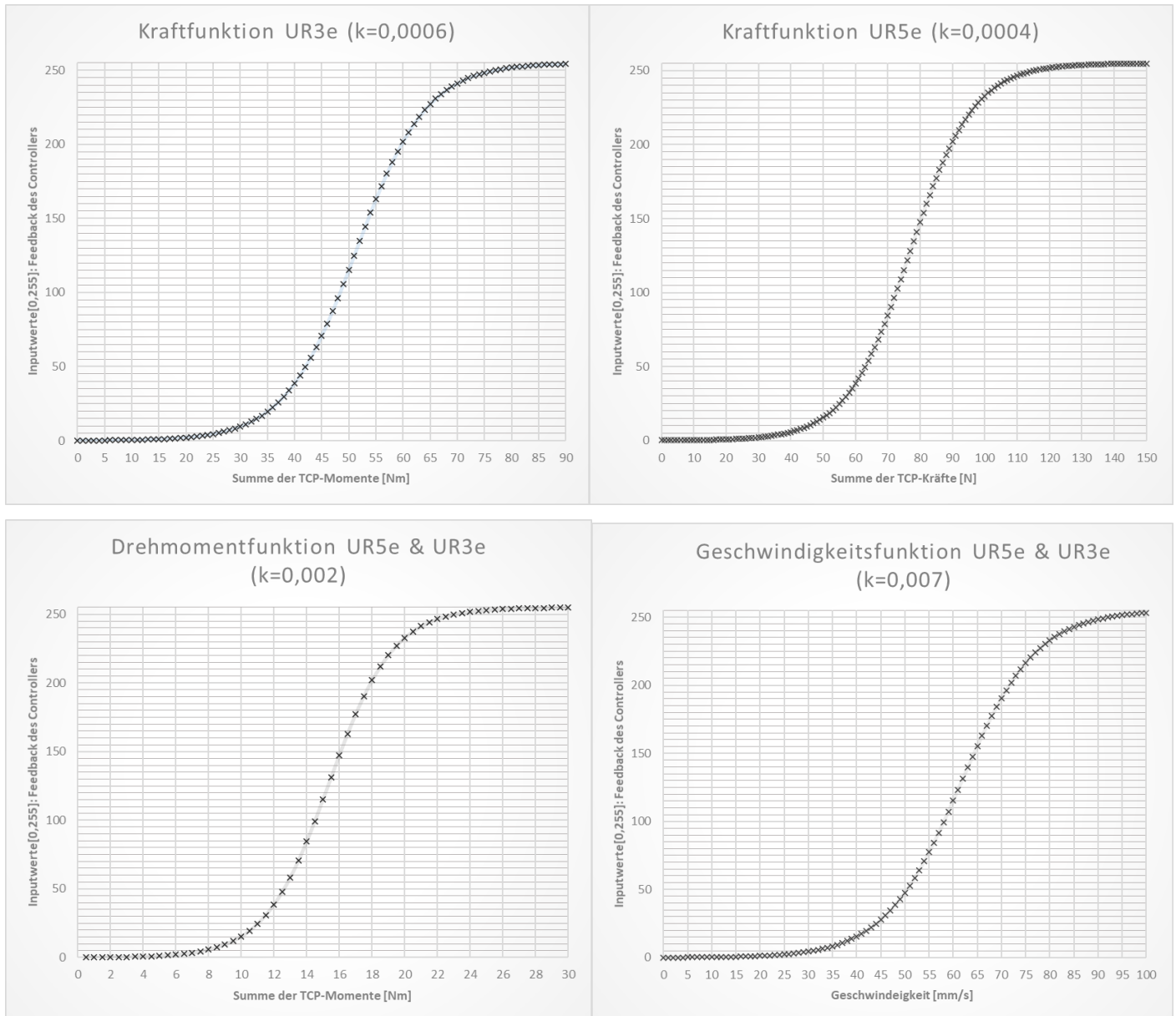


Abbildung 20: Sigmoidfunktion der drei Parameter des haptischen Feedbacks nach eigener Darstellung

In der Abbildung 20 ist die Funktion dargestellt. Es wird ersichtlich, wie sich die Intensität des haptischen Feedbacks in Abhängigkeit von Kraft, Moment und Geschwindigkeit ändert. Die Konstante k legt die Form der Kurve vor und wurde durch Testen ermittelt, sodass sowohl die obere als auch die untere Schranke einen größeren Bereich abdecken, um ein sinnvolles Feedback zu ermöglichen. Alle Funktionen werden in Integer-Werte konvertiert, da nur diese von den

Feedbackfunktionen des Controllers interpretiert werden können. Hieraus werden vier Variablen erzeugt: das Kraftfeedback (`force_feedback`), das Momentenfeedback (`moment_feedback`), das Geschwindigkeitsfeedback (`speed_feedback`) und eine Kombination aus allen drei Variablen, die den Mittelwert abbildet (`trigger_feedback`). Diese vier Variablen werden in einer Liste gespeichert und können an andere Funktionen übergeben werden.

4.2.3 Hauptfunktionen und Datenübergabe

Die Hauptfunktionen der Software ermöglichen die Steuerung des Roboters, geben das haptische Feedback wieder und updaten die GUI. Dies wird ermöglicht, indem Daten, Befehle und Funktionen an die Geräte und die GUI übergeben werden.

Die erste Funktion (`controller_feedback(dualsense, haptic_feedback)`) nutzt sowohl die Dualsense-Verbindung zum Controller, um die Funktionen und Daten zu übertragen, als auch die Liste, die in der Funktion `calculate_haptic_feedback` berechnet wurde. Die Vibration wird als haptisches Signal für die Momente und Kräfte genutzt. Der rechte Vibrator reagiert auf Momente, die auf den TCP wirken, während sich der linke Vibrator im Controller bei steigenden Kräften am TCP intensiviert. Somit soll der Roboterbediener erfahren, wenn zu hohe Lasten auf den Roboter einwirken und schnell reagieren können. Da die Trigger dauerhaft gedrückt sind, sobald sich der Roboter bewegt, ist hierdurch auch die Geschwindigkeit bestimmbar. Um eine ungewollte schnelle Bewegung am Roboter zu verhindern, ist in den R2- und L2-Triggertasten eine graduell ansteigende Kraft in Abhängigkeit zur Roboterbewegung implementiert. Das soll dem Roboterbediener eine Rückmeldung zur Geschwindigkeit geben und ein feinfühliges Arbeiten mit dem Roboter ermöglichen. Da der Roboter jedoch auch bei maximal eingestellter Geschwindigkeit gut bedienbar sein soll, ist die Geschwindigkeit nur anteilig als haptisches Feedback in den Triggertasten hinterlegt. Zu einem Anteil von zwei Drittel werden auch die Momente und Kräfte einbezogen. Das ermöglicht neben der Vibration auch ein reales Wahrnehmen von z. B. Objekten, die der Roboter berührt. Eine Kollision mit einem Widerstand (z. B. einem Objekt im Arbeitsraum des Roboters) verursacht ein haptisches Feedback der L2- und R2-Triggertasten und führt zu einem Nachempfinden des Widerstandes für den Bediener.

