UNIKASSEL
VERSITÄT

Universität Kassel
Fachbereich Maschinenbau
Arbeits- und Organisationspsychologie

Bachelorarbeit
Mechatronik

Titel: Intuitive Telepräsenzsteuerung eines anthropomorphen Manipulators

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

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

Kassel, den 20. März 2021

Abstract

Contents

1 Einleitung

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 Gefahrengüter 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, Pflegetätigkeiten 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. 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 Robotersysteme, 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 Kapitel 2.2.4 vorgestellt.

1.2 Zielstellung

An die bereits bestehenden Konzepte soll diese Arbeit anknüpfen und die Entwicklung eines immersiven und intuitiven Systems zur Manipulatorsteuerung wird präsentiert. Die nutzende Person soll möglichst einfach 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.

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. Dabei wird auf Arbeitsplatzgestaltung und die beiden technischen Bereiche Robotik und visuelle Kommunikation eingegangen.
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 eingangen 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

2.1 Arbeitsplatzgestaltung

2.1.1 Getaltungsrichtlinien

In der vorliegenden Arbeit wird 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 sowie die folgenden Richtlinien bedacht und evaluiert.

Für die Entwicklung von Robotersystemen hat (Asimov, 1942) in der Kurzgeschichte ”Runaround” die drei Robotergesetze formuliert:

- Ein Roboter darf einem menschlichen Wesen keinen Schaden zufügen oder durch Untätigkeit zulassen, dass einem menschlichen Wesen Schaden zugefügt wird.
- Ein Roboter muss den Befehlen gehorchen, die ihm von Menschen erteilt werden, es sei denn, dies würde gegen das erste Gebot verstoßen.
- Ein Roboter muss seine eigene Existenz schützen, solange solch ein Schutz nicht gegen das erste oder zweite Gebot verstößt.

Diese Richtlinien müssen ebenso in der Entwicklung berücksichtigt werden. Im konkreten Anwendungsfall bedeutet dies, dass der Roboter exakt das tut was von ihm gewünscht wird und seine Möglichkeiten für die Erledigung der Aufgabe vollkommen ausschöpft.

Für die Gestaltung ist neben den genannten notwendigen Richtlinien ebenso die Performanz des zu entwickelnden Systems wichtig. Die folgenden Gestaltungsmaßnahmen sind nicht sicherheitsrelevant, führen aber zu einer besseren Kontrolle des Systems.

2.1.2 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.”

Die erfolgreiche teleaktive Bedienung eines Systems ist abhängig vom Grad der Immersion. Dies wird durch den Einsatz von Head-Mounted Displays erreicht, welcher der nutzenden Person visuell den Eindruck vermitteln, sich an dem entsprechenden Ort zu befinden (Aykut, 2019) (Krause & Strunz, 1997). 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.

Immersion

Das lateinische Wort ”immersio” bedeutet ”in etwas Eintauchen” und ist der Ursprung für den Begriff der Immersion. Die nach (McMAHAN, 2003) 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 (Murray, 1997).

Um ein möglichst hohes Maß an Immersion zu erreichen, sind nach (McMAHAN, 2003) drei Aspekte wichtig:

- 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 gannten 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.

Intuition

(Mohs et al., 2006) bezieht Intuition ausschließlich auf Informationsverarbeitungsvorgänge des Menschen und definiert ein intuitiv nutzbares System als ohne Vorwissen effektiv bedienbar. Um demnach 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, Naumann, & Kindsmüller, 2007) bei der Gestaltung eines Systems eine intuitive Nutzung ermöglicht werden. Eine Liste aus sieben Kriterien wurde von (Mohs et al., 2006) aufgestellt:erarbeitet:

- Kompatibilität
- Konsistenz
- Gestaltgesetze
- Rückmeldungen
- Selbstbeschreibungsfähigkeit
- Affordances

