

Universität Kassel
Fachbereich Maschinenbau
Arbeits- und Organisationspsychologie

Bachelorarbeit
Mechatronik

**Intuitive Telepräsenzsteuerung eines anthropomorphen Manipulators
unter Verwendung von Head-Mounted Displays und Motion-Capturing**

Betreuer: Prof. Dr. phil. habil. Oliver Sträter
Dipl. Ing. Mehrach Saki, M.Sc.

Verfasser: Johannes Hölker
Matr.-Nr. 35192059
Kattenstraße 9
34119 Kassel
johannes.hoelker@uni-kassel.de

Kassel, den 20. März 2021

Inhaltsverzeichnis

Abstract	i
List of Figures	ii
1 Einleitung	1
1.1 Motivation	1
1.2 Zielstellung	2
1.3 Aufbau der Arbeit	3
2 Theorie	4
2.1 Telepräsenz	4
2.1.1 Immersion	5
2.1.2 Intuition	6
2.1.3 Anthropometrie und räumliche Gestaltung	7
2.2 Manipulatoren	10
2.2.1 Roboterarten	11
2.2.2 Manipulatorsteuerung	11
2.2.3 anthropomorphe Roboterkinematik	13
2.2.4 Stand der Technik	13
2.3 Fragestellung	14
3 Methoden	16
3.1 Hardware	16
3.1.1 Fuse Deposition Modeling	16
3.1.2 Amewi DC5821LV Standard RC Servo	16
3.1.3 Kodak Pixpro SP360 4K	17
3.1.4 HTC Vive Pro Eye	17
3.1.5 Eingabesysteme	18
3.2 Software	19
3.2.1 visuelle Übertragung	19
3.2.2 Robot Operating System	19
3.3 mathematische Grundlage für anthropomorphe Servosteuerung	20
4 Konzept	22
5 Umsetzung	24

5.1	Entwicklung des Manipulators	24
5.1.1	Gestaltung des Manipulators	25
5.1.2	Konstruktion der Bauteile	26
5.1.3	Elektronik	28
5.1.4	Manipulatorsteuerung	28
5.1.5	Montage des Manipulators	33
5.2	Bildübertragung	33
5.3	Telepräsenzsteuerung	35
5.3.1	Evaluation der Kameraposition	35
5.3.2	Einrichtung des gesamten Setups	37
6	Funktionsvalidierung	39
6.1	Aufgabenbeschreibung	39
6.2	Versuchsdurchführung	39
7	Diskussion und Ausblick	41
7.1	Auswertung	41
7.2	Limitationen	41
7.3	Ausblick	42
	References	43

Abstract

List of Figures

1	Gesichtsfeld, Blickfeld, Umblickfeld und Umblick-Gesichtsfeld des Menschen	8
2	Modell des menschlichen Arms mit Rotationsachsen und Gelenken	9
3	Greifraum des Menschen	10
4	Konzept für eine Telepräsenzsteuerung	22
5	kinematische Kette des Manipulators nach Denavit-Hartenberg Konvention	25
6	simulierte Kombination der Bauteile des Manipulators	27
7	Überschneidung des erreichbaren Arbeitsraums des Manipulator und des horizontalen Sehraums und technisch realisierten Sichtbereichs des Menschen	34
8	Überschneidung des Sichtbereichs der Kamera und Arbeitsbereichs des Manipulators	36
9	ausführende Umgebung einer Telepräsenzsteuerung mit Weitwinkelkamera, anthropomorphem Manipulator und Steuerkonsole	38

1 Einleitung

Zu Beginn dieser Arbeit wird die Motivation erläutert, aus welcher diese heraus geschrieben wird. Anschließend leitet sich daraus die Zielstellung ab und ein Überblick über die folgenden Kapitel wird präsentiert.

1.1 Motivation

Der erste ferngesteuerte Roboter, 1949 erfunden von Raymond Goertz, wurde dazu eingesetzt radioaktives Material gefahrenfrei zu bewegen. Die eingesetzte Methode des Master-Slave Manipulators übersetzte mechanisch die Bewegung des menschlichen Arms auf einen Manipulator. So konnten Gefahrgüter hinter einer durchsichtigen Trennwand bewegt werden. Dabei merkte Goertz an, dass die anwendende Person aus einem anderen Blickwinkel auf die auszuführende Aufgabe schauen muss und die Aufgabe so erschwert wird (Goertz, 1949).

Diese Problematik ist in der Robotik größtenteils durch eine Automatisierung der Manipulatoren gelöst worden. Durch ein hohes Maß an Sensorik und in kürzester Vergangenheit auch durch den Einsatz neuronaler Netze können sich Roboter weitgehend selbstständig bewegen (Siciliano & Khatib, 2008).

Es gibt jedoch weiterhin viele Aufgaben, bei denen ein Mensch anwesend sein muss. Die Wartung von industriellen Maschinen, Pflégetätigkeiten und kooperatives Arbeiten sind nur ein paar Beispiele. Viele Arbeitsumgebungen sind des Weiteren für den Menschen konzipiert und die Aufgaben sind komplex und damit nicht direkt automatisierbar (Fritsche, Unverzag, Peters, & Calandra, 2015). Um dennoch die Aufgaben lösen zu können, ohne dass ein Mensch anwesend ist, kann ein anthropomorpher ferngesteuerter Manipulator eingesetzt werden (Tanie, Society, & (Japan), 2003)(?, ?).

Herkömmliche Programmierverfahren erfordern des Weiteren viel Übung und Zeitaufwand. Die programmierende Person muss eine spezielle Ausbildung haben und braucht dennoch gewisse Zeit für die Umsetzung. Besonders in kleinen und mittelständischen Unternehmen bringen günstige Robotersys-

teme, welche einfach zu installieren, einzustellen, zu programmieren und zu warten sind, einen Vorteil. Um die genannten Arbeitsschritte zu vereinfachen erfordert es eine intuitive Steuerung des Roboters (Brogårdh, 2007, S.76) (Ehlers, 2019, S.190).

Räumlich in die Rolle des Roboters zu schlüpfen und aus der veränderten Perspektive heraus zu handeln eignet sich für eine Manipulatorsteuerung in für Menschen gefährlichen Umgebungen (Mareczek, 2020, S.9). Diverse Konzepte und Systeme um diese Problematik zu lösen wurden entwickelt und werden in ?? vorgestellt.

In der vorliegenden Arbeit wird demnach ein Arbeitsplatz gestaltet, welcher den Menschen mit Technik verbindet. Es werden Geräte eingesetzt, welche der Mensch tragen soll und so sein Umfeld beeinflussen. Für die gesamte Arbeitsgestaltung ist dabei wesentlich, dass nicht die technische Machbarkeit, sondern die "menschlichen Bedarfe und Möglichkeiten" im Vordergrund stehen (Sträter & Bengler, 2019). Für alle Entwicklungsschritte wird diese Richtlinie bedacht und evaluiert.

Aufgrund der aktuellen Situation des globalen Klimas (IPCC, 2021), soll die technische Entwicklung auch in Hinblick auf Nachhaltigkeit geschehen. Der Einsatz von Open Source Design (OSD) kann nach (Bonvoisin, 2016) während und zum Ende des Lebenszyklus eines technischen Produkts einen positiven Einfluss auf Nachhaltigkeit haben. Die Reparierbarkeit, Wiederverwendbarkeit und die lokal-gebundene Wertschöpfung werden durch Nutzung von OSD ermöglicht und haben damit eine erhöhte Effizienz bei der Ressourcennutzung.

1.2 Zielstellung

An die bereits bestehenden Konzepte soll diese Arbeit anknüpfen und die Entwicklung eines immersiven Systems zur Manipulatorsteuerung wird präsentiert. Die nutzende Person soll möglichst intuitiv einen Manipulator steuern können, ohne dass Vorerfahrung vorhanden ist. Dabei lautet das zentrale Ziel, die nutzende Person visuell in die Rolle des Roboters schlüpfen zu lassen.

Um dies zu erreichen wird mit den Technologien Motion-Capturing und Head-Mounted Displays (HMD) gearbeitet. Bei der Entwicklung des Manipulators wird soweit möglich mit OSD gearbeitet, in diesem Fall bedeutet dies den Einsatz von 3D-Druck und die Nutzung des Robot Operating Systems (ROS). Bei der gesamten Entwicklung soll des Weiteren ein auf den Menschen zugeschnittenes System entwickelt werden, wobei nicht die technische Machbarkeit im Vordergrund steht.

1.3 Aufbau der Arbeit

Nachdem nun die Hintergründe für die Entstehung dieser Arbeit erläutert wurden, werden zunächst die für das Verstehen der Arbeit nötigen wissenschaftlichen Erkenntnisse in Kapitel 2 zusammengefasst. Der Begriff und die Erreichung von Telepräsenz wird in 2.1 erläutert, sowie die Grundlagen für Manipulatoren in 2.2 gegeben

In Kapitel 3 werden die eingesetzten Methoden erklärt. Die der Umsetzung zugrundeliegende Konzeptidee wird in Kapitel 4 präsentiert und erläutert.

Anschließend wird auf das Vorgehen bei der Realisierung des Konzepts in Kapitel 5 eingegangen und aufgetretene Problematiken werden benannt.

Das entstandene System wird in Kapitel 6 auf die Funktion hin geprüft und validiert. Die benannten Problematiken und Schwierigkeiten bei der Umsetzung werden in Kapitel 7 diskutiert und ein Ausblick auf die zukünftige Forschung wird gegeben.

2 Theorie

Nachdem in das Thema eingeführt wurde, bildet dieses Kapitel die theoretische Grundlage für die vorliegende Arbeit. Der Begriff der und die Umsetzung von Telepräsenz werden erläutert. Die Anthropometrie im Hinblick auf den menschlichen Arm und Sehraum wird dargestellt und beschrieben. Anschließend wird auf die Technik eingegangen, indem die Robotik und im Speziellen die anthropomorphe Manipulatorsteuerung beschrieben wird. Aus dem vorgestellten Stand der Technik wird daraufhin die Forschungsfrage abgeleitet.

2.1 Telepräsenz

Für die vorliegende Arbeit hat der Begriff der Telepräsenz eine zentrale Bedeutung. (Mareczek, 2020, S. 9) definiert den Begriff wie folgt:

”Bei einem Telepräsenz- und Teleaktions-System soll der Bediener möglichst realitätsgetreu virtuell in eine entfernte Umgebung eintauchen. Hierzu wird ihm die entfernte Umgebung über alle Sinneskanäle vermittelt; der visuelle Kanal zum Beispiel mittels eines Stereo-Kamera-Paars in der entfernten Umgebung und einem binokularem Display in der lokalen Umgebung. [...] Im Idealfall kann der Bediener nicht mehr zwischen lokaler und entfernter Umgebung unterscheiden.”

Präsenz ist das erfolgreiche Gefühl ”da zu sein”, wohingegen Telepräsenz für Situationen reserviert ist, in denen eine Teleoperation durchgeführt wird (?). Bei einer Teleoperation bedient eine Person Werkzeuge in einer virtuellen Umgebung, welche die entfernte reale Umgebung wiedergibt (McMAHAN, 2003, S.77)

Die erfolgreiche teleaktive Bedienung eines Systems ist dabei abhängig vom Grad der Immersion. Ebenfalls für Telepräsenz wichtig ist die intuitive Bedienung. Die genannten Begriffe werden im Folgenden definiert und es werden Gestaltungsmaßnahmen gegeben, um ein hohes Maß an Immersion und Intuition zu erreichen.

An einem entfernten Ort telepräsent zu sein erfordert eine Instanz, die die menschliche Präsenz ersetzt. Die anthropomorphe Gestaltung dieser Instanz ist bei der Entwicklung einer immersiven VR-Umgebung wichtig, weshalb im Anschluss auf Anthropometrie eingegangen wird. Der Fokus liegt dabei auf dem menschlichen Sehraum als auch den Oberkörperproportionen und dabei in besonderem Maße auf dem menschlichen Arm.

2.1.1 Immersion

Das lateinische Wort "immersio" bedeutet "in etwas Eintauchen" und ist der Ursprung für den Begriff der Immersion. Die nach (McMAHAN, 2003, S.68) am meisten akzeptierte Definition von Immersion besagt, dass die nutzende Person mit Freude in eine simulierte Umgebung eintaucht und in dieser agieren und neue Fähigkeiten lernen kann:

"Immersion is a metaphorical term derived from the physical experience of being submerged in water. We seek the same feeling from a psychologically immersive experience that we do from a plunge in the ocean or swimming pool: the sensation of being surrounded by a completely other reality, as different as water is from air, that takes over all of our attention, our whole perceptual apparatus . . . in a participatory medium, immersion implies learning to swim, to do the things that the new environment makes possible" (Murray, 1997)

Die in das immersive Erlebnis eintauchende Person muss demnach lernen, sich in der neuen Umgebung zurechtzufinden, was je nach nutzender Person unterschiedlich gut funktioniert. Neben den technischen Voraussetzungen, wie weit das Erlebnis die nutzende Person auditiv und visuell umgibt, sind nach (McMAHAN, 2003, S.68) drei Aspekte wichtig, um ein möglichst hohes Maß an Immersion zu erreichen:

- die tatsächlich dargestellte Umgebung muss nahe an die Erwartungen der nutzenden Person rankommen
- die Aktionen der nutzenden Person müssen einen nicht-trivialen Effekt auf die Umgebung haben
- die Konventionen der dargestellten Welt müssen konsistent sein.