Die Funktion `update_gui_data` aktualisiert die Daten in der Benutzeroberfläche. Das ist ein wichtiger Bestandteil der Mensch-Maschine-Schnittstelle, weil hier erste Informationen zum Roboter ersichtlich werden. Eine Übersicht zu den Funktionalitäten des Controllers ist abgezeichnet, gefolgt von der IP des Roboters und der Frequenz. Anschließend werden alle Labels der variablen Anzeigewerte über eine `config()`-Funktion aus der `tkinter`-Bibliothek aktualisiert, indem der Text in dem vorher initiierten Label überschrieben wird. Dabei werden Operatoren und Bedingungen genutzt, um die Werte in einer für den Nutzer sinnvollen und übersichtlichen Art darzustellen. Der Bediener soll wichtige Informationen des Roboters auf einen Blick dargestellt bekommen, sodass eine optimale Interaktion mit dem Roboter gewährleistet ist. Mittels farblicher Indikatoren werden Grenzwerte in der Robotersteuerung angezeigt und für den Roboterbediener visualisiert, damit sich dieser nicht nur auf die Fülle an Zahlen konzentrieren muss, sondern ein weiteres Element des Mensch-Maschine-Interface nutzen kann.

In der Funktion `update_gui_data` (GUI-Aktualisierung) werden alle vorherigen Funktionen als globale Variablen in die Funktion geladen, aus diesem Grund darf `update_gui_data` in der endlosen While-Schleife des Hauptprogrammes erst aufgerufen werden, nachdem die übrigen Funktionen einmal in Variablen geschrieben wurden. In Abbildung 21 ist die GUI in der Version 0.5 der Software dargestellt.

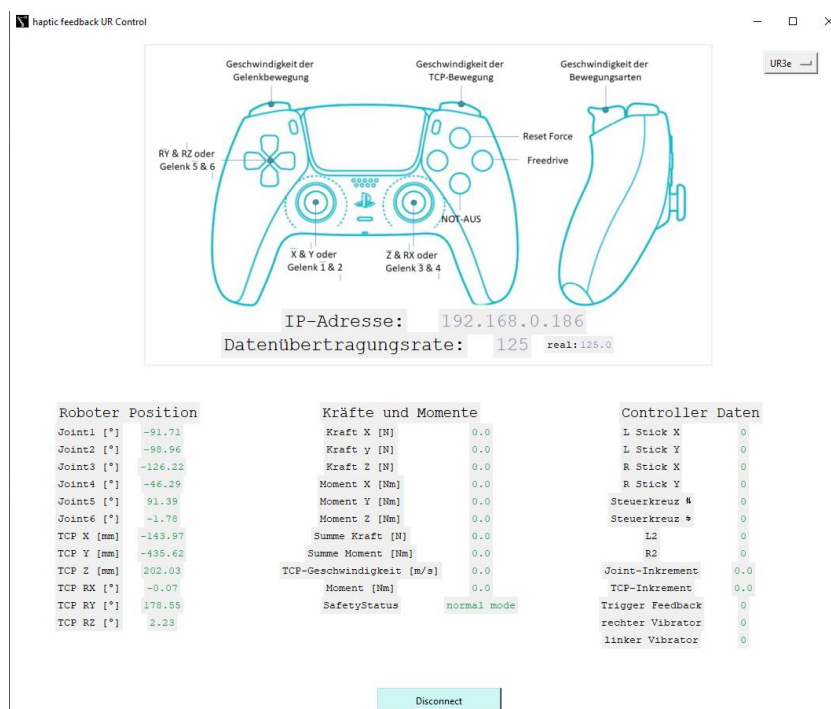


Abbildung 21: GUI des Programms zum Steuern des Roboters

Eine weitere Funktion des Programms ist `roboter_control`. Diese nutzt das `rtde_control` Befehl, um den Steuerungsbefehle an den Roboter, die eingelesenen Daten der Playstation-Controller, die Roboterdaten und die berechneten Geschwindigkeitsinkremente aus den L2- und R2-Triggertasten zu übertragen. Anfangs werden einige Konstanten festgelegt, die für die Funktion `servoJ()` zum Bewegen des Roboters notwendig sind. Diese sind jedoch nur bedingt zutreffend, da z. B. die Beschleunigung und Geschwindigkeit nicht darüber festgelegt werden, sondern durch die übermittelten Inkremente der neuen Positionswerte innerhalb der Frequenz. Dennoch benötigt die aufgerufene `servoJ`-Funktion diese Werte; sie kann damit Echtzeitregelung von Gelenkpositionen am Roboter vornehmen. Bewegungsänderungen erfolgen, indem entweder in TCP-Koordinaten oder in Gelenkachsen in einem der sechs Freiheitsgrade, die die freie Bewegung des Roboters ermöglichen, beim Betätigen der Taste ein Geschwindigkeitsinkrement addiert oder subtrahiert wird. Da die `ServoJ`-Funktion nur Gelenkwinkel übermitteln kann, ist für die Bewegung im Koordinatensystem eine inverse Kinematik (wie in Kapitel 2.1.5 beschrieben) durchzuführen. Dazu wird das vom Hersteller in den Roboter integrierte kinematische Modell verwendet. Das hat den Vorteil, dass das Programm für alle Modelle von Universal Robots universell einsetzbar ist.

```

229     # moves the robot in joint-movement with the servoJ function
230     if roboter_speed['joint'] > 0: # if L2 is pressed
231         rtde_control.servoJ(joint_q, velocity, acceleration, dt, lookahead_t, gain)
232
233     # moves the robot in tcp-movement, with calculating the inverse kinematics and use the servoJ function
234     elif roboter_speed['tcp'] > 0: # if R2 is pressed
235         tcp_position_to_q = rtde_control.getInverseKinematics(actual_tcp_pose) # calculate the inverse kinematics
236         rtde_control.servoJ(tcp_position_to_q, velocity, acceleration, dt, lookahead_t, gain)

```

Abbildung 22: Programmbeispiel der Bewegungssteuerung mittels der `ServoJ` Funktion

Zudem kann angenommen werden, dass der Roboterhersteller aufgrund seiner langjährigen Erfahrung optimierte Modelle und Rechenverfahren für seinen eigenen Robotertypen besitzt. Mittels der Funktion `getInverseKinematics(actual_tcp_pose)` wird die TCP-Position in eine Gelenkposition konvertiert und anschließend über die `ServoJ`-Funktion als Bewegungsbefehl an den Roboter übergeben (siehe Abbildung 22). Dabei wird die Rechnung im Robotercontroller durchgeführt und das hinterlegte kinematische Modell genutzt. Da für die Inverse Kinematik immer mehr Lösungsmöglichkeiten existieren und diese nicht auswählen oder favorisieren werden können, sind unerwartete oder schnelle Achsbewegungen möglich. Der Hersteller gibt für die Umrechnung der

indirekten Kinematik einen Fehler von $\pm 1 \cdot 10^{-10}$ m an. Im Programm wird kontinuierlich geprüft, ob eine Taste gedrückt wird; anschließend wird die Gelenkposition oder die TCP-Position um ein Inkrement verändert und mit der entsprechenden Bewegungsart angefahren.