2.1.3 Anthropometrie

menschlicher Sehraum

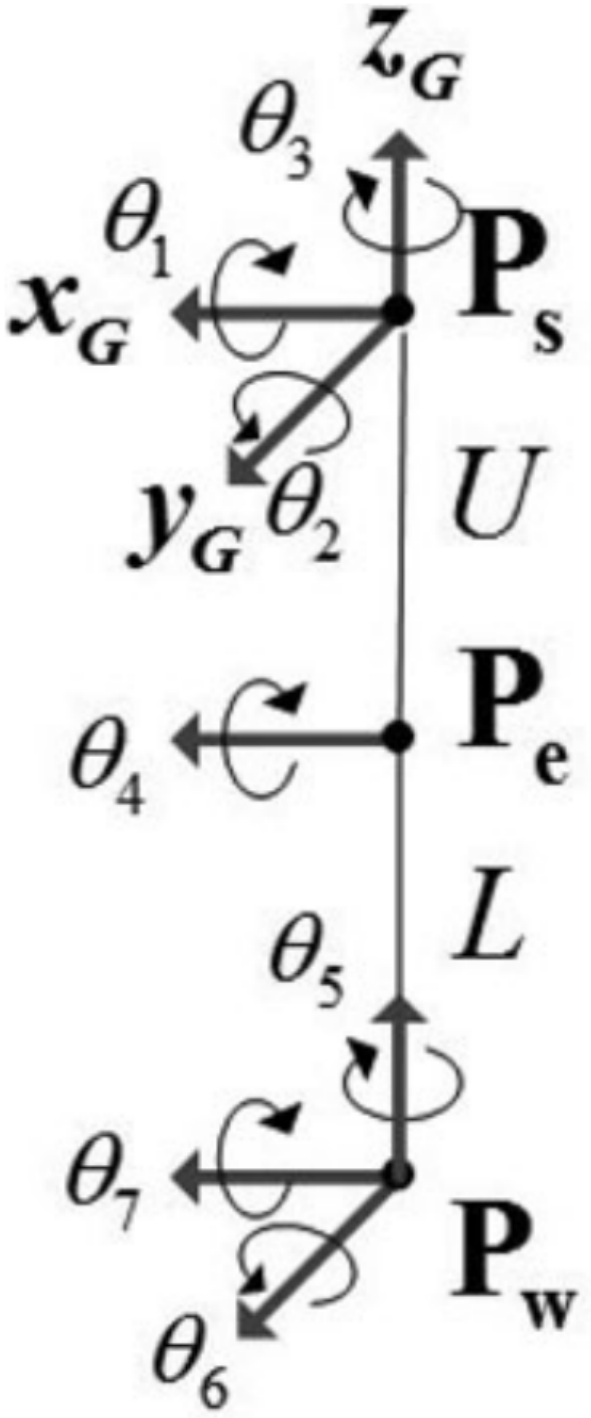
Der menschliche Sehraum ist nach (Landau, 1992) [Abbildung] eingeteilt in die Bereiche Gesichtsfeld, Blickfeld, Umblickfeld sowie das Umblick-Gesichtsfeld. Das Gesichtsfeld ist dabei als der Bereich des scharfen Sehens bei ruhenden Augen zu verstehen, das Blickfeld ist der bei bewegten Augen zu sehende 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° nach links und rechts und bei vertikaler Bewegung 120° mit nach hintem geneigtem Kopf und 93° bei einem nach vorne geneigtem Kopf. Dabei ist zu beachten, dass die angegebenen Beriche nicht als Optimalbereich für technische Anwendungen zu betrachten sind.

Oberkörperproportionen

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


z_G
θ_3
θ_1
x_G
$\mathbf{P_s}$
y_G
θ_2
U
θ_4
$\mathbf{P_e}$
L
θ_5
θ_7
$\mathbf{P_w}$
θ_6


$\mathbf{P_s}$
α
$\mathbf{P_e}$

2.2 visuelle Kommunikation

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.

2.2.1 Aufnahme - Fish-Eye Kamera

Weitwinkelkameras haben eine nach außen gewöblte [Fachbegriff] Linse, welche es ermöglicht, einen großen Bereich aus einem einzigen Blickwinkel aufzunehmen (Aykut, 2019, S.15). Bei stark ausgeprägter Wölbung kann eine Aufnahme über 180 Grad hinaus ermöglicht werden, wobei dann von Fish-Eye Kameras geredet wird.