Dabei wird das Maß an Immersion in Beschäftigung, Vertiefung und totale Immersion eingeteilt. Welche der Stufen bei der jeweiligen Anwendung erreicht wird, hängt von den oben genannten Punkten ab und ist bei jeder nutzenden Person und Situation unterschiedlich. (Brown & Cairns, 2004) wandten diese Einteilung auf Videospieler an und weiteten die Einteilung auf andere Bereiche aus.

Um den Menschen möglichst weit in die virtuelle Umgebung einzubinden und so einen hohen Grad an Immersion zu erreichen, ist es nach (?, ?, S.434) wichtig die Darstellung visueller Umgebungsinformationen, insbesondere die Wiedergabe einer korrekten Perspektive, synchron zwischen der ausführenden und steuernden Umgebung zu gestalten. Er nennt außerdem die Entkopplung von der realen Umgebung als ausschlaggebenden Punkt um einen hohen Grad an Immersion zu erreichen. Die räumliche Wahrnehmung erfordert eine stereoskopische Darstellung um die Parallaxe des linken und rechten Auges zu simulieren.

Meist werden Head-Mounted Displays (HMD) eingesetzt, um ein hohes Level an Immersion zu erreichen (Aykut, 2019) (Krause & Strunz, 1997). In Kapitel ?? wird darauf näher eingegangen.

2.1.2 Intuition

(Mohs et al., 2006) bezieht Intuition ausschließlich auf Informationsverarbeitungsvorgänge des Menschen und definiert ein intuitiv nutzbares System als ohne Vorwissen effektiv bedienbar. Die Effektivität einer Interaktion ist dabei gemäß der Begriffsbestimmung nach der ISO-Normenserie 9241 bewertet. Wie genau und vollständig das vorgegebene Ziel erreicht wurde, legt fest, ob die Interaktion intuitiv war oder nicht. Dabei gilt, dass die Aufgabendurchführung nicht nach einem idealen Weg folgt sondern die nutzende Person den für sie passenden Weg findet.

Um ein System intuitiv zu gestalten, muss es selbsterklärend für die nutzende Person sein. Wenn nicht bekannte Kontexte mit vertrauten und erwartungskonformen Inhalten kombiniert werden, kann nach (Mohs, Nau-

mann, & Kindsmüller, 2007) bei der Gestaltung eines Systems eine intuitive Nutzung ermöglicht werden. Dies deckt sich mit der folgenden Definition von Intuition. Diese legt den Fokus auf die Nutzung von bereits bekanntem Wissen, welches eingesetzt wird, um Intuition bei neuen Interaktionen zu erzeugen:

”Intuitive use of products involves utilising knowledge gained through other experience(s). Therefore, products that people use intuitively are those with features they have encountered before. Intuitive interaction is fast and generally non-conscious, so people may be unable to explain how they made decisions during intuitive interaction.” (Blackler & Hurtienne, 2007, S.37)

Passend zu der Definition wurden drei wesentliche Prinzipien zur intuitiven Interaktion wurden präsentiert:

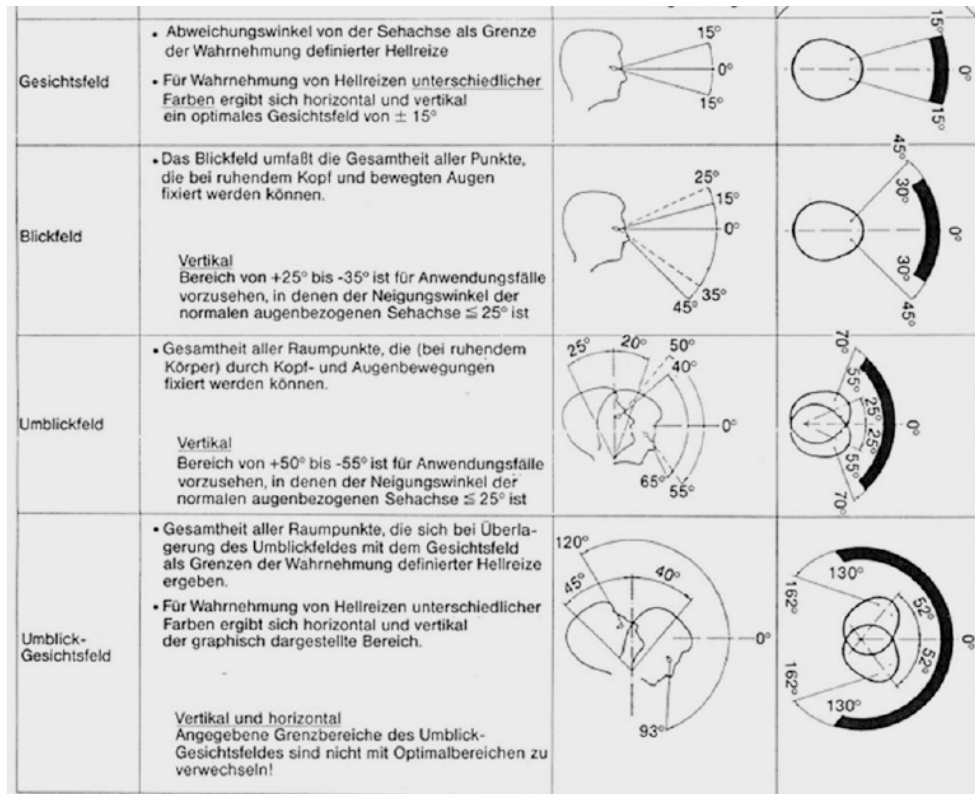
- Funktion, Platzierung und Erscheinen von neuen Funktionen sollen den bereits bekannten Funktionen möglichst ähneln.
- Die Funktionsbeschreibung unbekannter Features sollte mit Hilfe von bekannten Dingen beschrieben werden.
- Eine hohe Konsistenz von Funktion, Erscheinung und Ort unterschiedlicher Teile des Designs innerhalb der Oberfläche ist gewünscht.

Daraus lässt sich folgern, dass bei der Entwicklung von innovativen Produkten, welche nicht auf etablierten Konventionen beruhen, bereits bekannte Funktionen eingearbeitet werden sollten (Blackler & Hurtienne, 2007). Was bereits bekannt ist wird dabei größtenteils von den in der breiten Masse genutzten Produkten festgelegt.

2.1.3 Anthropometrie und räumliche Gestaltung

Der Begriff der Anthropometrie kommt aus dem Griechischen und bedeutet ”Lehre von den Maßen des menschlichen Körpers”. In der Arbeitswissenschaft wird zwischen Körpermaßen und Funktionsräumen des Menschen unterschieden. Um Gegenstände für die Allgemeinheit herzustellen, wird versucht allgemeine Größen zu finden. Aufgrund der großen Unterschiede zwischen Menschen können die Zahlenwerte jedoch abweichen (?, ?, S.480).

Abbildung 1: Gesichtsfeld, Blickfeld, Umblickfeld und Umblick-Gesichtsfeld des Menschen

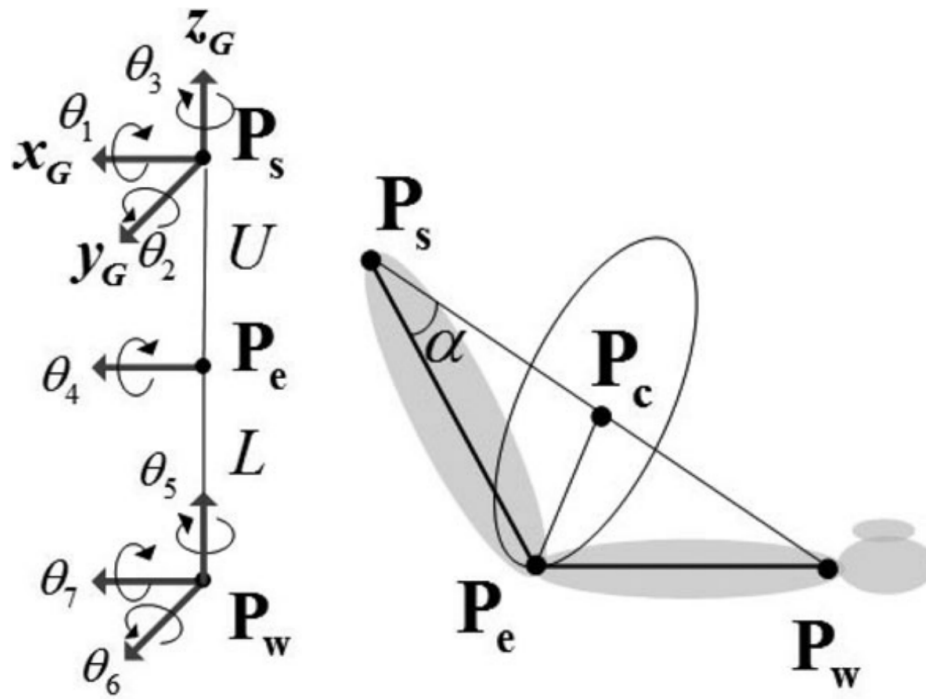


menschlicher Sehraum

Der menschliche Sehraum ist nach (Landau, 1992) Abbildung 1 eingeteilt in die Bereiche Gesichtsfeld, Blickfeld, Umblickfeld und das Umblick-Gesichtsfeld. Das Gesichtsfeld ist dabei als der Bereich des scharfen Sehens bei ruhenden Augen zu verstehen, das Blickfeld ist der bei bewegten Augen zu sehende Bereich und das Umblick-Gesichtsfeld schließt die Kopfbewegung mit ein und beschreibt den gesamten sichtbaren Bereich. Dabei ist die Bewegung des Oberkörpers nicht berücksichtigt.

Das Umblick-Gesichtsfeld beträgt bei horizontaler Bewegung 130° jeweils nach links und rechts und bei vertikaler Bewegung 120° mit nach hinten geneigtem Kopf und 93° bei einem nach vorne geneigtem Kopf. Dabei ist zu beachten, dass die angegebenen Bereiche nicht als Optimalbereich

Abbildung 2: Modell des menschlichen Arms mit Rotationsachsen und Gelenken



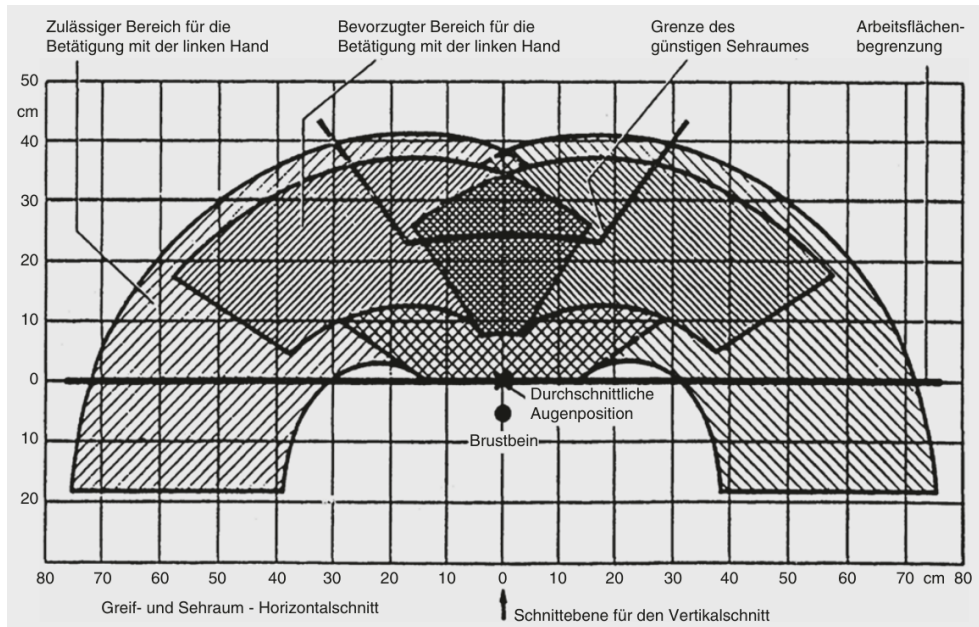
für technische Anwendungen zu betrachten sind.