Alle beschriebenen Funktionen werden in derselben While-Loop dauerhaft aufgerufen und somit in einer definierten Datenrate ausgeführt (siehe Abbildung 23).

```
612 while True:
613
614     time_before_loop = perf_counter() # time variable before the loop
615
616     # if loop which get executed in a specific frequency/datarate
617     if time_before_loop - time_after_loop >= 1. / DATARATE:
618         real_datarate = 1. / (time_before_loop - time_after_loop) # real frequency
619
620         # main programm/functions
621         roboter_data = receive_roboter_data(RTDE_READ)
622         controller_data = receive_controller_data(DUALSENSE)
623         force_moment = calculate_force_moment(roboter_data)
624         roboter_speed = calculate_roboter_speed(DATARATE, controller_data)
625         haptic_feedback = calculate_haptic_feedback(force_moment, roboter_data, SELECTED_ROBOTYP)
626         controller_feedback(DUALSENSE, haptic_feedback)
627         roboter_control(RTDE_CONTROL, controller_data, roboter_data, roboter_speed)
628
629         # GUI update
630         window.update()
631         window.after_idle(update_gui_data) # update the gui with the function
632
633         time_after_loop = perf_counter() # time variable after the loop
```

Abbildung 23: while-Funktion der erstellten Software

4.3 Anwendung

In Kapitel 2.3 wurden der aktuelle Forschungsstand und unterschiedliche Anwendungsfälle für Robotersteuerungen mit haptischem Feedback dargestellt. Viele dieser Anwendungsfälle lassen sich als theoretisches Konstrukt auf dieses Mock-up übertragen und könnten in folgenden wissenschaftlichen Versuchen verifiziert werden.

Steuerungen mit haptischem Feedback sind ideal für Telerobotik, also der Steuerung von Robotern aus der Distanz. Der Controller bietet diese Möglichkeiten inhärent durch seine WLAN- und Bluetooth-Schnittstellen. Das Mock-up ist allgemein auch mit einem anderen Controller für die Telerobotik geeignet, da der Computer sich nur im gleichen Netzwerk wie der Roboter befinden muss. Eine Steuerung über große Distanz wäre einfach zu realisieren und für die typischen Einsatzgebiete der Teleoperation der Robotik anwendbar. Im Gegensatz zu konventionellen Steuerungen ergäbe sich durch die Komponente des haptischen Feedbacks sogar ein weiterer Mehrwert, da typische Industriesteuerungen dieses Feature nicht inkludieren. In Einsatzorten, die Gefahren für den Menschen bergen, werden Teleoperatoren eingesetzt oder deren Einsatz diskutiert. Hierzu zählen unter anderem Situationen, in denen der Roboterarm mit gefährlichen Materialien hantieren muss, wie z. B. heiße Werkstücke, radioaktive Elemente oder toxische Substanzen. Das Arbeiten in gefährlichen Umgebungen kann ebenfalls durch einen Teleoperator unterstützt werden. Der Einsatz von Roboterarmen in verstrahlten Gebieten mit einem fahrerlosen Transportsystem ist denkbar, aber auch in Katastrophengebieten, in denen noch eine aktive Gefahr für den Menschen besteht. Von der Bergung bis zur Unterstützung bei der Suche in abgelegenen Gebieten sind Anwendungsszenarien denkbar.

In Teilgebieten der Unterwasser- oder Weltraumforschung liegt ebenfalls hohes Potenzial für die Anwendung eines ferngesteuerten Roboterarms mit zusätzlicher Unterstützung durch das haptische Feedback. Wie schon im theoretischen Fundament aufgegriffen, existieren für die Weltraumforschung individuelle Einzellösungen, die an die gegebenen Bedingungen angepasst sind und als Beispiel für den Nutzen des haptischen Feedbacks dienen. In Bereichen der Weltraumforschung werden diese Systeme, z. B. telerobotische Raumsonden, be-

reits eingesetzt. Die Tiefsee bietet Anwendungsfelder in der Forschung, bei Reparaturen von Bohrinseln oder der Errichtung von Offshore-Windparks.

Weitere Einsatzgebiete sind der medizinische und militärische Bereich, wobei gerade der medizinische Bereich besondere Anforderungen aufweist, die mit dem aktuellen Mock-up nicht adressiert werden. Die Entwicklung könnte jedoch auch Anreize für dieses Gebiet liefern, da in der Chirurgie spätestens mit dem Da-Vinci-Chirurgiesystem das Thema der robotergestützten Arbeit in der Forschung und angewandten Medizin gegenwärtig ist. Weitere Unternehmen arbeiten an ähnlichen Systemen, wobei das haptische Feedback des Controllers für den Chirurgen zum Unterscheiden unterschiedlicher Gewebearbeiten eine der Schlüsselkomponenten ist.

Das Mock-up ist jedoch für keinen dieser Anwendungsfälle spezialisiert und würde für den produktiven Einsatz ein Re-Design für den jeweiligen spezifischen Einsatzzweck benötigen. Es könnte jedoch in den jeweiligen Bereichen als Denkanstoß und Ideengeber fungieren sowie eine Motivation sein, die bestehenden Systeme mit haptischem Feedback zu erweitern.

Ein weiterer schon oft angedeuteter Anwendungsfall könnte die Aus- und Weiterbildung bzw. die Gewinnung von neuen Fachkräften werden. Die Gamification, also die Übertragung von Spielkonzepten in die reale Arbeitswelt, könnte Technologiehemmnisse bei ungelernten Arbeitern überwinden, da Ängste vor komplizierten Systemen und Herausforderungen spielerisch genommen werden. Die Hürde, Neues auszuprobieren und sich auf unbekannte Technologien, Arbeitsprozesse und Anwendungen einzulassen, könnte für die Arbeiter durch etwas Bekanntes – den Gamecontroller – verringert werden. Die Arbeiter könnten ihre positiv gearteten Erfahrungen und Gedanken in Verbindung mit dem Controller im ersten Ansatz auf die Arbeitswelt übertragen.

Weiterhin könnten Schüler und Studenten leichter an die Technologie der Robotik herangeführt werden. Sie könnten motiviert werden, indem sie in ein Spiel gesetzt werden, in welchem sie mit dem Roboter mittels Gamecontroller produktive Aufgaben lösen. Durch die spielerische Komponente des Mock-ups könnte schon frühzeitig eine Begeisterung für die Ingenieurwissenschaften geweckt werden. Innere Widerstände der Schüler und Studenten gegenüber dem technologisch Ungewissen und Komplexen, wie es in der Robotertechnik existiert, können auf diese Weise überwunden werden. Aus diesem Grund wird das Mock-up Teil des Robotik-Technikums der Hochschule Mittweida als Technolo-

giedemonstrator von innovativen Robotikanwendungen. Mit einer Hands-on-Mentalität soll die Robotersteuerung mittels Gamecontroller hier von Studenten ausprobiert und erlebt werden.