2.2.2 Vorverarbeitung - radiale Verzerrung

Bei dem Gebrauch von Weitwinkelkameras tritt in dem entstehenden Bild eine radiale Verzerrung auf (Aykut, 2019, S.15). Um einen zentralen Bildpunkt herum verschieben sich die Bildpunkte nach innen oder außen. Dabei verschiebt sich der zentrale Bildpunkt nicht und bildet den passenden mittigen Pixel des realen Bildes ab (Weng, Cohen, & Herniou, 1992, S.968). Verschieben sich die Bildpunkte nach außen, spricht man von ”barrel distortion”. Diese tritt bei Fish-Eye Kameras auf.

Um das Bild ohne die fehlerhafte Verzerrung darzustellen wird es auf eine Kugel projiziert. Die nutzende Person soll sich demnach bei der Darstellung im Mittelpunkt dieser Kugel befinden. Um dies zu erreichen ist die gleichrechteckige Projektion am meisten verbreitet. Dabei wird die Kugel in Kreise von Breitengraden (horizontale Kreise) unterteilt, welche dann auf horizontale Linien auf die 2D-Bildebene projiziert werden.

2.2.3 Darstellung - Head Mounted Display

Wie in Kapitel 2.1.2 von (Mareczek, 2020) erwähnt, werden binokulare Displays genutzt um visuell Immersion zu erzeugen. Diese Displays sind in Datenbrillen integriert, welche die nutzende Person trägt und somit nicht

mehr die Außenwelt in der sie sich befindet, sieht.

Von (Aykut, 2019) wird die Erweiterung der Datenbrille, das Head-Mounted Display (HMD), empfohlen, welches auf den Displays die jeweiligen Bilder einer stereospokischen Kamera abspielt. Das HMD besitzt außerdem Umgebungssensoren, welche es der nutzen Person erlauben, den Kopf zu bewegen und damit die Bilder auf den Displays an die Kopfposition anzupassen.

Bei der Nutzung 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 mitgeilt werden.

2.3 Robotik

2.3.1 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).

2.3.2 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 Berechung 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.

2.3.3 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 Endeffektorkoordinatesystems nach (?, ?) (DH-Konvention) vorgenommen. Pro Gelenk ist nach der DH-Konvention nur eine Bewegungsachse vorgesehen (Mareczek, 2020, S.42).
Um nun einen Roboter anthropomorph zu gestalten wird die in Kapitel 2.1.3 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).

2.3.4 Server/Datenübertragung

2.3.5 Stand der Technik/bisherige Lösungsansätze

[cite Fritsche et al]. stellten ein System mit einer VR-Brille und MotionTracking mithilfe einer Kinect Kamera vor. Es wurde ein iCub gesteuert, welcher die Schulter des Menschen
Remotesteuerung eines (Jiang, McCoy, Lee, & Tan, 2017) stellten ein fahrbares System eines Roboterarms exklusive des Schultergelnks vor, welches mithilfe von Beschleunigungssensoren am menschlichen Arm gesteuert wurde. (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.

2.4 Forschungsfrage/Fragestellung

Die Programmierung und Steuerung von Manipulatoren geschieht zur Zeit noch über ein Umdenken des Anwenders indem er 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 schwierig, da die Programmierung mithilfe von Kameras aus verschiedenen Blickwinkeln erfolgen muss.

Ein System, welches die Technologien der Robotik und der Datenbrillen kombiniert, kann helfen diese Problematik zu lösen. Durch den Einsatz von HMDs soll die anwendende Person die Perspektive des Roboters einnehmen und den Roboterarm intuitiv steuern können.

Die in dieser Arbeit behandelte Forschungsfrage lautet demnach: Wie lässt sich ein System konstruieren, um immersive Telepräsenz zu erreichen und so eine intuitive Steuerung eines Manipulators zu ermöglichen?