Kinematik des menschlichen Arms

Ein Modell des menschlichen Arms mit allen Freiheitsgraden wurde von (Hyunchul Kim, Miller, Byl, Abrams, & Rosen, 2012) erstellt Abbildung 2. Demnach hat der menschliche Arm 7 Freiheitsgrade. Davon sind drei im Schultergelenk, einer im Ellbogen und wiederum drei im Handgelenk.

Angenommen die Blickrichtung in der Abbildung ist in x-Richtung und der Arm ist herabhängend, hat das Schultergelenk (P_s) neben der Rotation um den Oberarm (θ_3) die beiden Freiheitsgrade θ_1 und θ_2 . Diese beiden resultierten in ein Kugelgelenk, welches durch Quaternionen dargestellt werden kann (Saki, 2020). Das Ellbogengelenk (P_e) rotiert um die Achse des Ellbogens (θ_4) und die Rotation um den Vorderarm beschreibt θ_5 . Die Flexion (θ_6) und Abduktion (θ_7) des Handgelenks (P_w) bilden das Ende der kinematischen Kette.

Abbildung 3: Greifraum des Menschen



Der menschliche Arm kann damit einen bestimmten Bereich um den Oberkörper herum erreichen. Dieser Bereich wird als Greifraum bezeichnet und ist in Abbildung 3 dargestellt. Erkennbar ist, dass der zulässige Bereich für die jeweilige Hand hinter der Arbeitsflächenbegrenzung liegt und stark von dem bevorzugten Bereich für Betätigungen abweicht. Die durchschnittliche Augenposition liegt räumlich vor dem Brustbein. (?, ?)

2.2 Manipulatoren

Nachdem auf die menschlichen Aspekte bei Mensch-Roboter-Systemen eingegangen wurde, behandelt dieses Kapitel den Stand der Technik in der Robotik. Zunächst wird auf die unterschiedlichen Roboterarten eingegangen, die Formen der Manipulatorsteuerung werden unterschieden und anschließend wird aus Relevanz zu dem vorliegenden Thema anthropomorphe Roboterkinematik behandelt.

Für die Entwicklung von Robotersystemen sind unterschiedliche Gestaltungsrichtlinien zu beachten. Die grundlegenden drei Robotergesetze

wurden 1940 von Asimov formuliert und besagen, dass der Roboter exakt das tut was von ihm gewünscht wird und seine Möglichkeiten für die Erledigung der Aufgabe vollkommen ausschöpft, ohne dabei Menschen zu verletzen oder sich selber zu zerstören (Siciliano & Khatib, 2008). Diese Richtlinien müssen in der Entwicklung berücksichtigt werden.

2.2.1 Roboterarten

Die in dieser Arbeit genutzte Definition von Robotik leitet sich aus dem Wort "robota" ab, welches in dem 1920 erschienen Drama von Karel Čapek vorkommt und die Grundzüge eines Roboters definiert. Demnach verrichten Roboter untergeordnete Arbeit für den Menschen (Siciliano & Khatib, 2008, S.1) (Mareczek, 2020, S.1).

Es gibt eine große Zahl an solchen untergeordneten Arbeitsaufgaben, die automatisiert erledigt werden sollen. Dazu gehören unter anderem Fertigungsaufgaben in der Industrie, Haushaltsaufgaben für den Heimgebrauch und Transportaufgaben. Die Anwendungszwecke sind dabei breit gefächert und werden laufend erweitert. Für die genannten Anwendungen gibt es dazu passende Roboter, welche die für sie bestimmte Aufgabe bestmöglich ausführen sollen.

Der in der Einleitung erwähnte Roboter von Goertz benutzte das Master-Slave Prinzip und konnte auf kurze Distanzen das "Manipulieren gefährlicher Gegenstände" (Mareczek, 2020, S.10) bewerkstelligen. Diese Form eines Roboters wird deshalb auch Manipulator genannt. Der Manipulator kann als Oberbegriff für Roboterarme angesehen werden und unterscheidet sich gegenüber Industrierobotern durch den fehlenden automatischen Betrieb und die direkte Steuerung durch einen Menschen (Mareczek, 2020, S.11).

2.2.2 Manipulatorsteuerung

Um einen Manipulator zu steuern kommen unterschiedliche Methoden in Frage. Beim weit verbreiteten Teach-In wird der Roboterarm händisch bewegt, die Trajektorien aufgezeichnet und anschließend abgespielt (Ehlers,

2019, S.190).

Befindet sich zwischen der Steuerungs- und anwendenden Umgebung eine Barriere sodass die steuernde Person physisch nicht in die ausführende Umgebung gelangen kann spricht man von Teleoperation (Siciliano & Khatib, 2008, S.741).

In den meisten Anwendungsfällen der Teleoperation wird das Werkzeug am Ende der kinematischen Kette, der sogenannte Endeffektor (Mareczek, 2020, S.8), direkt gesteuert und seine gewünschte Position wird bestimmt. Dies geschieht meist über einen Joystick oder mithilfe von haptischen Eingabegeräten (Ehlers, 2019, S.190). Aus der jeweiligen Position des Endeffektors ergeben sich die einzelnen Gelenkstellungen, was in mehrere Lösungen für dieselbe Position resultiert. Dazu wird auf der ausführenden Seite eine Steuerung benötigt, welche die nötigen Gelenkstellungen berechnet. Diese Form der Teleoperation wird überwachte Steuerung genannt. Das gegenteilige Extrem ist die direkte Steuerung, welches im Folgenden beschrieben wird. Wenn die Berechnung auf beide Seiten aufgeteilt wird, spricht man von geteilter Steuerung (Siciliano & Khatib, 2008, S.746).

Bei der direkten Steuerung wird die gesamte Bewegung des Slaves vom Master kontrolliert und es befindet sich keine autonome Berechnung in der Remote-Umgebung (Siciliano & Khatib, 2008, S.746). Dabei wird jedes Gelenk einzeln und unabhängig voneinander als Einzeleingang-Einzelausgangs (SISO) Komponente bedient. Dies bringt einige Vorteile. Die Kommunikation zwischen den einzelnen Gelenken wird vermieden, die Bewegung ist skalierbar und es können günstigere Bauteile verwendet werden. (Siciliano & Khatib, 2008, S.137)

Die direkte Steuerung kann dabei von dem menschlichen Arm übernommen werden. Im Idealfall bewegt sich der Slave, also der Roboterarm, simultan zum Master, dem menschlichen Arm (Mareczek, 2020, S.11).

Bei einer solchen Remote-Steuerung ist nach (Tanie et al., 2003) vorteilhaft, den Manipulator anthropomorph zu gestalten. Ebenso nennt er zwei weitere Gründe für die Nachahmung der menschlichen Gestalt bei der Konstruktion

eines Roboters. Anthropomorphe Roboter lösen bei mit ihnen kollaborierenden Menschen Emotionen aus und können außerdem Geräte bedienen, welche für den Menschen konstruiert worden sind. Jedoch ist es in den meisten Fällen nachteilig, die menschliche Gestalt nachzuahmen und es genügt die für die Aufgabe nötigen Funktionen zu realisieren (Tanie et al., 2003). Die Nachahmung kann dabei sowohl optisch als auch kinematisch in unterschiedlichem Maß umgesetzt werden. Im folgenden Kapitel wird dazu auf die Grundlagen der Roboterkinematik eingegangen.

2.2.3 anthropomorphe Roboterkinematik

Ein kinematisches Gelenk ist eine Verbindung zwischen zwei Körpern, welches die relative Bewegung zueinander einschränkt. Dabei wird angenommen, dass die Körper komplett starr sind. Ein Roboter besteht dementsprechend aus mehreren Körpern und kinematischen Gelenken und bildet eine kinematische Kette (Siciliano & Khatib, 2008, S.9). Üblicherweise wird die Bestimmung des Endeffektorkoordinatensystems nach (?, ?) (DH-Konvention) vorgenommen. Pro Gelenk ist nach der DH-Konvention nur eine Bewegungsachse vorgesehen (Mareczek, 2020, S.42).

(?, ?, S.39) empfiehlt, für die Steuerung von anthropomorphen robotischen Systemen Quaternionen zu verwenden, da diese im Gegensatz zu Eulerwinkeln bei jeder Berechnung eine Lösung ergeben.

Um nun einen Roboter anthropomorph zu gestalten wird die in Kapitel ?? erwähnte Kinematik des menschlichen Arms benutzt um die kinematische Kette des Roboterarms zu gestalten. Um eine menschliche Bewegung zu vollführen müssen mindestens 6 DoFs vorhanden sein. Ein weiterer ist nötig um den Endeffektor zu bedienen (Goertz, 1949).

2.2.4 Stand der Technik

Das in 1.2 beschriebene Ziel wurde auf verschiedene Arten und mithilfe von unterschiedlichen Methoden versucht zu erreichen. Dabei wurde der Fokus jedes Mal anders gesetzt.

Eine Manipulatorsteuerung von Mit dem Aufkommen von Virtual Reality Brillen, konnte der Wunsch von Telepräsenzsteuerungen umgesetzt werden. (Fritsche et al., 2015) stellten ein System mit einer VR-Brille und MotionTracking mithilfe einer Kinect Kamera vor. Es wurde ein Roboter namens iCub gesteuert, welcher eine Miniaturausgabe eines Menschen darstellt. Dabei besitzt der Arm des iCub nur vier Freiheitsgrade und es traten bei der visuellen Übertragung eine Zeitverzögerung von 0,1 s und bei der Steuerung des Roboters von 0,8 s auf.

(Park, Jung, & Bae, 2017) nutzten ebenfalls Beschleunigungssensoren um einen 7DoF Manipulator zu steuern. Dazu verwendeten sie eine VR-Brille, welche die Bewegung des Roboterkopfes steuert um die Kamera zu bewegen.

2.3 Fragestellung

Die Programmierung und Steuerung von Manipulatoren geschieht zur Zeit über ein Umdenken der anwendenden Person indem sie von einer anderen Perspektive auf das System schaut und es so steuert. Das kann Komplikationen erzeugen, erfordert Einarbeitung und Erfahrung und macht die Steuerung schwierig (Ehlers, 2019, S.190). Die Programmierung aus der Ferne ist dementsprechend aufwendig, da sie mit Hilfe von Kameras aus verschiedenen Blickwinkeln erfolgen muss.

Indem die anwendende Person ihren Arm bewegen und einen Manipulator so direkt steuern kann, werden bekannte Abläufe beibehalten. Dadurch kann nach (Blackler & Hurtienne, 2007) Intuition erreicht werden. Für die direkte und unabhängige Steuerung ist wie in Kapitel 2.2.2 die anthropomorphe Gestaltung des Manipulators hilfreich. Die kinematischen Ketten von den in 2.2.4 gezeigten Manipulatoren sind jedoch nicht direkt anthropomorph, da sie nicht alle Freiheitsgrade des menschlichen Arms besitzen.

Die Entwicklung eines anthropomorphen Manipulators, welcher die in Kapitel 2.1.3 genannten Freiheitsgrade besitzt, ist daher ein Ziel dieser Arbeit. Ein System, welches die Technologien der Robotik und Datenbrillen kombiniert, kann helfen die genannte Problematik des Umdenkens bei

der Programmierung zu lösen. Durch den Einsatz von HMDs soll die anwendende Person die Perspektive des Roboters einnehmen und den Manipulator intuitiv steuern können.

Die in dieser Arbeit behandelte Forschungsfrage lautet demnach: Wie lässt sich ein System konstruieren, welches immersive Telepräsenz bei der Steuerung eines anthropomorphen Manipulators erreicht und dabei ohne Vorwissen bedienbar ist?

Um diese Frage zu beantworten wird zunächst auf die verwendete Methodik in 3 eingegangen. Anschließend werden die unterschiedlichen Geräte und Programme in 4 in Zusammenhang gebracht, sodass ein Konzept entsteht, welches in 5 umgesetzt wird. Um die Beantwortung der Forschungsfrage messbar zu machen wird anschließend eine Funktionsvalidierung durchgeführt, welche das Kapitel 6 beschreibt. Die abschließende Beantwortung und Diskussion der Forschungsfrage geschieht in 7.

3 Methoden

Um das Ziel der intuitiven Telepräsenzsteuerung zu erreichen werden verschiedene Technologien und Programme verwendet, welche dieses Kapitel behandelt. Das Kapitel wird dabei in Hardware und Software unterteilt. Auf Seiten der Hardware sind für den Bau des Manipulators 3D-Druck und Servomotoren relevant. Für die Bildübertragung ist für Aufnahme und Wiedergabe jeweils ein Gerät nötig. Außerdem werden Eingabesystem in virtuellen Umgebungen benötigt. Die Software wird unterteilt in visuelle Übertragung und die Kommunikation mit Robotern.

3.1 Hardware

3.1.1 Fuse Deposition Modeling

Fuse Deposition Modeling (FDM) ist eine weit verbreitete 3d-Druck-Technologie um Bauteile schnell, individuell und kostengünstig herzustellen. Mit dem Filament Polylactic Acid (PLA) ist der 3D-Druck einfach durchführbar und bietet eine gute Möglichkeit Prototypen herzustellen. Das thermoplastische Filament ist des Weiteren recycle- und kompostierbar (?, ?).