4.4 Sicherheitsbetrachtung

Kollaborierende Roboter wurden für die direkte Zusammenarbeit mit dem Menschen entworfen. Sie gehören jedoch in die technische Sparte der Industrieroboter und müssen gewisse Standards einhalten, um zu gewährleisten, dass sie sicher mit dem Menschen zusammenarbeiten. Sie sind keine inhärent sicheren Maschinen, sondern werden in der Regel als unvollständige Maschine ausgeliefert. Erst anhand des konkreten Anwendungsfalls und des Robotersystems wird entschieden, ob der kollaborative Roboter auch im echten Arbeitsumfeld mit dem Menschen zusammenarbeiten darf. Die Mehrzahl der als Cobots vermarkteten Robotersysteme arbeiten anschließend gleichrangig wie Industrieroboter, d. h. hinter einem Sicherheitszaun.

Vor der Inbetriebnahme jeder Roboteranlage muss sichergestellt werden, dass sie der EG-Maschinenrichtlinie 2006/42/EG entspricht. Die Norm EN ISO 10218 Teil 1 und 2 konkretisiert die Sicherheitsanforderungen für Roboter. Es wird jedoch davon ausgegangen, dass der Gamecontroller als Steuerung nachträglich in ein bestehendes Robotersystem integriert wird, das durch das Konformitätsbewertungsverfahren als sicher bewertet wurde. Hier ist nun zu prüfen, ob es sich bei der Integration des Gamecontrollers um eine wesentliche Veränderung hinsichtlich der Anwendung der Maschinenrichtlinie auf umgebaute Maschinen handelt. Wenn diese Veränderungen wesentlich wären, müsste das gesamte Konformitätsverfahren erneut durchlaufen werden. Dabei ist die Änderung zunächst im Hinblick auf ihre sicherheitsrelevanten Auswirkungen zu untersuchen. Es ist also zu prüfen, ob sich neue Gefährdungen ergeben oder ob sich ein bereits vorhandenes Risiko erhöht. Sollte einer der beiden Fälle zutreffen, ist festzustellen, ob es die Möglichkeit gibt, die veränderte Maschine mit einfachen Schutzeinrichtungen in einen sicheren Zustand zu überführen.

Robotersysteme verfügen in der Regel über mindestens zwei Betriebsmodi mit unterschiedlichen Sicherheitsanforderungen. Diese sind zum einen ein Teach-in-Modus, in dem der Roboter mittels Handprogrammiergerät gesteuert wird und das Roboterprogramm erstellen werden kann, und zum anderen ein Auto-

matikmodus, bei dem das Programm (oft in hoher Geschwindigkeit) abgefahren wird. Im Teach-in-Modus steuert der Roboterbediener den Roboter bewusst in geringer Geschwindigkeit, weswegen hier andere (oft weniger restriktive) Sicherheitssysteme aktiv sind. In diesem Bereich der Roboteroperation ist das vorgestellte Mock-up angesiedelt. Oft sind dabei bestimmte Sicherheitssysteme (z. B. Lichtschranken oder Radarsensoren) deaktiviert, damit der Bediener nahe an den Roboter herantreten und präzise Einstellungen vornehmen kann. Um dennoch die Sicherheit des Bedieners zu gewährleisten, ist die Maximalgeschwindigkeit stark gedrosselt. Am Handprogrammiergerät befindet sich oft ein Totmannschalter als Zustimmungstaster, um festzustellen, ob ein Mensch anwesend und handlungsfähig ist, sowie ein Notausschalter nach Norm EN ISO 13850.

Der Gamecontroller ist so ausgelegt, dass die hinteren Triggertasten als Art Zustimmungstaster fungieren, sodass eine Roboterbewegung nur bei deren Betätigung ermöglicht wird. Somit müssen mindestens zwei Tasten gleichzeitig bedient werden, damit die Bewegung stattfindet. Ein Notausschalter nach Norm EN ISO 13850 ist nicht direkt am Controller, jedoch in unmittelbarer Nähe angebracht, sodass ein Halt im Notfall sofort ausgelöst werden kann. Zudem wurde die Taste X als softwareseitiger Not-Aus implementiert, der jedoch aufgrund der nötigen Anforderungen nur im Rahmen des Forschungsprojektes genutzt werden kann, jedoch nicht den Ansprüchen einer produktiven Umgebung entspricht. Die Bewegungen, die der Roboter durch die Software ausführen kann, sind im Verhältnis zu Bewegungen von industriellen Roboterapplikationen im Automatikmodus deutlich langsamer, sodass die Gefahr für den Benutzer als gering zu bewerten ist. Es entstehen weder neuen Gefährdungen, noch werden die existierenden Risiken durch den Demonstrator erhöht, weswegen es sich bei der Erweiterung des vorhandenen Robotersystems mit dem Mock-up in der betrachteten Situation nicht um eine wesentliche Änderung nach Produktsicherheitsgesetz handelt. Jedoch muss die Änderung dokumentiert und durch eine Gefährdungsanalyse vor dem Betrieb ergänzt werden. Eine Nutzung sollte nur von geschultem Personal erfolgen. Zu beachten ist, dass nach einer Änderung von Parametern in der Software eine neue Sicherheitslage vorliegen kann. Insgesamt ist das Mock-up aktuell ein Forschungs- und Teststand und kein robustes Produktivsystem, weshalb es immer mit Vorsicht von Experten verwendet werden sollte.

5 Diskussion

Im letzten Kapitel der Abhandlung werden die Forschungsergebnisse diskutiert. Die Forschungsfrage wird beantwortet und anschließend werden die Ergebnisse zusammengefasst und ein Ausblick in die zukünftige Forschung gegeben.

5.1 Beantwortung der Forschungsfrage

Ziel dieses Projekts war es, einen kollaborativen Roboterarm mit sechs Freiheitsgraden mittels Controller zu steuern, der ein haptisches Feedback an den Roboterbediener übermittelt. Der Cobot ist so konstruiert und ausgelegt, dass er ein Zusammenarbeiten mit dem Menschen ermöglicht. Mit Sensorik ausgestattet, kann der Roboter auf ihn einwirkende Kräfte und Momente erkennen und in Gefahrensituationen einen Not-Halt vollführen.

Die vom Roboter aufgenommenen Kräfte werden nun mithilfe eines Sony Dual-Sense-Gamecontrollers an den Roboterbediener übertragen, sodass dieser empfinden kann, was der Roboter durch seine Sensoren ertastet. Mit Python wurde eine Software mit Benutzeroberfläche für den Roboterbediener programmiert, um die Steuerung zu realisieren und das Konzept des haptischen Feedbacks umzusetzen. Somit wurde ein technisches Mock-up realisiert, das als Proof of Concept der Forschungsfrage dient. Die Machbarkeit eines Roboter-Steuerungssystems mit haptischem Feedback in kostengünstiger und jedem zugänglicher Weise wurde demonstriert. Dieses kann nun weiterentwickelt und genutzt werden.

Mit dem Mock-up wurde die technische Realisierung einer Roboter-Bedingung mit haptischem Feedback gezeigt. Es wurde gezeigt, dass die Robotersteuerung mit einem handelsüblichen Controller um weitere Funktionen erweitert und reales Livefeedback an den Roboterbediener übertragen werden kann. Konzepte, die für Computerspiele genutzt werden, um mittels neuer Technologie ein neues Spielerlebnis durch haptische Empfindungen zu schaffen, können für die Robotik adaptiert werden. Das technische Mock-up hat gezeigt, dass Möglich-

keiten existieren, um einen 6-Achs-Roboterarm feinfühler zu steuern und die Kräfte, die auf den Roboter wirken, für den Bediener erlebbar zu machen.