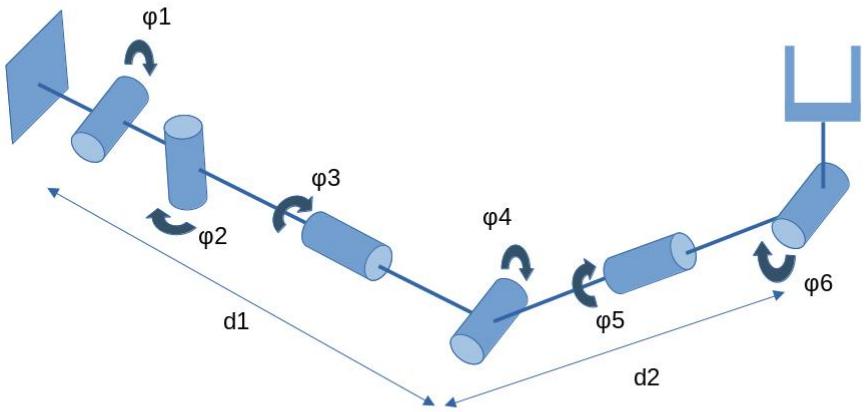
3 Methoden

3.1 Motion Tracking

Um die Bewegung eines menschlichen Arms aufzuzeichnen und darstellen zu können, wird mit sogenannten Motion-Capturing Systemen gearbeitet. Diese messen die Bewegung der Gliedmaßen und können so die Gelenkstellungen ausgeben. Es gibt unterschiedliche Messmöglichkeiten, welche in optische und inertiale (getragene) Systeme unterteilt werden können. Optische Systeme benötigen eine umfangreiche Installation, weshalb wie bei (Park et al., 2017) in dieser Arbeit auf ein Messsystem mithilfe von [IMUs] gesetzt wird. Diese IMU zeichnen die einzelnen Beschleunigungen und Drehungen auf und können so die Relativbewegung zwischen den jeweiligen Knochen darstellen.

3.2 Manipulator des Instituts

Der verwendete Manipulator wurde am Fachgebiet A&O für die Steuerung per Motion Capturing entwickelt und ist deshalb in Bezug auf die Kinematik anthropomorph konstruiert. [ist die kinematische Kette des verwendeten Roboterarms zu entnehmen. Erkennbar ist, dass exklusive der Bewegung des Greifers sechs Freiheitsgrade in dem System bestehen. Dies ist einer weniger als nach (Hyunchul Kim et al., 2012) für die anthropomorphe Gestaltung nötig. Die Abduktion des Handgelenks ist aus technischen Gründen nicht berücksichtigt worden. Durch die gezeigte kinematische Kette kann der Manipulator direkt und unabhängig gesteuert werden (siehe Kapitel 2.2.2). Die Winkel des menschlichen Arms werden direkt auf die jeweiligen Gelenke des Manipulators übertragen. Die Gelenke werden durch Servomotoren realisiert. Der Arbeitsbereich des Manipulators umfasst einen Kugelkeil von ungefähr 120 ° [siehe Abbildung X].



Angebunden sind die einzelnen Servomotoren an einen Arduino Due und eine externe Stromversorgung. Die Servomotoren bekommen über pulsweitenmodulierte Signale die Winkel, welche sie jeweils einnehmen sollen. Die Stromversorgung wird mit einem DC/DC-Wandler bereitgestellt und kann mit einem Not-Aus Taster unterbrochen werden.

Der Arduino stellt die grundlegenden Funktionen, wie Start und Stopp der Bewegung sowie einen Einrichtmodus zur Verfügung. Eine serielle Schnittstelle über USB steht zur Verfügung, um mit der Peripherie zu kommunizieren.

3.3 Bildübertragung

3.3.1 HTC Vive

Die HTC Vive 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.

Außerdem besitzt die Brille Audioausgaben für beide Ohren. Sie wird über einen speziellen Adapter an den Computer angeschlossen und kommuniziert mit diesem über SteamVR.

3.3.2 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.

3.3.3 StereoStitch Live

Für die Echtzeit Generierung von 360° Videos in 60fps empfiehlt (Lee, Um, Lim, Seo, & Gwak, 2021) das Programm StereoStitch.

3.3.4 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 Liveübetragung des aufgenommenen Bildes zu ermöglichen.