Für die Gestaltung von 3D-Druck Bauteilen sind zur Vereinfachung des Drucks und zur Erhöhung der Stabilität die folgenden Hinweise hilfreich. Um einen späteren Bruch zwischen zwei Schichten zu verhindern sollten besonders belastete Stellen horizontal gedruckt werden. Weiterhin ist ein 3D-Druck mit Überhang zu verhindern, da diese nur mit Hilfe von Stützmaterial möglich sind.

3.1.2 Amewi DC5821LV Standard RC Servo

Modellbau Servomotoren besitzen eine leichte und kompakte Bauweise und werden zweiseitig auf einer Unterlage festgeschraubt. Sie können Winkel zwischen 0° und 180° anfahren, indem sie mit einem pulsweitenmodulierten (PWM) Signal gesteuert werden. Die Länge der Pulse gibt den gewünschten Winkel an, wobei meist ein 50 Hz- Signal benutzt wird. Die Spannungsebene

liegt zwischen 4,8V und 7,4V und bei einer angelegten Spannung von 6V erzeugt der verwendete RC Servo ein Stellmoment von 214Ncm. Er besitzt ein Metallgetriebe und ist kugellagert.

3.1.3 Kodak Pixpro SP360 4K

Die Pixpro SP360 4K der Firma Kodak ist eine Fisheye Weitwinkelkamera, welche mit 60fps 1440x1440 Videos aufnehmen kann. Mit der sphärischen 360° Linse ermöglicht sie ein 235° großes Sichtfeld. Sie bietet unter anderem die Möglichkeit über Mikro-HDMI an einen Computer angeschlossen zu werden und so eine Liveübertragung des aufgenommenen Bildes zu ermöglichen.

Weitwinkelkameras haben eine nach außen gewölbte [Linse, welche es ermöglicht, einen großen Bereich aus einem einzigen Blickwinkel aufzunehmen. Dabei tritt in dem entstehenden Bild eine radiale Verzerrung auf. Um einen zentralen Bildpunkt herum verschieben sich die Bildpunkte nach innen oder außen. Verschieben sich die Bildpunkte nach außen, spricht man von "barrel distortion" (Aykut, 2019, S.15). Diese tritt bei Fish-Eye Kameras und somit auch bei der Pixpro auf. Der zentrale Bildpunkt verschiebt sich dabei nicht und bildet den passenden mittigen Pixel des realen Bildes ab (Weng, Cohen, & Herniou, 1992, S.968).

3.1.4 HTC Vive Pro Eye

Wie in Kapitel 2.1.1 erwähnt, werden binokulare Displays genutzt um visuell Immersion zu erzeugen. Die Displays werden so vor den Augen angebracht, dass die tragende Person ihre Umgebung ausblenden und dabei immer noch den Kopf bewegen kann. Das Bild auf den Displays ist dabei mit Hilfe von Umgebungssensoren an die Bewegung des Kopfes gekoppelt. Bei Bewegung passt sich das virtuelle Bild der realen Kopfposition an, sodass ein immersives Erlebnis erreicht wird.

Die HTC Vive Pro Eye ist ein Head Mounted Display (HMD) mit integrierten Umgebungssensoren. Diese erlauben der nutzenden Person in

eine virtuelle Welt einzutauchen, weshalb die Brille in die Kategorie der Virtual-Reality Brillen (VR-Brillen) fällt. Das Sichtfeld beträgt 110°, die Aktualisierungsrate 90 Hz und die Auflösung der zwei 3,5 Zoll Displays 1440x1600 Pixel pro Auge. Des Weiteren besitzt die Brille Audioausgaben für beide Ohren. Sie wird über einen speziellen Adapter an den Computer angeschlossen und kommuniziert mit diesem über SteamVR (siehe 3.2.1).

Bei der Verwendung von HMDs muss auf die Nutzungsdauer und die richtige Anwendung geachtet werden, um eine Motion Sickness vorzubeugen (Aykut, 2019). Das Auftreten dieser Bewegungskrankheit führt zum Ende des Telepräsenzerlebnisses und muss verhindert werden. Aus diesem Grund muss bei dem geringsten Schwindelgefühl die Nutzung des Systems gestoppt werden. Dies muss der nutzenden Person im Vorhinein mitgeteilt werden.

3.1.5 Eingabesysteme

(?, ?) unterscheidet die Eingabetechnologien in virtuellen Umgebungen in Tracking-Systeme und 3D-Eingabegeräte. 3D-Eingabegeräte dienen meist zur Bedienung in Menüs und sind entweder durch physische Bedienelemente oder Datenhandschuhe realisiert. Tracking-Systeme werden zur Bestimmung der räumlichen Orientierung und Position einzelner Körperteile genutzt. So kann ein simuliertes Abbild des Menschen aufgenommen und verarbeitet werden. Zur Umsetzung kommen elektromagnetische, optische, mechanische, sowie Ultraschall und Beschleunigungssensoren in Frage.

Inertiale Messsysteme nutzen die genannten Beschleunigungssensoren und sind weit verbreitet, als Gesamtsystem für den gesamten Körper verfügbar und daher einfach anzuwenden. Neuronmocap ist eine dieser Firmen und stellt mit dem Programm Axis Neuron ebenfalls die Software zur Verarbeitung der aufgenommenen Daten zu Verfügung. Die Inertial-Tracker (Inertial Measuring Unit, IMU) werden an jedem relativ zueinander beweglichen Körperteil angebracht und messen die Winkelbeschleunigung. Über Koppelverfahren kann so die Position ermittelt werden.

3.2 Software

3.2.1 visuelle Übertragung

StereoStitch Live

Für die Echtzeit Generierung von 360° Videos in 60fps empfiehlt (Lee, Um, Lim, Seo, & Gwak, 2021) das Programm StereoStitch der Firma Dermandar. Das Programm bietet die Möglichkeit Ein- und Ausgabegeräte frei zu wählen, sowie die Ausrichtung der Kamera zu verändern. Die direkte Darstellung auf einer VR-Brille bietet das Program ebenso an. Dazu startet es direkt SteamVR.

Um das Bild auf den Displays ohne die in 3.1.3 beschriebene fehlerhafte radiale Verzerrung darzustellen wird die gleichrechteckige Projektion angewandt. Dazu wird das Bild sozusagen von innen auf eine Kugel projiziert. Die nutzende Person soll sich demnach bei der Darstellung im Mittelpunkt dieser Kugel befinden. Dabei wird die Kugel in Kreise von Breitengraden (horizontale Kreise) unterteilt, welche dann auf horizontale Linien auf die 2D-Bildebene projiziert werden. Die nötigen Berechnungen werden von dem Programm Stereostitch Live übernommen.

SteamVR

SteamVR ist eine weitverbreitete VR-Menu Umgebung, welche das Starten und Organisieren von Anwendungen, sowie das Einrichten der virtuellen Umgebung innerhalb einer VR-Brille ermöglicht. Es ist mit nahezu allen VR-Brillen kompatibel. So auch mit der zuvor genannten HTC Vive.

3.2.2 Robot Operating System

Um auf die unterschiedlichen Anwendungen und Anforderungen in der Produktion einzugehen, empfiehlt (Brogårdh, 2007) ein modulares Roboterprogramm. Auf der Suche nach einem Framework, welches eine für Robotersteuerungen optimierte Umgebung bietet und beliebig erweiterbar ist, zeigte sich das von (Quigley et al., 2009) vorgestellte Robot Operat-

ing System (ROS) als am weitesten verbreitet. ROS ermöglicht sowohl ein Zusammenspiel der unterschiedlichen Plattformen (SteamVR, UDP-Listener, Arduino-Server), als auch die Übertragung zwischen mehreren Akteuren in einem gemeinsamen Netzwerk. Dies ermöglicht beliebige Erweiterungen des Systems. Des Weiteren unterstützt ROS viele Programmiersprachen und ist als OpenSource-Software frei verfügbar und erweiterbar.

Die grundlegende Struktur eines ROS-Netzwerks basiert auf der Peer-to-Peer Technologie, sodass alle Akteure unmittelbar miteinander kommunizieren können. Den zentralen Knoten bildet der ROS-Core, welcher das ROS-Netzwerk initiiert. Daraufhin können alle weiteren Prozesse gestartet werden, welche untereinander kommunizieren können.

Das Kommunikationsprotokoll regelt die Übertragung von Nachrichten, indem ROS-Topics erstellt werden. ROS-Topics stehen global zur Verfügung und können von ROS-Nodes abgerufen (subscribe) und verändert (publish) werden. Sie basieren auf TCP/IP oder UDP und können mit Hilfe der ROS-Messages beliebig aufgebaut werden. ROS-Messages sind typisierte Datenstrukturen, welche aus beliebig vielen primitiven Datentypen bestehen.

Die über das ROS-Netzwerk verteilten Prozesse, ROS-Nodes, führen die Berechnungen aus und können in beliebigen Programmiersprachen verfasst werden, wobei Python und C++ am weitesten verbreitet sind.

3.3 mathematische Grundlage für anthropomorphe Servosteuerung

Die Steuerung von Servomotoren erfordert zu jedem Zeitpunkt einen Wert zwischen 0° und 180° . Dieser Zahlenwert soll in dem vorliegenden Fall die Gelenkstellung des menschlichen Arms darstellen. Wie in dem vorigen Abschnitt erwähnt, geben die IMUs die Bewegung der einzelnen Körperteile wieder. Um daraus einen Winkel für jeweiligen Servomotoren zu erhalten, muss folgende Berechnung durchgeführt werden. Die Berechnung ist von (Saki, 2020) entwickelt worden.

Zu Beginn wird auf gegebene Eulerwinkeln eine Rotationsvektor angewandt, um daraus Quaternionen zu erhalten 1

$$\vec{V}_{rot}(\vec{e}) = \begin{pmatrix} \cos \frac{e_y}{2} \cos \frac{e_z}{2} \cos \frac{e_x}{2} - \sin \frac{e_y}{2} \sin \frac{e_z}{2} \sin \frac{e_x}{2} \\ \cos \frac{e_y}{2} \cos \frac{e_z}{2} \sin \frac{e_x}{2} - \sin \frac{e_y}{2} \sin \frac{e_z}{2} \cos \frac{e_x}{2} \\ \sin \frac{e_y}{2} \cos \frac{e_z}{2} \cos \frac{e_x}{2} - \cos \frac{e_y}{2} \sin \frac{e_z}{2} \sin \frac{e_x}{2} \\ \cos \frac{e_y}{2} \sin \frac{e_z}{2} \cos \frac{e_x}{2} - \sin \frac{e_y}{2} \cos \frac{e_z}{2} \sin \frac{e_x}{2} \end{pmatrix}^T \quad (1)$$

Die inverse Matrix bei Quaternionen lautet:

$$\vec{V}_{inv}(\vec{w}) = \begin{pmatrix} w_0 & -w_1 & -w_2 & -w_3 \end{pmatrix} \quad (2)$$

Die erhaltenen Quaternionen werden daraufhin verknüpft, um die Position des Vorderarms zu erhalten:

$$\vec{V}_{positionforearm} = \begin{pmatrix} forearmposition_0 \\ forearmposition_1 \\ forearmposition_2 \\ forearmposition_3 \end{pmatrix} \quad (3)$$

$$= \vec{V}_{rot,Arm} \times \vec{V}_{rot,Shoulder} \times \begin{pmatrix} 0 \\ -26.5 \\ 0 \\ 0 \end{pmatrix} \times \vec{V}_{inv}(\vec{V}_{rot,Shoulder}) \times \vec{V}_{inv}(\vec{V}_{rot,Arm}) \quad (4)$$