5.2 Diskussion der Ergebnisse

Die Erstellung des Mock-ups ist ein erster Schritt auf dem Forschungsgebiet des haptischen Feedbacks bei der Robotersteuerung, der mit einfachen Mitteln umgesetzt werden und adaptiert werden kann. Er dient als Grundlage, die Forschung voranzutreiben und Steuerungskonzepte zu entwerfen. Das Mock-up zeigt, dass es möglich ist, eine Robotersteuerung mit haptischem Feedback mit einfachsten Mitteln der Gebrauchstechnik umzusetzen. Das eröffnet Perspektiven einer nachfolgenden Nutzung von Robotersystemen in unterschiedlichen Anwendungsgebieten. Jedoch ist abzugrenzen, dass es sich um ein erstes Mock-up als Proof of Concept handelt, das weit entfernt von einem produktiven Einsatz und einem echten Mehrwert für Roboterprogrammierung ist. Es ist vielmehr eine erste Richtungsweisung und ein Ideenkonstrukt, um die Robotersteuerung neu zu denken und neue Potenziale auszuschöpfen.

5.3 Methodische Limitationen

Durch die Zeit- und Budgetlimitation wurden Entscheidungen getroffen, die eine realistische und qualitativ angemessene Lösung zur Beantwortung der Forschungsfrage ermöglichten. Mit einem höheren Zeit- und Kosteneinsatz hätte jedoch ein Mock-up geschaffen werden können, das eine detaillierte Darstellungen des haptischen Feedbacks ermöglicht. Von Versuchen mit Experten der Roboterbedienung, über Vergleichstest der Programmierung mittels Controller und konventionelle Steuerung, bis hin zu Messungen steht eine Bandbreite an wissenschaftlichen Methoden bereit, um die Eigenschaften des technischen Mock-ups zu evaluieren. Die Machbarkeitsstudie mittels Umsetzungsprojekt ist stets erst der Beginn der wissenschaftlichen Arbeit und schafft die Grundlage für weitere Beobachtungen, Tests und Versuche.

Bei der Umsetzung des haptischen Feedbacks erfolgte eine Beschränkung auf die Vibration und die haptischen Triggertasten. Wie im theoretischen Fundament dargestellt, existieren eine Vielzahl weiterer Elemente des Feedbacks, mit

denen z. B. eine klare Oberflächenbeschaffenheit erkannt werden kann. Hier liegt Potenzial, dem Roboterbediener mehr Informationen über die Haptik mitzuteilen und den Roboter erlebbar zu machen.

Mit einem eigens entwickelten individuellen Controller hätte die Steuerung des Roboters mehr Möglichkeiten erhalten und optimiert werden können. Im theoretischen Gedankengang entstand die Überlegung, von zwei Joysticks mit je drei Freiheitsgraden zum Steuern der sechs Achsen, so dass Translation und Rotation in ihren drei Freiheitsgraden durch je einen Stick gesteuert werden könnten. Das wäre eine Erweiterung des ABB Systems und wirkte als theoretisches Konzept sehr intuitiv.

Im anschließenden und finalen Test hätte das Mock-up in realen Anwendungen getestet werden müssen, wobei das Feedback von Roboterbedienern ein Mehrwert für das Mock-up gewesen wäre, um die Funktionen zu evaluieren und in späteren Softwareiterationen zu optimieren.

5.4 Schlussbetrachtung

Unter Berücksichtigung der gewonnenen Erkenntnisse wurde ersichtlich, dass alternative Robotersteuerungen mit haptischem Feedback heutzutage schnell und kostengünstig umzusetzen sind. Verbrauchselektronik aus anderen Anwendungsbereichen bietet günstige Optionen sowie ausreichend verbaute Technik und Sensorik, um in Forschung und Industrie Mehrwerte zu schaffen und Ideen auszutesten. Das Mock-up liefert interessante Erkenntnisse zu alternativen Steuerungstechniken von Robotern.

Weitere Untersuchungen wären jedoch notwendig, um das breite Themenfeld des haptischen Feedbacks bei Robotersteuerungen umfassend zu ergründen. Während der Forschungsarbeit ergaben sich Fragen, die durch weitere Studien untersucht werden könnten. Diese betrafen die Überlegung, ob die Hürden des Einstiegs in die Robotik hiermit gemindert und die Fachkräftegewinnung sowie die Aus- und Weiterbildung erleichtert werden können. Weiterhin sollte untersucht werden, ob ein effizienteres und schnelleres Roboterprogrammieren in der Produktion, insbesondere hinsichtlich der einfachen Individualisierung des Steuerungskonzeptes möglich werden könnte. Auf Grundlage der Ideen in Bezug auf die Anwendungsgebiete müssten angewandte Versuche oder Befragungen erfolgen, um den Nutzen des haptischen Feedbacks zu evaluieren.

Attachments

Tabelle 2: Eingabegeräte mit haptischem Feedback (1/4)




PRODUKTNAME	SONY DUALSENSE CONTROLLER	HAPTIX GLOVES	TOUCH VON 3D SYSTEMS
HERSTELLER	Sony	Shadowrobot	3D Systems
BESCHREIBUNG	<p>haptisches Feedback mittels dynamischer Vibrationen und adaptive Trigger-Tasten mit verschiedenen Stufen von Kraft</p>  <p>https://www.playstation.com/de-de/accessories/dualsense-wireless-controller/</p>	<p>Präzise Steuerung und haptisches Feedback, um realistische Berührungen, Kraftrückmeldung und präzises Motion-Tracking zu ermöglichen.</p>  <p>https://www.shadowrobot.com/telerobots/</p>	<p>Haptisches Eingabegerät der Mittelklasse, dass als Motorgerät taktilen Feedback geben kann. (Force feedback: 3 Degrees of Freedom x,y,z)</p>  <p>https://de.3dsystems.com/haptics-devices/touch</p>
PREIS	62.99€ (inkl. MwSt)	\$80000 USD + Shipping costs	NAN (eigene Schätzung durch Angebote in Online-Shops für gebrauchte Produkte: Neupreis ~3.000-8.000€)
SCHNITTSTELLE	Bluetooth 5.1; USB C	Windows C++ API	USB 2.0
FEEDBACK	Vibration, variablen Gegendruck der Triggertasten, Farb-/Lichtindikatoren	Kraftfeedback: 8 lbs per finger Tactile Feedback: stufenlos zwischen 0 und bis zu 2 mm Eindringtiefe; Vibrationsfeedback; Scherkräfte; 66 über 130 Positionen gegen die Haut des Benutzers betätigen	Force feedback: 3 Degrees of Freedom x,y,z
STEUERUNG	Handelsüblicher Controller mit zwei Joysticks	Handschuh mit Handgesten und -bewegungen	Stiftähnliches Eingabegerät mit 3 Freiheitsgraden

Tabelle 3: Eingabegeräte mit haptischem Feedback (2/4)