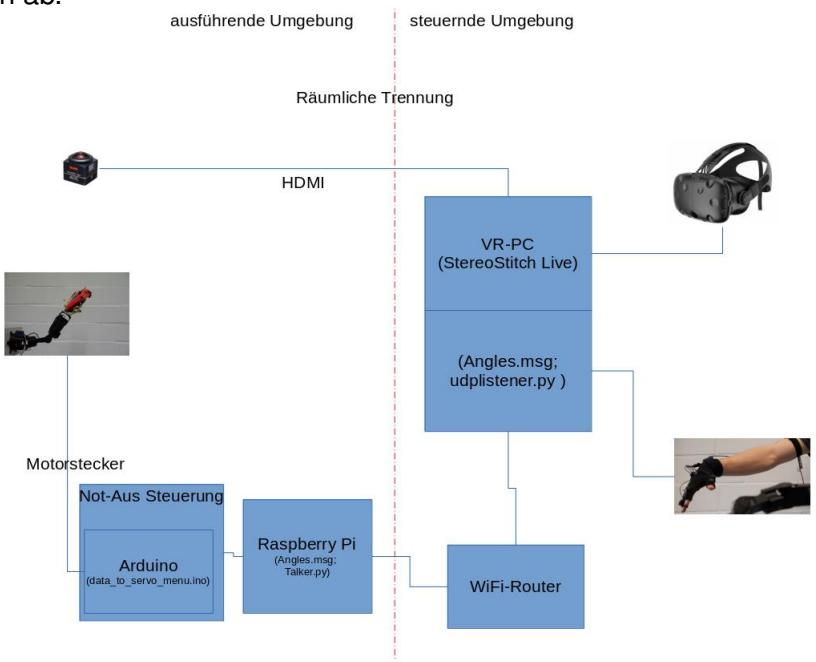
3.4 Robot Operating System

Um auf die unteschieldichen 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 Operating 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 Programmierspraches und ist als OpenSource-Software frei verfügbar und erweiterbar.
Die grundlegende Struktur eines ROS-Netzwerks besteht aus einem Kern, welcher auf einem Pc gestartet wird. Alle anderen an den gleichen Router angeschlossenen Geräte können mit diesem Kern kommunizieren, indem sogenannte ROS-Nodes gestartet werden. Diese Prozesse führen die Berechnungen aus. Mithilfe der ROS-Messages werden Informationen auf den ROS-Topics bereitgestellt. ROS-Messages sind typisierte Datenstrukturen welche aus beliebig vielen primitiven Datentypen bestehen und welche auf die jeweiligen Anforderungen angepasst werden können. Die ROS-Topics stehen global zur Verfügung und können von Nodes abgerufen (subscribe) und verändert (publish) werden (Quigley et al., 2009).

4 Konzept

Die vorgestellten Technologien ermöglichen nun eine Umsetzung des geplanten Ziels. Um eine bessere Übersicht zu bekommen, wird wie in Abbildung XX 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.



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. Mit einer Weitwinkelkamera wird das Umfeld des Roboterarms aufgenommen und mithilfe eines HDMI-Kabels und eine HDMI-Input-Karte auf den Rechner mit der VR-Brille übertragen. Das ankommende Bild wird anschließend für die VR-Brille zugeschnitten und gestreckt/gestaucht.
Die Bewegung wird mit dem Motion Capturing Anzug von der anwendenden Person aufgenommen und auf dem gleichen System direkt in Winkel umgerechnet. 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.

5 Umsetzung

Das in Kapitel 4 skizzierte Konzept soll nun umgesetzt werden. Die verwendeten Geräte, Programme, sowie die Anwendung der Methoden werden in den folgenden Kapiteln erläutert. Zuerst werden die beiden Datenströme und anschließend das Hardware-Setup beschrieben.

Die in Kapitel 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 tatsächichen Bewegung des Roboterarms und anschließend der Aufnahme und Übetragung des Kamerabildes auf die HMD-Displays.