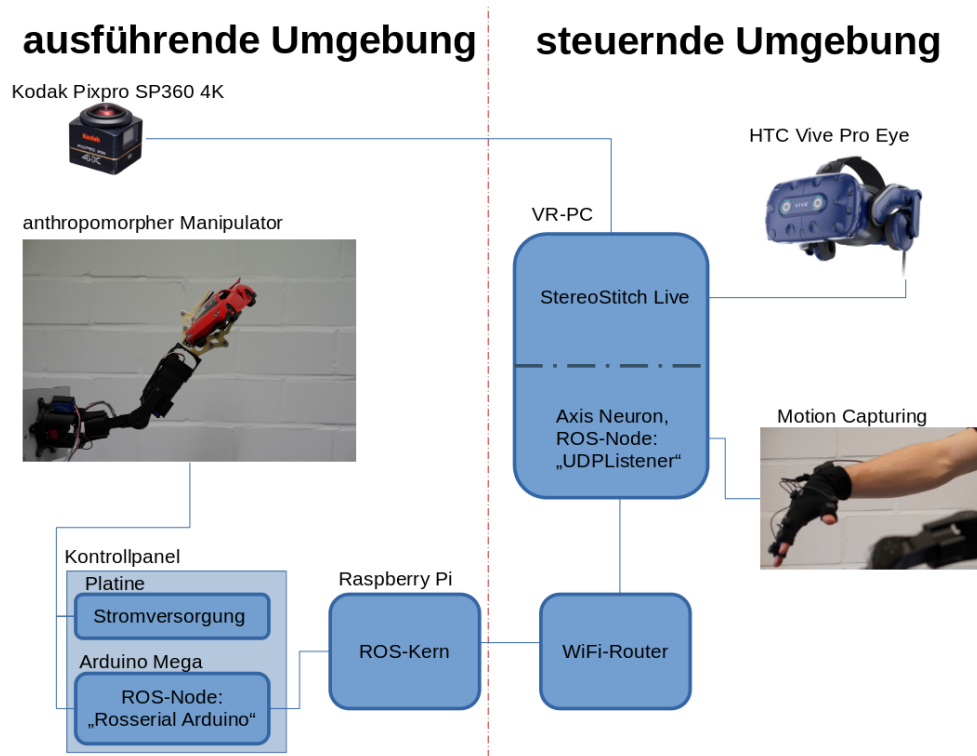
Daraus werden die benötigten Winkel berechnet mit:

$$\phi_{base} = 180 - \arctan \frac{forearmposition_3}{forearmposition_2} \quad (5)$$

Das Ergebnis der Berechnungen wird zum Ende in einen Vektor zusammengefasst 6.

$$\vec{V} = \begin{pmatrix} \phi_{base} \\ \phi_{shoulder} \\ \phi_{elbow} \\ \phi_{upperarmrotation} \\ \phi_{wristrotation} \\ \phi_{wrist} \\ \phi_{gripper} \end{pmatrix} \quad (6)$$

Abbildung 4: Konzept für eine Telepräsenzsteuerung



4 Konzept

Die vorgestellten Technologien ermöglichen nun eine Umsetzung des geplanten Ziels. Um eine bessere Übersicht zu bekommen, wird wie in Abbildung 4 zu erkennen, das Szenario in zwei Bereiche unterteilt. Die steuernde Umgebung wird von der Person genutzt, welche den Arm steuern möchte und die ausführende Umgebung bildet den Roboterarm und dessen Arbeitsbereich ab. Die räumliche Trennung wird durch eine rote Linie symbolisiert.

Dabei gibt es zwei Datenströme, die essentiell für das System sind. Zum einen wird die Sicht über den Arbeitsbereich auf die VR-Brille und zum anderen die Bewegung des Motion-Capturing Anzugs auf den Roboterarm übertragen.

Visuelle Kommunikation von einer entfernten unerreichbaren Umgebung zur nutzenden Person ist nach (Aykut, 2019, S.26) unterteilbar in

4 Schritte. Eine typische Datenübertragung bei einem "Remote Reality System" kann in Aufnahme, Vorverarbeitung, Komprimierung und Darstellung von 360° Bildern unterteilt werden. Mit einer Kodak Pixpro SP360 4K wird das Umfeld des Roboterarms aufgenommen und mithilfe eines HDMI-Kabels und eine HDMI-Input-Karte auf den Rechner "VR-PC" übertragen. Mithilfe des Programms StereoStitch Live wird das ankommende Bild verarbeitet (siehe 3.2.1 und anschließend auf der HTC Vive Pro Eye dargestellt).

Die Bewegung wird mit dem Motion Capturing Anzug von der anwendenden Person aufgenommen und auf dem VR-PC mithilfe des Programms Axis Neuron auf einen UDP-Server geschrieben. Eine ROS-Node liest die ankommenden BVH-Daten und rechnet sie direkt in Winkel um. Diese Winkel werden dann über ein ROS-Netzwerk auf einen am Roboterarm angebrachten Rechner gesandt und dort über einen Arduino an die jeweiligen Motoren übermittelt. Der Arduino soll des Weiteren grundlegende Kontrollfunktionen liefern, welche auf einer Steuerkonsole realisiert werden. Der Arduino und die nötige Elektronik, wie die Stromversorgung der Servomotoren werden in der Konsole untergebracht.

5 Umsetzung

Die Realisierung des in Kapitel 4 skizzierten Konzepts wird im Folgenden erläutert. Dazu wird auf die Nutzung der Methoden aus Kapitel 3 eingegangen. Zu Beginn wird die Entwicklung des verwendeten Manipulators beschrieben, indem zuerst die Konstruktion und Zusammensetzung der Hardware und anschließend die Konfiguration der steuernden Elemente beschrieben wird. Daraufhin wird die Umsetzung der Bildübertragung des in Abbildung 4 dargestellten Datenstroms von der Kamera bis zu den HMDs erläutert. Die Zusammenführung der beiden Systeme wird zum Ende dieses Kapitels behandelt.

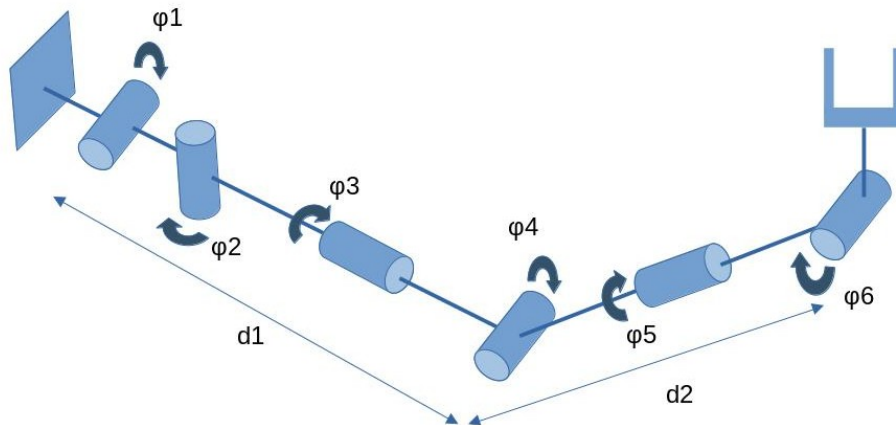
Die in Kapitel 1 und 2 genannten Gestaltungsrichtlinien werden bestmöglich umgesetzt. Zentrale Bedeutung hat die Übertragungsdauer der beiden Datenströme. Diese hat direkten Einfluss auf die Immersion des Systems, da die nutzende Person bei Bewegung des eigenen Arms eine Bewegung des Roboterarms erwartet. Die Übertragungsdauer setzt sich aus zwei Zeiten zusammen. Beginnend mit der Bewegung des Arms der nutzenden Person bis zur tatsächlichen Bewegung des Roboterarms und anschließend der Aufnahme und Übertragung des Kamerabildes auf die HMD-Displays.

5.1 Entwicklung des Manipulators

Die Bewegung in der ausführenden Umgebung soll möglichst intuitiv zu steuern sein. Daher wird in dieser Arbeit versucht einen Manipulator zu schaffen, der sich wie der menschliche Arm bewegen kann und die Winkel des menschlichen Arms direkt übernimmt. Dies deckt sich mit der Forderung für intuitive Interaktionen aus 2.1.2, bereits bekannte Funktionen zu nutzen, um eine neue Technologie intuitiv bedienbar zu machen. Wenn die Bewegungen des Manipulators den Erwartungen der nutzenden Person entsprechen und konsistent ablaufen, kann diese das vorgegebene Ziel vollständig und genau erreichen.

Auf die dazu nötige Gestaltung wird im Folgenden näher eingegangen. Anschließend wird die Konstruktion der Bauteile, die Realisierung der nötigen Elektronik, die Konfiguration der Manipulatorsteuerung und ab-

Abbildung 5: kinematische Kette des Manipulators nach Denavit-Hartenberg Konvention



schließlich die Montage des gesamten Manipulators erläutert.

5.1.1 Gestaltung des Manipulators

Um einen anthropomorphen Roboterarm zu konstruieren sind sieben Freiheitsgrade nötig (siehe 2.1.3). Wegen der kompakten Bauweise werden Modellbau Servomotoren (RC Servo) eingesetzt, um diese zu realisieren. Trotz des geringen Gewichts der Motoren, welches mit zunehmendem Abstand zur Schulter größeren Einfluss hat, wird auf einen Freiheitsgrad, die Abduktion des Handgelenks, verzichtet. Damit bleiben sechs Motoren für den Arm und ein Servomotor für den Endeffektor. Die resultierende kinematische Kette des Manipulators wird entsprechend der Denavit-Hartenberg-Konvention erstellt und ist der folgenden Abbildung zu entnehmen. Durch die gezeigte kinematische Kette kann der Manipulator direkt und unabhängig gesteuert werden, da die Winkel des menschlichen Arms direkt auf die jeweiligen Gelenke des Manipulators übertragen werden können (siehe 2.2.2).

Jeder Servomotor bildet eine entsprechende Gelenkachse ab und muss dementsprechend an der Position, wo die Gelenkachse des Menschen zu finden wäre, angebracht werden. Um die Motoren mit dem in der kine-

matischen Kette folgenden Bauteil zu verbinden, werden Aluminiumplatten mit einer Kerbverzahnung verwendet, welche mithilfe von M4-Schrauben gleichzeitig an dem Bauteil sowie an dem Motor befestigt werden können und so die Übertragung der Rotation ermöglichen.

Der Endeffektor wird von der Firma Arduino aus dem Tinkerkit Braccio entnommen und besteht aus 2 Zangen, welche sich zueinander bewegen können und so einen Greifer realisieren (siehe Abbildung 9).

5.1.2 Konstruktion der Bauteile

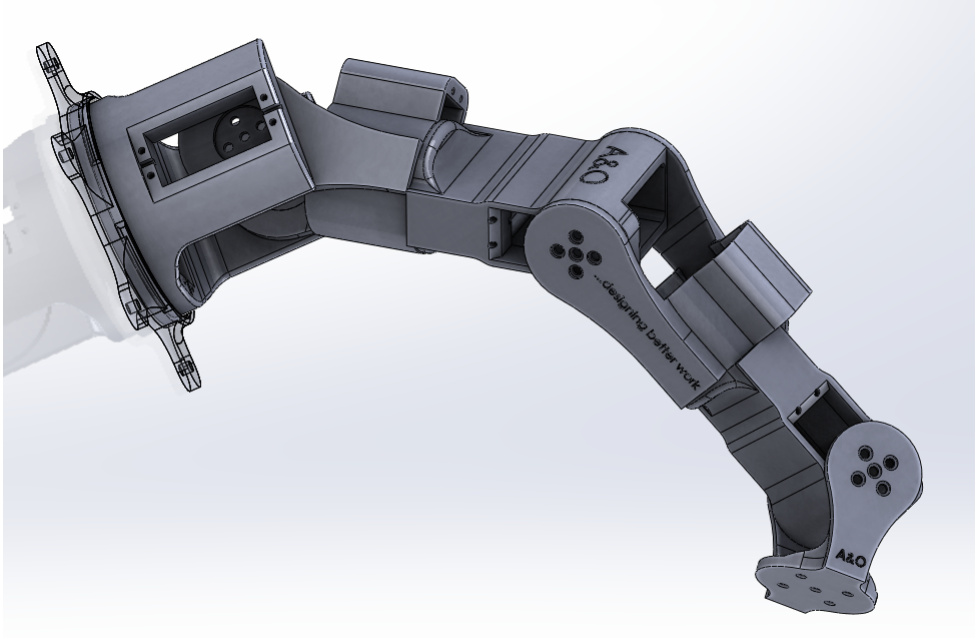
Um die Servomotoren an der passenden Position zu montieren werden Bauteile benötigt, welche Aufnahmen für die Motoren sowie für die zur Verbindung nötigen Kerbverzahnungsplatten bereitstellen. Die Bauteile bieten eine Abschirmung der Motoren und der Motorleitungen nach außen, soweit dies nicht die Bewegung einschränkt. Um den 3D-Druck zu ermöglichen werden die in 3.1.1 genannten Gestaltungshinweise befolgt.

Die Aufteilung der Bauteile erfolgt nach der Position der jeweiligen Gelenkachsen, welche in Abbildung 5 dargestellt sind. An jeder Gelenkachse muss der Manipulator beweglich sein, wodurch ein weiteres Bauteil benötigt wird. Dies resultiert in sieben Bauteile. Dabei wird der Greifer als ein Bauteil betrachtet.

Beginnend mit dem Schultergelenk sind zwei Bewegungsachsen nötig. Das initiale Bauteil dient zur Befestigung des gesamten Manipulators an einer dafür passenden Halterung und muss eine Aufnahme für die Kerbverzahnungsplatte besitzen. Des Weiteren sind Führungsschienen für eine stabile Bewegung nötig.