PRODUKTNAME	PHANTOM PREMIUM	REACTIVE GRIP MOTION CONTROLLER	STRATOS EXPLORE
HERSTELLER	3D Systems	Tactical Haptics	Ultraleap
BESCHREIBUNG	<p>Haptischen Eingabegeräte mit taktilem Feedback unterschiedlicher Art sowie verschiedene Bewegungsrichtungen und unterschiedliche Steifigkeit. (Force Feedback: bis zu 6 degrees of Freedom x, y, z, Tx, Ty, Tz)</p>  <p>https://de.3dsystems.com/haptics-devices/3d-systems-phantom-premium/features</p>	<p>Bewegungs- und Kraftinformationen werden mithilfe von taktilem Feedback vermitteln. Das Gerät verwendet hierzu Gleitplatten in der Oberfläche des Griiffs des Controllers, die die Scher- und Reibungskräfte imitieren.</p>  <p>https://tacticalhaptics.com/products/</p>	<p>Taktilen Feedback in der Luft durch Druckpunkte die durch Ultraschallwellen erzeugt werden.</p>  <p>https://www.ultraleap.com/product/stratos-explore/#overview</p>
PREIS	NAN (eigene Schätzung durch Angebote in Online-Shops für gebrauchte Produkte: Neupreis ~8.000-15.000€)	\$3,999 USD	2500,00€
SCHNITTSTELLE	Parallel-Port	NAN	USB type-C
FEEDBACK	Force Feedback: bis zu 6 degrees of Freedom x, y, z, Tx, Ty, Tz	taktilen Feedback mittels Gleitplatten (Scher- und Reibungskräfte)	haptische Mid-Air-Technologie (Ultraschall), um unterschiedliche Sinneseindrücke zu erschaffen
STEUERUNG	Stiftähnliches Eingabegerät mit 6 Freiheitsgraden	3-Achsen-Rotation in der freien Luft mit einer Vielzahl von Knöpfen und Schaltern	Handgesten und -bewegungen

Tabelle 4: Eingabegeräte mit haptischem Feedback (3/4)

PRODUKTNAME	CLS-E FORCE FEEDBACK JOYSTICK	FSSB R3 LIGHTING	FORCE-FEEDBACK JOYSTICK
HERSTELLER	Brunner	EMYCSA – RealSimulator	Sensodrive
BESCHREIBUNG	<p>Active Force Feedback Joystick konzipiert für die Flugsimulation.</p>  <p>https://www.brunner-innovation.swiss/produkt/cls-e-joystick/?lang=de</p>	<p>Joystick konzipiert für die Flightsimulation mit einstellbarem Feedback.</p>  <p>https://realsimulator.com/fssb-r3-lighting/</p>	<p>Drei aktive Freiheitsgrade mit Force-Feedback.</p>  <p>https://www.sensodrive.de/produkte-leistungen/force-feedback-joystick.php</p>
PREIS	1.249,00€	450,00€ +360,00€	25.000 € (Lieferzeit: mehrere Monate)
SCHNITTSTELLE	USB or CAN	USB	CAN-Schnittstelle (PEAK Library in Python)
FEEDBACK	Active Force Feedback	Active Force Feedback	Drehmoment, Dämpfung, Reibung, Federsteifigkeit
STEUERUNG	Drei-Achsen-Joystick	Drei-Achsen-Joystick	Drei-Achsen-Joystick (3DoF)

Tabelle 5: Eingabegeräte mit haptischem Feedback (4/4)

PRODUKTNAME	SPACEMOUSE	90JAM-JOYSTICK	HE1-32-R-A-5-BK
HERSTELLER	3dconnextin / megatron	ALTHEN BV	ALTHEN BV
BESCHREIBUNG	<p>Intuitive Steuerung komplexer 3D-Bewegungen mit 6 Freiheitsgrade (6DoF)</p>  <p>https://3dconnexion.com/de/spacemouse/</p>	<p>Drei-Achsen-Joystick mit Cobra-Knopf und 5 Schalter</p>  <p>(Angebot per E-Mail)</p>	<p>Kleiner Standardindustri joystick mit eingeschränkter Konfiguration.</p>  <p>(Angebot per E-Mail)</p>
PREIS	~160 € netto	€ 2400,- ex. + 12-14 Wochen Lieferzeit	€250,-netto (weniger Wochen Lieferzeit)
SCHNITTSTELLE	UART- (seriell) oder USB-Schnittstell	USB-Umwandlungsplatine	USB
FEEDBACK	Kein Forcefeedback nur Federrückstellung.	Vibration	keine Standardoptionen für haptisches Feedback
STEUERUNG	Steuerung komplexer 3D-Bewegungen mit 6 Freiheitsgraden	Drei-Achsen-Joystick mit Cobra-Knopf und 5 Schalter	Drei-Achsen-Joystick

Literaturverzeichnis

Aaltonen, Iina; Salmi, Timo (2019): Experiences and expectations of collaborative robots in industry and academia: barriers and development needs. In: *Procedia Manufacturing* 38, S. 1151–1158. DOI: 10.1016/j.promfg.2020.01.204.

Berg, Julia; Lottermoser, Albrecht; Richter, Christoph; Reinhart, Gunther (2019): Human-Robot-Interaction for mobile industrial robot teams. In: *Procedia CIRP* 79, S. 614–619. DOI: 10.1016/j.procir.2019.02.080.

Buxbaum, Hans-Jürgen (2020): Mensch-Roboter-Kollaboration. Wiesbaden, Heidelberg: Springer Gabler.

Colgate, J. E.; Wannasuphoprasit, Witaya; Peshkin, Michael (1996): Cobots: robots for collaboration with human operators. In: *ASME International Mechanical Engineering Congress and Exposition*, S. 433–439.

Decker, Michael (2011): Service-Roboter im Blick der Technikfolgenabschätzung. In: *TATuP-Zeitschrift für Technikfolgenabschätzung in Theorie und Praxis* 20.2, S. 76–79.

Denavit, J.; Hartenberg, R. S. (1955): A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices. In: *Journal of Applied Mechanics* 22 (2), S. 215–221. DOI: 10.1115/1.4011045.

Devol, George C., JR (1961): Programmed article transfer. Veröffentlichungsnr: Patent US2988237A.

DLR Institut für Robotik und Mechatronik: USB-Force-Feedback-Joystick. https://www.dlr.de/rm/desktopdefault.aspx/tabid-3805/6223_read-8990/.

Dzedzickis, Andrius; Subačiūtė-Žemaitienė, Jurga; Šutinys, Ernestas; Samukaitė-Bubnienė, Urtė; Bučinskas, Vytautas (2022): Advanced Applications of Industrial Robotics: New Trends and Possibilities. In: *Applied Sciences* 12 (1), S. 135. DOI: 10.3390/app12010135.

DIN EN ISO 9241-110:2020-10, 2020: Ergonomie der Mensch-System-Interaktion - Teil 110: Interaktionsprinzipien.

Forlizzi, Jodi; DiSalvo, Carl (2006): Service robots in the domestic environment. In: Michael A. Goodrich, Alan C. Schultz und David J. Bruemmer (Hg.): Proceeding of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction - HRI '06. Proceeding of the 1st ACM SIGCHI/SIGART conference. Salt Lake City, Utah, USA, 02.03.2006 - 03.03.2006. New York, New York, USA: ACM Press, S. 258.