5.1 Kontrolle des Roboterarms

5.1.1 udplistener.py - VR-PC

Die vom Motion Capturing Anzug in der steuernden Umgebung 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 und stellen keinen Bestandteil dieser Arbeit dar. Als Ausgabe werden Winkel produziert, welche die Position der jeweiligen Motoren und Zeitpunkte repräsentieren.
Das Skript udplistener.py ist im Anhang zu finden.
Gestartet wird das Skript auf dem jeweiligen Rechner mit dem Befehl:

```
$ rosrun testing scripts\udplistener.py
```

5.1.2 Angles.msg

Um diese von udplistener.py produzierten sieben Zahlenwerte über das Rosnetzwerk zu versenden, wird eine ROS Message erstellt, welche für genau

diesen Anwendungsfall konzipiert wird. Die zu übermittelnden Daten sind die sieben Integer Werte, welche die Winkel der einzelnen Gelenke repräsentieren, sowie der Zeitpunkt zu dem sie versandt werden, die ID des jeweiligen Frames und die Sequenznummer zur Durchnummerierung der fortlaufenden Frames. Die drei letztgenannten Informationen sind nicht notwendig, helfen aber bei der Fehlersuche und Weiterentwicklung des Systems.

5.1.3 roscore, rosserial_arduino - Raspberry Pi

Der Raspberry Pi in der ausführenden Umgebung dient als Kern des Ros Netzwerks. Er erzeugt eine Instanz, auf der an den gleichen Router angeschlossene Geräte Topics lesen und schreiben können. Auf diesem wird dementsprechend beim Start über den Befehl

```
$ roscore
```

der ROS-Kern gestartet.
Des Weiteren muss eine serielle Verbindung zum Arduino aufgebaut werden, was über den Befehl

```
$ rosrun rosserial_arduino serial_node.py _port:=/dev/ttyACM0 _baud:=115200
```

gestartet wird.

5.1.4 Arduino

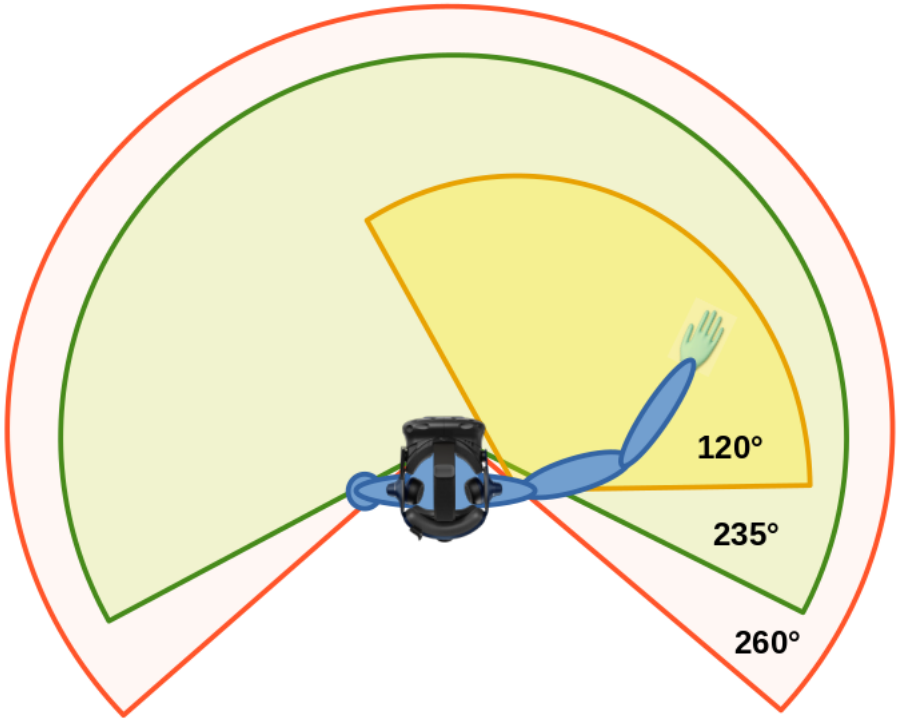
Der an den Raspberry Pi angeschlossene Arduino empfängt nun die über die serielle Verbindung bereitgestellten Informationen und schickt sie direkt an die Motoren weiter. Das Skript data_to_servo_menu_ros.ino [siehe Anhang] erledigt diese Aufgabe und stellt gleichzeitig die direkte Steuerung über physische Knöpfe zur Verfügung, sodass die Bewegung des Roboterarms jederzeit gestoppt werden kann.