Das darauf folgende Bauteil wird in die Schienen eingesetzt und mit Hilfe der Kerbverzahnungsplatte, welche direkt mit dem in dem Bauteil montierten Motor verbunden ist, an das Initialbauteil befestigt. Das Bauteil besitzt eine weitere Motoraufnahme für die Realisierung der Bewegungsachse ϕ_2 .

Abbildung 6: simulierte Kombination der Bauteile des Manipulators



Das Bauteil des Oberarms, welches mit der Schulter verbunden wird, soll die Drehung des Oberarms um die eigene Achse ermöglichen. Dies gelingt, indem der Servomotor "vor Kopf" eingebaut wird und das folgende Bauteil nur über die Übersetzungsplatte verbunden wird. Das Bauteil ist in Abbildung X zu sehen.

Jedes Bauteile muss zwei Aufnahmen passend für die gewählten Servomotoren bieten um die Motoren und die (Übersetzungsplatte) befestigen zu können. Die gewählten Motoren werden mithilfe von vier M4 Schrauben befestigt. Die Aufnahme muss demnach folgende Maße besitzen. Die Aufnahme der runden (Übersetzungsplatten) wird mit 5 Durchgangslöchern realisiert. Die genannten resultierenden Bauteile und deren Zusammensetzung werden in Abbildung X dargestellt.

5.1.3 Elektronik

Die Servomotoren haben die drei Anschlüsse: 6V, GND und PWM-Signal. Die Spannungsversorgung erfolgt über einen DC/DC-Wandler, welcher 24V Versorgungsspannung auf 6V konvertiert. Daran können alle 7 Servomotoren angeschlossen werden. Ebenso kann der GND an alle Servomotoren angeschlossen werden. Daher wird die Spannungsversorgung mit einer durchgehenden Leitung realisiert, welche den den jeweiligen Motoren eine Abzweigung hat.

Die PWM-Signale dagegen kommen von dem Arduino Mega und müssen mit jeweil einer Leitung realisiert werden.

Die Stromversorgung muss zu jeder Zeit spannungsfrei gemacht werden können. Diese Vorgabe wird mit einem Not-Aus Taster umgesetzt. So kann Verbindung vor dem DC-DC-Wandler unterbrochen werden.

5.1.4 Manipulatorsteuerung

Die Kontrolle des Manipulators wird mit einem Raspberry Pi realisiert, Der Raspberry Pi stellt das Herzstück des ROS-Netzwerks dar und erlaubt die Kommunikation mit der Peripherie. Die Installation und Konfiguration des Computers wird im Folgenden Abschnitt ?? erläutert. Dieser befindet sich wie in Abbildung 4 zu sehen in der ausführenden Umgebung und ist über eine serielle Verbindung an einen Arduino Mega angeschlossen. Auf die Konfiguration des Arduino wird in Kapitel 5.1.4 näher eingegangen. Dieser liefert die PWM-Signale für die einzelnen Servomotoren. In der steuernden Umgebung befindet sich ein VR-Rechner, welcher die Signale des Motion-Capturing Anzugs verarbeitet. Dies geschieht mit dem in ?? erläuterten Skript.

Raspberry Pi

Zu Beginn wird das Betriebssystem Ubuntu Server 20.04.4 LTS (64-Bit) auf den Speicher, eine Micro-SD-Karte, geschrieben. Die Konfiguration folgt der offiziellen Dokumentation (*Installation*, 2022). Mithilfe einer SSH-Verbindung wird daraufhin ein Nutzer eingerichtet und die Kennung im Netzwerk verändert.

```
$ sudo adduser ros
$ sudo usermod -aG sudo dialout ros
$ ssh-keygen -t ed25519 -C "johannes.hoelker@uni-kassel.de"
```

Mithilfe von nmcli d wird eine WiFi-Verbindung eingerichtet.

Im Anschluss wird ROS Noetic auf dem Raspberry Pi installiert. (*Documentation - ROS Wiki*, 2022) beinhaltet die offizielle Dokumentation und Tutorials, welche die Konfiguration erklären und anhand derer das Setup vorgenommen wird:

```
$ sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu $(lsb_release -sc) main" > /etc/apt/sources.list'
$ curl -s https://raw.githubusercontent.com/ros/rosdistro/master/ros.asc | sudo apt-key add -
$ sudo apt update
$ sudo apt install ros-noetic-ros-base
$ sudo apt install rost-noetic-robot-upstart
$ sudo apt install rost-noetic-rosserial-arduino
$ echo "source /opt/ros/noetic/setup.bash" >> ~/.bashrc
$ sudo apt install python3-rosdep python3-rosinstall python3-rosinstall-generator python3-wstool build-essential
$ sudo rosdep init
$ rosdep update

$ mkdir -p ros_ws/src
$ cd ros_ws
$ catkin_make
$ source devel/setup.bash
$
$ catkin_create_pkg anthropomorphic_manipulator std_msgs rospy message_generation
$ catkin_make
```

Mit einer ROS-Message wird die Kommunikation der Winkel zwischen ausführender und steuernder Umgebung realisiert. Dazu werden in einer msg Datei die einzelnen Felder der Nachricht festgelegt. Für jeden Motor wird ein Zahlenwert benötigt. Außerdem bekommt jede Nachricht einen Header, welcher den Zeitstempel, eine Frame-ID und eine fortlaufende Sequenz-ID beinhaltet:

```
Header header
uint16 base
uint16 shoulder
uint16 elbow
uint16 upper_rot
uint16 wrist_rot
uint16 wrist_ver
uint16 gripper
```

Die Datei Angles.msg wird mithilfe der folgenden Befehle in dem neuen Paket "anthropomorphic_manipulator" eingebunden:

```
$ mkdir msg
$ nano msg/Angles.msg
```

In der package.xml Datei werden die folgenden beiden Zeilen auskommentiert

```
<build_depend>message_generation </build_depend>
<exec_depend>message_runtime </exec_depend>
```

und in der Datei CMakeLists.txt folgende Zeilen ergänzt:

```
find_package(catkin REQUIRED COMPONENTS
  roscpp
  rospy
  std_msgs
  message_generation
)
catkin_package(
  CATKIN_DEPENDS message_runtime
)
add_message_files(
  FILES
  Angles.msg
)
generate_messages(
  DEPENDENCIES
  std_msgs
)
```

Mit dem Befehl

```
$ catkin_make
```

im ROS-Workspace "ros_ws" ist die Erstellung der ROS-Message abgeschlossen.

Um die serielle Verbindung zum Arduino herzustellen, wird das Paket `rosserial_arduino` benutzt. Dazu muss die Arduino IDE installiert werden, welche den Arduino flashen kann und für die Erstellung der nötigen Bibliotheken zuständig ist. Diese wird mithilfe einer `.tar.xz` datei auf den Raspberry kopiert und anschließend mit dem Befehl:

```
$ tar -xf /home/ros/arduino_ws/arduino-1.8.19-linuxaarch64.tar.xz
```

extrahiert. Das Programm lässt sich über Port Forwarding auf einem Rechner mit SSH Verbindung starten nachdem der Befehl

```
$ sudo apt install xorg
```

ausgeführt wurde. Nachdem der Ordner für die Skethes festgelegt wurde, werden die nötigen ROS-Bibliotheken mit folgendem Befehl erstellt:

```
$ rosrn rosserial_arduino make_libraries.py /home/ros/Arduino/libraries
```

Um den Raspberry Pi nach dem Bootvorgang einsatzbereit zu haben, wird das Paket `robot-upstart` genutzt. Dieses führt während des Bootvorgangs mithilfe einer `ros-launch` Datei alle nötigen Befehle aus. Dazu wird in dem erstellten ROS-Pakets ein Ordner `launch` erstellt,

```
$ roscd anthropomorphic_manipulator
$ mkdir launch
$ cd launch
```

, welcher die Datei startup.launch enthält:

```
<launch>
<node name="arduino_talker" pkg="roserial_arduino" type="serial_node.py" args="_port:=/dev/ttyACM0 _baud:=115200" />
</launch>
```

Damit der Nutzer während des Bootvorgangs rechtzeitig die nötigen Rechte bekommt, wird eine udev-Regel mit dem Inhalt

```
ACTION=="add", KERNEL=="ttyACM*", MODE=="0666"
KERNEL=="ttyACM0", MODE=="0666"
```

benötigt, welche als /etc/udev/rules.d/local.rules gespeichert wird.

Abschließend für die Raspberry Pi Konfiguration wird die erwähnte Bootsequenz mit dem Befehl

```
$ rosrund robot_upstart install anthropomorphic_manipulator/launch/startup.launch
```

initialisiert und die beiden Befehle

```
$ roscore
```

und

```
$ rosrund roserial_arduino serial_node.py _port:=/dev/ttyACM0 _baud:=115200
```

werden dadurch automatisch ausgeführt. Der erste Befehl startet den genannten ROS-Kern und der zweite baut eine serielle Verbindung zum Arduino auf.

Arduino Mega

Der an den Raspberry Pi angeschlossene Arduino Mega hat die Aufgabe, die über die serielle Verbindung bereitgestellten Informationen an die Servomotoren weiterzuleiten. Außerdem wird eine Steuerung bereitgestellt, welche die Bewegung jederzeit anhalten, sowie eine Einrichtposition anfahren kann. Das Skript data_to_servo_menu_ros.ino [siehe Anhang] erledigt diese Aufgabe und die Erstellung des Skripts wird im Folgenden beschrieben.

Um die drei Zustände, Einrichten, Stopp und Steuerung aus der Peripherie, zu unterscheiden, werden zu Beginn drei binäre Variablen (val_AC,

val_90, val.Stop) festgelegt. Wenn einer diese Werte wahr wird, tritt der jeweilige Zustand ein. Die Pinbelegung für die physischen Knöpfe und LEDs wird zugewiesen und der Modus eingestellt. Für die drei Zustände gibt es jeweils einen Drucktaster sowie eine Status-LED (siehe Abbildung 9). Des Weiteren wird ein Subscriber auf das Topic "servo" des ROS-Netzwerks festgelegt. Dieses benötigt unter anderem einen Callback ("servo_cb"). Zum Ende der Initialisierung wird eine nach unten zeigende Startposition angefahren, um, wenn die Stromversversorgung hergestellt wird, keine unerwartete ausgreifende Bewegung zu machen.

Die Schleife hat die Aufgabe, bei jeder Iteration zu überprüfen, ob ein Taster gedrückt wurde. Dies geschieht in den if-else Abfragen, welche daraufhin die genannten binären Variablen setzen. Anschließend werden die LEDs auf den gewünschten Wert gesetzt und die PWM-Pins bekommen den Output, der für den jeweiligen Modus nötig ist. Bei der Einrichtposition bedeutet dies ein Werte von 90 auf jeden Motor außer dem Greifer (siehe Klasse angle_90()), bei der Bewegung wird die vom ROS-Netzwerk empfangenen Werte weitergegeben (siehe Klasse recvWithEndMarker()) und bei Stopp die Werte beibehalten, welche bei der vorigen Iteration vorlagen.

Nachdem das Skript fertiggestellt ist, wird es mit dem Programm Arduino IDE auf den Arduino geschrieben. Aufgrund der Kommunikation mit dem Raspberry Pi kommt es daraufhin zu unterschiedlichen Fehlern, welche durch nicht genannte Zeilen behoben wurden und auf die nicht näher eingegangen wird.

VR-PC

Auf dem Rechner in der steuernden Umgebung wird ROS, sowie Axis Neuron installiert. Dabei wird der Anleitung für ROS für windows gefolgt. Es muss ebenfalls die Nachricht Angles.msg eingebunden werden. Die vom Motion Capturing Anzug aufgenommenen Daten werden in dem .bvh Format von dem Programm Axis Neuron der Firma Neuronmocap auf einen UDP Server geschrieben. [cite the code siehe Anhang]

Daraufhin liest eine ROS-Node mithilfe eines UDP-Listeners auf dem gleichen Rechner die Daten ein und verarbeitet sie [siehe Anhang]. Die

dazu nötigen Berechnungen werden von der ursprünglichen Steuerung übernommen. Als Ausgabe werden Winkel produziert, welche die Position der jeweiligen Motoren und Zeitpunkte repräsentieren. Das Skript `udplistener.py` ist im Anhang zu finden und wird mithilfe des Befehls

```
roslaunch testing scripts/udplistener.py
```

gestartet.