Gao, Shuo; Yan, Shuo; Zhao, Hang; Nathan, Arokia (2021): Haptic Feedback. In: Shuo Gao, Shuo Yan, Hang Zhao und Arokia Nathan (Hg.): Touch-Based Human-Machine Interaction. Cham: Springer International Publishing, S. 91–108.

Haslwanter, Thomas (2021): Python. In: Thomas Haslwanter (Hg.): Hands-on Signal Analysis with Python. An Introduction. 1st ed. 2021. Cham: Springer International Publishing; Imprint Springer (Springer eBook Collection), S. 7–39.

Haun, Matthias (2013): Handbuch Robotik. Programmieren und Einsatz intelligenter Roboter. 2. Aufl. Berlin, Heidelberg: Springer Vieweg (VDI-Buch). Online verfügbar unter <http://swbplus.bsz-bw.de/bsz393865347cov.htm>.

Hertzberg, Joachim; Lingemann, Kai; Nüchter, Andreas (2012): Mobile Roboter. Berlin, Heidelberg: Springer Berlin Heidelberg.

Hesse, Stefan; Schnell, Gerhard (2014): Sensoren für die Prozess- und Fabrikautomation. Funktion - Ausführung - Anwendung. 6., korr. und verb. Aufl. Wiesbaden: Springer Vieweg.

Horn, Berthold K. P.; Hilden, Hugh M.; Negahdaripour, Shahriar (1988): Closed-form solution of absolute orientation using orthonormal matrices. In: *J. Opt. Soc. Am. A* 5 (7), S. 1127. DOI: 10.1364/JOSAA.5.001127.

Husty, Manfred; Karger, Adolf; Sachs, Hans; Steinhilper, Waldemar (1997): Kinematik und Robotik. Berlin, Heidelberg, New York: Springer.

DIN EN ISO 10218-1:2011, Januar 2012: Industrieroboter – Sicherheitsanforderungen – Teil 1: Roboter.

DIN EN ISO 10218-2:2011, Juni 2012: Industrieroboter – Sicherheitsanforderungen – Teil 2: Robotersysteme und Integration.

Ji, Wei; Wang, Lihui (2019): Industrial robotic machining: a review. In: *Int J Adv Manuf Technol* 103 (1-4), S. 1239–1255. DOI: 10.1007/s00170-019-03403-z.

Khurshid, Rebecca P.; Fitter, Naomi T.; Fedalei, Elizabeth A.; Kuchenbecker, Katherine J. (2017): Effects of Grip-Force, Contact, and Acceleration Feedback on a Teleoperated Pick-and-Place Task. In: *IEEE transactions on haptics* 10 (1), S. 40–53. DOI: 10.1109/TOH.2016.2573301.

Krath, Jeanine; Schürmann, Linda; Korflesch, Harald F.O. von (2021): Revealing the theoretical basis of gamification: A systematic review and analysis of theory in research on gamification, serious games and game-based learning. In: *Computers in Human Behavior* 125, S. 106963. DOI: 10.1016/j.chb.2021.106963.

Lieberman, Jeff; Breazeal, Cynthia (2007): Development of a Wearable Vibrotactile Feedback Suit for Accelerated Human Motor Learning. In: *Proceedings 2007 IEEE International Conference on Robotics and Automation*. IEEE International Conference on Robotics and Automation. Rome, Italy, 2007: IEEE, S. 4001–4006.

Linda Onnasch; Xenia Maier; Thomas Jürgensohn (2016): Mensch-Roboter-Interaktion - Eine Taxonomie für alle Anwendungsfälle.

Loomis; Lederman (1986): Tactual perception. K. R. Boff, L. Kaufman, & J. P. Thomas (Eds.). In: *Handbook of perception and human performance, Vol. 2. Cognitive processes and performance*, S. 1–41.

Love, Jonathan (2007): Human Factors. In: Jonathan Love (Hg.): *Process automation handbook. A guide to theory and practice*. Goldaming, London: Springer, S. 457–461.

Mareczek, Jörg (2020a): *Modellbildung von Kinematik und Dynamik*. Berlin, Heidelberg: Springer Vieweg (Lehrbuch, Band 1). Online verfügbar unter <http://www.springer.com/>.

Mareczek, Jörg (2020b): *Pfad- und Bahnplanung, Antriebsauslegung, Regelung*. Berlin, Heidelberg: Springer Vieweg (Lehrbuch, Band 2).

Martínez, Jonatan; García, Arturo S.; Martínez, Diego; Molina, José P.; González, Pascual (2011): Texture Recognition: Evaluating Force, Vibrotactile and Real Feedback. In: David Hutchison, Takeo Kanade, Josef Kittler, Jon M. Kleinberg, Friedemann Mattern, John C. Mitchell et al. (Hg.): *Human-Computer Interaction – INTERACT 2011*, Bd. 6949. Berlin, Heidelberg: Springer Berlin Heidelberg (Lecture Notes in Computer Science), S. 612–615.

McMahan, W.; Gewirtz, J.; Standish, D.; Martin, P.; Kunkel, J. A.; Lilavois, M. et al. (2011): Tool Contact Acceleration Feedback for Telerobotic Surgery. In: *IEEE transactions on haptics* 4 (3), S. 210–220. DOI: 10.1109/TOH.2011.31.

Nawrat, Zbigniew; Rohr, Kamil; Fürjes, Péter; Mucha, Lukasz; Lis, Krzysztof; Radó, János et al. (2016): Force Feedback Control System Dedicated for Robin Heart Surgical Robot. In: *Procedia Engineering* 168, S. 185–188. DOI: 10.1016/j.proeng.2016.11.213.

Nguyen, Vinh; Melkote, Shreyes N. (2019): Modeling of flange-mounted force sensor frequency response function for inverse filtering of forces in robotic milling. In: *Procedia Manufacturing* 34, S. 804–812. DOI: 10.1016/j.promfg.2019.06.203.

Norberto Pires, J. (2005): Robot-by-voice: experiments on commanding an industrial robot using the human voice. In: *Industrial Robot* 32 (6), S. 505–511. DOI: 10.1108/01439910510629244.

Okamura, Allison M. (2009): Haptic feedback in robot-assisted minimally invasive surgery. In: *Current opinion in urology* 19 (1), S. 102–107. DOI: 10.1097/MOU.0b013e32831a478c.

Pacchierotti, Claudio (2015): Cutaneous Haptic Feedback in Robotic Teleoperation. Cham: Springer International Publishing (Springer Series on Touch and Haptic Systems Ser, v.0). Online verfügbar unter <http://gbv.eblib.com/patron/FullRecord.aspx?p=4084477>.

Pielot, Martin; Boll, Susanne (2010): Tactile Wayfinder. Comparison of Tactile Waypoint Navigation with Commercial Pedestrian Navigation Systems. In: *OFFIS Institute for Information Technology, Germany*.

Plenk, Valentin (2019): Grundlagen der Automatisierungstechnik kompakt. Wiesbaden: Springer Vieweg (Lehrbuch). Online verfügbar unter <http://www.springer.com/>.