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 2.2 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 Kapitel 2.X 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 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 folgenden Abbilung zu entnehmen. Dabei entspricht der rote Beriech dem horizontalem Sehraum des Anwenders, der grüne Bereich dem technisch realisierten Sichtbereich durch die Kodak Pixpro SP360 4K und der gelbe Bereich dem erreichbaren Arbeitsraum des Roboterarms.



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.

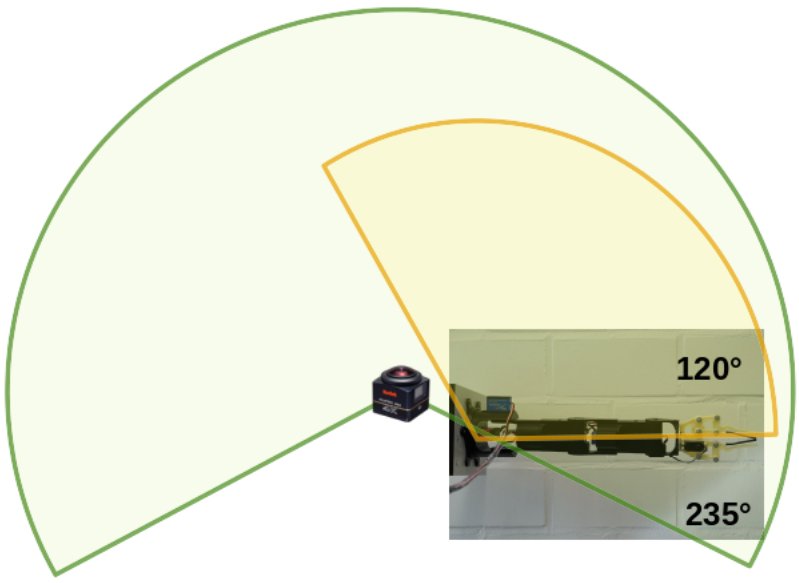
Die Kamera wird demnach über eine HDMI-Input Karte an den Rechner angeschlossen und das ankommende [Format] Bild wird anschließend umgerechnet um auf den beiden Bildschirmen der VR-Brille dargestellt werden zu können. Für diesen Zweck wird das Programm Stereostitch Live der Firma Dermandar genutzt.
Das Programm StereoStitch Live arbeitet dabei mit einer GUI, welche es der nutzenden Person ermöglicht, Einstellungen bezüglich der Ausrichtung der Kamera vorzunehmen, sowie das Live-Rendering auf eine VR-Brille zu starten. Das Programm startet daraufhin SteamVR, welches direkt mit der VR-Brille kommmuniziert.
Das in Kapitel 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 Hardwaresetup

5.3.1 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 3 erläutert, einen Kugelkeil von ungefähr 120°, in Abbildung X 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 X grün gekennzeichnet und beträgt 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

berücksichten.
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

6 Funktionsvalidierung

Damit die nutzende Person räumlich das Gefühl bekommt es sei ihr eigener Arm, soll ein möglichst großes Maß an Immersion erzeugt werden. Die in Kapitel 2.X genannten Aspekte wurden bei der Entwicklung berücksichtigt.

7 Diskussion und Ausblick

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 dementsrpechend 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 einsatz 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 Handegelenks nicht berücksichtig worden. Dies kann eventuell dazu führen, dass die nutzende Person bei bestimmten Bewegungen nicht mehr das Gefühl hast, 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).

References

Asimov, I. (1942). *I, robot* (this paperback edition published by HarperVoyager 2013 ed.). London: HarperVoyager.

Aykut, T. (2019). Towards Immersive Telepresence: Stereoscopic 360-degree Vision in Realtime. , 148.

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

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

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

Jiang, J., McCoy, A., Lee, E., & Tan, L. (2017, October). Development of a motion controlled robotic arm. In *2017 IEEE 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON)* (pp. 101–105). New York City, NY: IEEE. Retrieved 2021-02-06, from http://ieeexplore.ieee.org/document/8248998/ doi: 10.1109/UEMCON.2017.8248998

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., & Pohlmeyer, 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