5.1.5 Montage des Manipulators

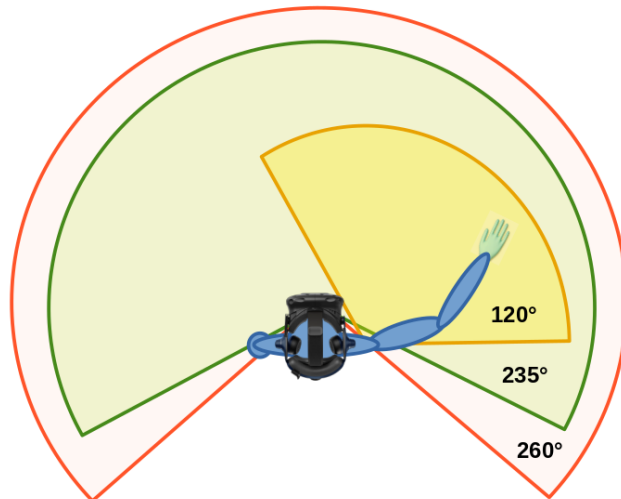
Um die spannungsführenden Teile unerreichbar für nutzende Personen zu machen, wird eine Blackbox eingesetzt, welche die zuvor beschriebene Elektronik und den Arduino beinhaltet. Die Box dient als Bedienpult für die grundlegenden Funktionen und bietet Aufnahmen für die in ?? beschriebenen nötigen Taster und Status-LEDs, sowie eine Halterung für den Not-Aus. In Abbildung 9 ist die fertiggestellte Box zu sehen.

5.2 Bildübertragung

Um in Echtzeit das Bild von der ausführenden zur steuernden Umgebung zu übertragen ist eine schnelle Datenverbindung und Bildzusammensetzung nötig. Die in Kapitel 3.2.1 beschriebenen Prozessschritte zur visuellen Kommunikation werden im Folgenden behandelt und es wird eine begründete Auswahl der Komponenten für die Umsetzung getroffen.

Für die Aufnahme von Weitwinkel-Bildern hat Kodak eine Kamera konzipiert, welche mit universellen Halterungen frei justierbar montiert werden kann und deshalb gut für die Umsetzung geeignet ist. Die Kodak Pixpro SP360 4K hat ein 235° Fish-Eye Weitwinkelobjektiv. Wie in 2.1.3 erwähnt, beträgt die horizontale Drehung des Kopfes 130° in jeweils eine Richtung, was in einen horizontalen Sehraum von 260° in beide Richtungen resultiert. Der vertikale Sehraum von 120° nach hinten und 93° nach vorne sichtbarem Bereich resultiert zu 213° vertikaler Verdrehung. Die horizontale Drehung liegt dabei über den technisch machbaren 235°, die vertikale Drehung unterhalb. Aufgrund der Empfehlung von (Landau, 1992), den Sehbereich

Abbildung 7: Überschneidung des erreichbaren Arbeitsraums des Manipulator und des horizontalen Sehraums und technisch realisierten Sichtbereichs des Menschen



nicht vollständig für Anwendungen zu benutzen, wird trotz des größeren Sehbereichs diese Kamera gewählt. Außerhalb des Bereichs ist anstatt des Kamerabildes eine schwarze Fläche zu sehen. Das Umfeld mit den genannten Größen ist der Abbildung 7 zu entnehmen. Dabei entspricht der rote Bereich dem horizontalen Sehraum des Anwenders, der grüne Bereich dem technisch realisierten Sichtbereich durch die Kodak Pixpro SP360 4K und der gelbe Bereich dem erreichbaren Arbeitsraum des Manipulators.

Als Übertragung zur steuernden Umgebung kommen unterschiedliche Lösungen in Frage. Eine kabellose Lösung über das WLAN-Netz wird präferiert, da dadurch eine räumliche Trennung möglich wird. Um die nötige Übertragungsgeschwindigkeit zu erreichen wird jedoch aufgrund der hohen Datenmenge ein direkter Anschluss an den Rechner in der steuernden Umgebung gewählt.

Die Kamera wird demnach über eine HDMI-Input Karte an den Rechner angeschlossen und das ankommende [Format] Bild wird umgerechnet um auf den HMD dargestellt werden zu können. Für diesen Zweck wird das Programm Stereostitch Live der Firma Dermandar genutzt (siehe 3.2.1). Das in Kapitel 3.3.3 erläuterte Programm SteamVR bietet die Möglichkeit,

die Orientierung der Standposition zurückzusetzen. Damit kann die Ausrichtung der initialen Blickrichtung auf die jeweilige Situation angepasst werden. Diese Funktion wird bei der Einrichtung des Hardwaresetup wichtig.

5.3 Telepräsenzsteuerung

Die beiden zuvor präsentierten Systeme werden in diesem Kapitel zusammengeführt um das Ziel einer Telepräsenzsteuerung zu erreichen. Die nutzende Person soll gleichzeitig mit dem Motion-Capturing Anzug den Manipulator steuern und mit den HMDs das Bilder der Weitwinkelkamera sehen. Um ein immersives Erlebnis zu schaffen wird versucht, die nutzende Person visuell in die Rolle des Roboters schlüpfen zu lassen. Dazu ist die Position der Kamera wichtig, da sie die Position des Kopfes simuliert und die Perspektive für die nutzende Person angibt. Anschließend an die Evaluation der Kameraposition wird das beschriebene System gestartet und die dafür nötigen Hinweise werden gegeben.

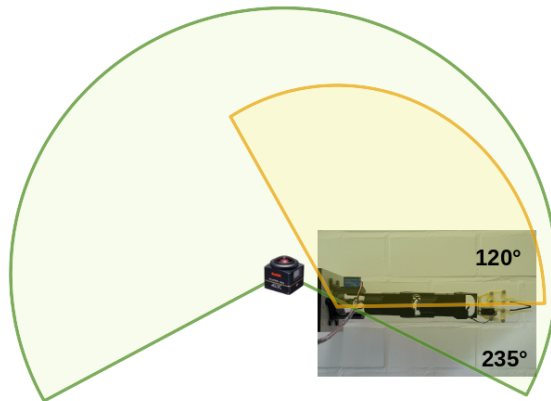
Um weiterhin die Anforderungen von (McMAHAN, 2003) zu erfüllen, sollten die Erwartungen der anwendenden Person im Vorhinein auf das zu erwartende Erlebnis eingestellt werden. Dieser Aspekt lässt sich in der GUI lösen, indem die nutzende Person auf die Umgebung durch schriftliche Hinweise vorbereitet wird.

Damit die Aktionen der nutzenden Person keinen trivialen Effekt auf die Umgebung haben, da

5.3.1 Evaluation der Kameraposition

Der Sehbereich innerhalb der VR-Brille soll nun den gesamten Arbeitsbereich erfassen, welcher von der steuernden Person mit dem Roboterarm angefahren werden kann. Dieser erreichbare Arbeitsbereich umfasst, wie in Kapitel 5.1.5 erläutert, einen Kegel von ungefähr 120°, in Abbildung 8 durch den gelben Kreisausschnitt gekennzeichnet. Dieser Bereich soll nun innerhalb des von der Kamera erfassten Bereichs liegen, damit jede Bewegung des Manipulators von der steuernden Person gesehen wird. Dieser Bereich ist in Abbildung 8 grün gekennzeichnet und beträgt

Abbildung 8: Überschneidung des Sichtbereichs der Kamera und Arbeitsbereichs des Manipulators



235°.

Des Weiteren wird, um die in Kapitel 2 erwähnten Aspekte von (McMAHAN, 2003) umzusetzen, versucht die Kamera bezüglich des Roboterarms auf die gleiche Position zu bringen, an der normalerweise der Kopf zu finden wäre. Um dies zu erreichen wird folgende Methode angewandt:

Zuerst wird das Motion Capturing kalibriert, denn die Kontrolle des Roboterarms ist nach Kalibrierung des Motion-Capturing Systems feststehend. Die Rotation des Oberkörpers wird demnach nicht erkannt, sodass der Oberkörper der Person immer in die gleiche Richtung zeigen muss.

Anschließend wird in der VR-Umgebung die in Kapitel 5.2 erwähnte Funktion des "Sitzposition zurücksetzen" eingesetzt. So haben nun die beiden System, VR-Brille und Motion-Capturing, die gleiche Initialposition und die Kameraposition kann anschließend angepasst werden.

Zunächst wird die Position der Schulter eingestellt. Die anwendende Person schaut mit dem Kopf zu ihrer Schulter und setzt daraufhin die VR-Brille auf, während er/sie weiterhin zur gleichen Stelle schaut. Befindet sich an der gleichen Stelle nun das Kugelgelenk des Roboterarms, ist die Position passend,

ansonsten muss die Höhe der Kamera sowie der Abstand zwischen Kamera und Roboterarm angepasst werden.

Während der Roboterarm als Nächstes per Motion Capturing kontrolliert wird, werden verschiedene eindeutige Posen angefahren. Diese Posen bestehen aus einem ausgestreckten Arm zur Seite, nach oben und nach vorne. Die anwendende Person schaut diesmal mit dem Kopf zu ihrer Hand. Daraufhin setzt sie wie schon bei der Schulter die VR-rille auf, während sie weiterhin zur gleichen Stelle schaut. Befindet sich an der gleichen Stelle nun der Greifer des Roboterarms, ist die Position passend. Wenn nicht muss die Differenz der angeschauten zur realen Position des Greifers durch Rotieren und Bewegen der Kamera ausgeglichen werden.

Mit einem iterativen Durchlaufen der drei Positionen mit jeweiligem Einstellen der Kamerahalterung wird so eine passende Kameraposition erreicht. Anschließend hat die nutzende Person bei der gleichen Bewegung des Kopfes anstatt des eigenen, den fremden Roboterarm an der gleichen Stelle, an der sie den eigenen Arm erwarten würde.

Die beschriebene Methode muss bei Wechseln der nutzenden Person neu angewandt werden, um die veränderten Körpermaße und Standposition zu berücksichtigen. Das System wird möglichst flexibel und anpassbar gebaut, damit jede Person das System nutzen und bedienen kann.

Das entstandene Setup ist der folgenden Abbildung zu sehen.

5.3.2 Einrichtung des gesamten Setups

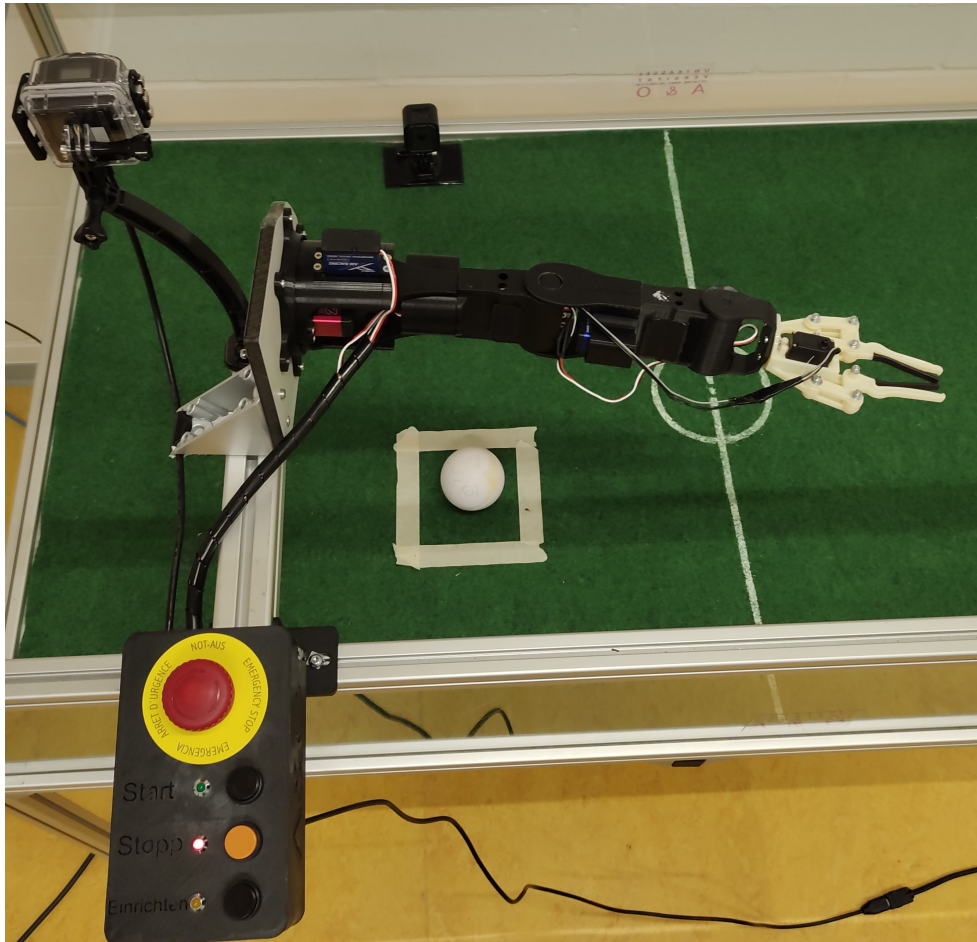
Mit den folgenden Schritten wird die gesamte Telepräsenzsteuerung gestartet. In der ausführenden Umgebung ist folgender Ablauf nötig:

1. Stromversorgung anschließen
2. Anschluss der Pixpro an ein MikroHDMI-Kabel und Einstellen zur Live-Übertragung

In der steuernden Umgebung sind folgende Schritte notwendig:

1. Start von Axis Neuron und Verbindung mit Motion-Capturing Anzug
2. Start von Stereostitch Live und Etablieren der Übertragung auf die HTC Vive

Abbildung 9: ausführende Umgebung einer Telepräsenzsteuerung mit Weitwinkelkamera, anthropomorphem Manipulator und Steuerkonsole



3. Start des Skripts `udplistener.py` in steuernder Umgebung

6 Funktionsvalidierung

Ob das präsentierte System den Zweck der Zielstellung erfüllt und somit die Forschungsfrage beantwortet wird, soll nun anhand von ProbandInnen getestet werden. Die ProbandInnen benutzen das System um eine Aufgabe zu erledigen. Die Aufgabe sieht bei allen nutzenden Personen gleich aus und ist bezüglich der erfolgreichen Ausführung klar definiert. Damit wird messbar, ob die Interaktion intuitiv ist 2.1.2.