Pott, Andreas; Dietz, Thomas (2019): Industrielle Robotersysteme. Entscheidungswissen für die Planung und Umsetzung wirtschaftlicher Roboterlösungen. Wiesbaden: Springer Vieweg. Online verfügbar unter <http://www.springer.com/>.

Rahman, Reduanur; Rahman, Md Sajid; Bhuiyan, Jillor Rahman (2019): Joystick controlled industrial robotic system with robotic arm. In: 2019 IEEE Interna-

tional Conference on Robotics, Automation, Artificial-intelligence and Internet-of-Things (RAAICON). 2019 IEEE International Conference on Robotics, Automation, Artificial-intelligence and Internet-of-Things (RAAICON). Dhaka, Bangladesh: IEEE, S. 31–34.

Riecke, Cornelia; Artigas, Jordi; Balachandran, Ribin; Bayer, Ralph; Beyer, Alexander; Bernhard, Brunner et al. (2016): KONTUR-2 MISSION: THE DLR FORCE FEEDBACK JOYSTICK FOR SPACE TELEMANNIPULATION FROM THE ISS. The International Symposium on Artificial Intelligence. In: *Robotics and Automation in Space*. Online verfügbar unter <https://elib.dlr.de/105449/2/k2iSAIRAS.pdf>.

Roveda, Loris; Magni, Mauro; Cantoni, Martina; Piga, Dario; Bucca, Giuseppe (2021): Human–robot collaboration in sensorless assembly task learning enhanced by uncertainties adaptation via Bayesian Optimization. In: *Robotics and Autonomous Systems* 136, S. 103711. DOI: 10.1016/j.robot.2020.103711.

Sailer, Michael (2016): Die Wirkung von Gamification auf Motivation und Leistung. Empirische Studien im Kontext manueller Arbeitsprozesse. Wiesbaden: Springer Fachmedien Wiesbaden. Online verfügbar unter <http://gbv.ebib.com/patron/FullRecord.aspx?p=4527983>.

Scheggi, Stefano; Chinello, Francesco; Prattichizzo, Domenico (2012): Vibrotactile haptic feedback for human-robot interaction in leader-follower tasks. In: Fillia Makedon (Hg.): Proceedings of the 5th International Conference on Pervasive Technologies Related to Assistive Environments - PETRA '12. the 5th International Conference. Heraklion, Crete, Greece, 06.06.2012 - 08.06.2012. New York, New York, USA: ACM Press, S. 1.

Schoonmaker, Ryan E.; Cao, Caroline G.L. (2007): Vibrotactile force feedback system for minimally invasive surgical procedures. In: 2006 IEEE International Conference on Systems, Man and Cybernetics. Taipei, Taiwan, 8 - 11 October 2006. 2006 IEEE International Conference on Systems, Man and Cybernetics. Taipei, Taiwan, 10/8/2006 - 10/11/2006. Institute of Electrical and Electronics Engineers; IEEE Systems, Man, and Cybernetics Society. Piscataway, NJ: IEEE Service Center, S. 2464–2469.

SDU Robotics (Hg.): About SDU Robotics. Maersk Mc-Kinney Moller Institut. Online verfügbar unter <https://www.sdu.dk/en/forskning/sdurobotics/about>, zuletzt geprüft am 19.03.2022.

SDU Robotics (Hg.): Introduction. Maersk Mc-Kinney Moller Institut. Online verfügbar unter https://sdurobotics.gitlab.io/ur_rtde/introduction/introduction.html, zuletzt geprüft am 19.03.2022.

SDU Robotics (Hg.): ur_rtde. Maersk Mc-Kinney Moller Institut. Online verfügbar unter https://gitlab.com/sdurobotics/ur_rtde, zuletzt geprüft am 19.03.2022.

Seibold, Ulrich Stefan (2013): An Advanced Force Feedback Tool Design for Minimally Invasive Robotic Surgery. In: *Deutsches Zentrum für Luft- und Raumfahrt*.

Šekoranja, Bojan; Bašić, Denis; Švaco, Marko; Šuligoj, Filip; Jerbić, Bojan (2014): Human-Robot Interaction Based on Use of Capacitive Sensors. In: *Procedia Engineering* 69, S. 464–468. DOI: 10.1016/j.proeng.2014.03.013.

Siciliano, Bruno; Khatib, Oussama (Hg.) (2008): Springer handbook of robotics. With ... 84 tables. Berlin, Heidelberg: Springer.

Tlach, Vladimír; Kuric, Ivan; Kumičáková, Darina; Rengevič, Alexander (2017): Possibilities of a Robotic End of Arm Tooling Control within the Software Platform ROS. In: *Procedia Engineering* 192, S. 875–880. DOI: 10.1016/j.proeng.2017.06.151.

Universal Robots: UNSERE GESCHICHTE. Online verfügbar unter <https://www.universal-robots.com/de/uber-universal-robots/unsere-geschichte/>, zuletzt geprüft am 20.03.2022.

Wagner, Maximilian; Avdic, Dennis; Heß, Peter (2016): Gamepad Control for Industrial Robots - New Ideas for the Improvement of Existing Control Devices. In: Proceedings of the 13th International Conference on Informatics in Control, Automation and Robotics. 13th International Conference on Informatics in Control, Automation and Robotics. Lisbon, Portugal: SCITEPRESS - Science and Technology Publications, S. 368–373.

Weber, Wolfgang (2017): Industrieroboter. Methoden der Steuerung und Regelung. 3., neu bearbeitete Auflage. München: Fachbuchverlag Leipzig im Carl Hanser Verlag. Online verfügbar unter

http://files.hanser.de/hanser/docs/20170123_217123111-73_978-3-446-43355-7_Leseprobe.pdf.

Winkler, Alexander (2020): Vorlesungsmitschriften zur Lehrveranstaltung Robotik. (Version vom 15.12.2020). Hg. v. Hochschule Mittweida. Mittweida.

Xiao, Maya (2021): The Collaborative Robot Market 2021–28: Grounds for Optimism After A Turbulent Two Years. INTERACT ANALYSIS GROUP HOLDINGS LIMITED. Online verfügbar unter <https://www.interactanalysis.com/the-collaborative-robot-market-2021-28-grounds-for-optimism-after-a-turbulent-two-years/>, zuletzt geprüft am 20.03.2022.

Zangemeister, Christof (2014): Nutzwertanalyse in der Systemtechnik. Eine Methodik zur multidimensionalen Bewertung und Auswahl von Projektalternativen. Teilw. zugl.: Berlin, Univ., Diss., 1970. 5., erw. Aufl. Norderstedt: Books on Demand.

Zeichen, Gerfried; Fürst, Karl (2000): Automatisierte Industrieprozesse. Wien, New York: Springer.

Selbstständigkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und nur unter Verwendung der angegebenen Literatur und Hilfsmittel angefertigt habe.

Stellen, die wörtlich oder sinngemäß aus Quellen entnommen wurden, sind als solche kenntlich gemacht.

Diese Arbeit wurde in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegt.

Mittweida, den 30.04.2022

Felix Hillemeier