Dabei werden mit dem System vertraute, so wie unerfahrene ProbandInnen gemessen. Die Vorerfahrung wird in einem Fragebogen abgefragt. Ebenso wird das Alter und das Geschlecht erfasst.

6.1 Aufgabenbeschreibung

Mithilfe des Manipulators muss eine Kugel von einer bestimmten Position in einen anderen Bereich bewegt werden. Dazu haben die ProbandInnen einen festen Zeitrahmen zu Verfügung. Die Grundposition lässt sich Abbildung X und die gewünschte Endposition Abbildung X entnehmen. Dabei kann es vorkommen, dass der Ball aus dem Greifer fällt und aus dem erreichbaren Arbeitsraum des Roboters herausrollt. Passiert dies, gilt die Aufgabe für den jeweiligen Durchlauf als nicht bestanden und der nächste Durchlauf wird gestartet, indem die Kugel wieder auf den Startpunkt gelegt wird. Die Zeit läuft dabei weiter. Alle teilnehmenden Personen führen die Aufgabe fünf mal aus und das Resultat wird im Fragebogen notiert.

6.2 Versuchsdurchführung

Die ProbandInnen bekommen einen Fragebogen (siehe Anhang) auf welchem folgende Daten vermerkt werden: Geschlecht, Alter, Vorerfahrung, Aufgabe erfolgreich absolviert. Des Weiteren wurden die Versuchspersonen über die Nutzung ihrer Daten informiert und eine Einverständniserklärung wurde unterschrieben.

Anschließend bekommt die teilnehmende Person eine kurze Einführung, bestehend aus Sicherheitshinweisen und der Aufgabenbeschreibung:

- Bedienung der Not-Aus Einrichtung
- Aufklärung über Motion Sickness und die Bitte der sofortigen Beendigung bei Auftreten von Übelkeit
- Erklärung der Aufgabe und des gewünschten Ziels
- Erläuterung der Technik und der Funktionsweise
- Beschreibung der korrekten Nutzung

Die teilnehmende Person zieht den Motion-Capturing Anzug an und setzt die VR-Brille auf. Nach der Kalibrierung beider Systeme gibt es eine kurze Eingewöhnungsphase von einer Minute. Danach kann der zeitliche Rahmen gestartet werden und die Person versucht die Aufgabe zu lösen. Nach jedem Durchlauf wird die Erreichung des Ziels notiert. Ob das Ziel erreicht wurde ist in 6.1 definiert.

Die Versuchspersonen konnten daraufhin die eigene Erfahrung schildern und eine Einschätzung bezüglich der Immersion und der aufgetretenen Zeitverzögerung abgeben.

7 Diskussion und Ausblick

7.1 Auswertung

7.2 Limitationen

Sicherheit bei Crash, etc.

Bei der Evaluierung der Kameraposition kann nur auf einen bestimmten Anwender eingegangen werden, welcher bestimmte Körpermaße hat. Dadurch ist die Position nicht allgemeingültig. Die Körpermaße an sich sind dementsprechend auch ein Problem, da der Roboterarm kürzere Achsverbindungen hat (vgl. Abbildung X) und die Position der Hand demnach nicht genau die Position des Greifers darstellen kann.

Da die Kamera nur einfach vorhanden ist, kann keine Tiefenwahrnehmung stattfinden.

Rotation des Oberkörpers darf nicht stattfinden. Sowohl bei Axis Neuron nicht vorgesehen als auch bei der Kamera nicht möglich. Schulter bleibt sozusagen starr. Es kommt zu Fehlern wenn sie trotzdem bewegt wird.

Das System ist nicht komplett Remote. Um es für einen echten Remote ein-satz umzurüsten muss das Kamerabild über einen RTMP-Stream gebuffert und über das Internet übertragen werden. Zudem muss ROS über das Netzwerk hinaus kommunizieren (VPN-Server) und so die Winkel übertragen.

Wie schon in Kapitel 3.X erwähnt, ist die Abduktion des Handgelenks nicht berücksichtigt worden. Dies kann eventuell dazu führen, dass die nutzende Person bei bestimmten Bewegungen nicht mehr das Gefühl hat, den Roboterarm frei steuern zu können.

Bei einer direkten Steuerung des menschlichen Arms muss bedacht werden, inwieweit Geschwindigkeit und Beschleunigung der menschlichen Bewegung auf den Manipulator übertragen werden können. Die maximale Geschwindigkeit der Gelenkbewegung wird vor allem durch die Spannungsebene der Servomotoren beschränkt. Eine hohe Beschleunigung benötigt eine hohe Leistung, was zu einem hohen Strombedarf resultiert. Damit limitiert die Stromversorgung inwieweit die menschliche Geschwindigkeit und Beschleunigung übertragen werden kann (Siciliano & Khatib, 2008, S.83).

7.3 Ausblick

References

- Aykut, T. (2019). Towards Immersive Telepresence: Stereoscopic 360-degree Vision in Realtime. , 148.
- Blackler, A., & Hurtienne, J. (2007). Towards a unified view of intuitive interaction: definitions, models and tools across the world. (13), 36–54.
- Bonvoisin, J. (2016). Implications of Open Source Design for Sustainability. In R. Setchi, R. J. Howlett, Y. Liu, & P. Theobald (Eds.), *Sustainable Design and Manufacturing 2016* (Vol. 52, pp. 49–59). Cham: Springer International Publishing. Retrieved 2021-02-06, from http://link.springer.com/10.1007/978-3-319-32098-4_5 (Series Title: Smart Innovation, Systems and Technologies) doi: 10.1007/978-3-319-32098-4_5
- Brogårdh, T. (2007). Present and future robot control development—An industrial perspective. *Annual Reviews in Control*, 31(1), 69–79. Retrieved 2021-07-17, from <https://linkinghub.elsevier.com/retrieve/pii/S1367578807000077> doi: 10.1016/j.arcontrol.2007.01.002
- Brown, E., & Cairns, P. (2004). A grounded investigation of game immersion. In *Extended abstracts of the 2004 conference on Human factors and computing systems - CHI '04* (p. 1297). Vienna, Austria: ACM Press. Retrieved 2021-07-12, from <http://portal.acm.org/citation.cfm?doid=985921.986048> doi: 10.1145/985921.986048
- Documentation - ROS Wiki*. (2022). Retrieved 2022-02-26, from <http://wiki.ros.org/>
- Ehlers, K. (2019). *Echtzeitfähige 3D Posenbestimmung des Menschen in der Robotik: Methoden und Anwendungen*. Wiesbaden: Springer Fachmedien Wiesbaden. Retrieved 2022-01-12, from <http://link.springer.com/10.1007/978-3-658-24822-2> doi: 10.1007/978-3-658-24822-2
- Fritsche, L., Unverzag, F., Peters, J., & Calandra, R. (2015, November). First-person tele-operation of a humanoid robot. In *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)* (pp. 997–1002). Seoul, South Korea: IEEE. Retrieved 2021-07-06, from <http://ieeexplore.ieee.org/document/7363475/> doi: 10.1109/HUMANOIDS.2015.7363475
- Goertz, R. C. (1949). Master-slave Manipulator. , 16.

- Hyunchul Kim, Miller, L. M., Byl, N., Abrams, G., & Rosen, J. (2012, June). Redundancy Resolution of the Human Arm and an Upper Limb Exoskeleton. *IEEE Transactions on Biomedical Engineering*, 59(6), 1770–1779. Retrieved 2022-01-18, from <http://ieeexplore.ieee.org/document/6182581/> doi: 10.1109/TBME.2012.2194489
- Installation. (2022). Retrieved 2022-02-27, from <https://ubuntu.com/server/docs/installation>
- IPCC. (2021). *Climate Change 2021: The Physical Science Basis. Contribution of Working Group I to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change* (Tech. Rep.). Retrieved 2022-01-07, from https://www.ipcc.ch/report/ar6/wg1/downloads/report/IPCC_AR6_WGI_Full_Report.pdf
- Krause, C., & Strunz, U. (1997). Telepräsenz bei mobilen Robotern. , 9.
- Landau, K. (Ed.). (1992). *Die Arbeit im Dienstleistungsbetrieb: Grundzüge einer Arbeitswissenschaft der personenbezogenen Dienstleistung*. Stuttgart: Ulmer.
- Lee, H., Um, G., Lim, S. Y., Seo, J., & Gwak, M. (2021, November). Real-time multi-GPU-based 8KVR stitching and streaming on 5G MEC/-Cloud environments. *ETRI Journal*, etrij.2021–0210. Retrieved 2022-02-11, from <https://onlinelibrary.wiley.com/doi/10.4218/etrij.2021-0210> doi: 10.4218/etrij.2021-0210
- Mareczek, J. (2020). *Grundlagen der Roboter-Manipulatoren – Band 1: Modellbildung von Kinematik und Dynamik*. Berlin, Heidelberg: Springer Berlin Heidelberg. Retrieved 2022-01-18, from <http://link.springer.com/10.1007/978-3-662-52759-7> doi: 10.1007/978-3-662-52759-7
- McMAHAN, A. (2003). Immersion, Engagement, and Presence. , 20.
- Mohs, C., Hurtienne, J., Naumann, A., Kindsmüller, M. C., Meyer, H., & Pohlmeier, A. (2006). IUUI – Intuitive Use of User Interfaces. , 4.
- Mohs, C., Naumann, A., & Kindsmüller, M. C. (2007). Mensch-Technik-Interaktion: intuitiv, erwartungskonform oder vertraut? (13), 25–35.
- Murray, J. H. (1997). *Hamlet on the holodeck: the future of narrative in cyberspace* (Updated edition ed.). Cambridge, Massachusetts: The MIT Press.
- Park, S., Jung, Y., & Bae, J. (2017, July). InterActive and intuitive control interFace for a Tele-operAted Robot (AVATAR) system. In *2017 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*

- (pp. 241–246). Munich, Germany: IEEE. Retrieved 2021-07-06, from <http://ieeexplore.ieee.org/document/8014024/> doi: 10.1109/AIM.2017.8014024
- Quigley, M., Gerkey, B., Conley, K., Faust, J., Foote, T., Leibs, J., ... Ng, A. (2009). ROS: an open-source Robot Operating System. , 6.
- Siciliano, B., & Khatib, O. (Eds.). (2008). *Springer handbook of robotics*. Berlin: Springer. (OCLC: ocn153562054)
- Sträter, O., & Bengler, K. (2019, September). Positionspapier Digitalisierung der Arbeitswelt. *Zeitschrift für Arbeitswissenschaft*, 73(3), 243–245. Retrieved 2021-02-06, from <http://link.springer.com/10.1007/s41449-019-00161-2> doi: 10.1007/s41449-019-00161-2
- Tanie, K., Society, I. R. a. A., & (Japan), K. J. S. G. (Eds.). (2003). *MFI 2003: proceedings of IEEE International conference on multisensor fusion and integration for intelligent systems, July 30 - August 1, 2003, National Center of Sciences, Tokyo, Japan*. Piscataway, NJ: IEEE. (Meeting Name: IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (2000-))
- Weng, J., Cohen, P., & Herniou, M. (1992, October). Camera calibration with distortion models and accuracy evaluation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(10), 965–980. Retrieved 2022-02-11, from <http://ieeexplore.ieee.org/document/159901/> doi: 10.1109/34.159901

Fragebogen zur Funktionsvalidierung einer Telepräsenzsteuerung

Alter: _____

Geschlecht: m: ☐ w: ☐ d: ☐

Vorerfahrung mit dem System? JA: ☐ NEIN: ☐

Aufgabe erfolgreich im zeitlichen Rahmen abgeschlossen? (Wird vom Versuchsleiter ausgefüllt)

Durchlauf 1: JA: ☐ NEIN: ☐

Durchlauf 2: JA: ☐ NEIN: ☐

Durchlauf 3: JA: ☐ NEIN: ☐

Durchlauf 4: JA: ☐ NEIN: ☐

Durchlauf 5: JA: ☐ NEIN: ☐

Welche Auffälligkeiten hatte das System?: _____

