

Hendrik Wiese  
Entwicklung einer Basisplattform für  
Telepräsenzsysteme

Bachelorthesis eingereicht im Rahmen der Bachelorprüfung  
im Studiengang Mechatronik  
an der Fakultät Technik und Informatik  
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer : Prof. Dr. rer. nat. Thomas Lehmann  
Zweitgutachter : Prof. Dr.-Ing. Andreas Meisel

Abgegeben am 15.04.2014

**Hendrik Wiese**

**Thema der Bachelorthesis**

Entwicklung einer Basisplattform für Telepräsenzsysteme

**Stichworte**

Telepräsenz, Robotik, Stereovision, Mapping, Raumklang

**Kurzzusammenfassung**

In einer Zeit, in der örtliche Flexibilität beruflich wie auch privat immer größeren Stellenwert einnimmt, gewinnt auch die Fähigkeit, an mehreren Orten gleichzeitig zu sein, immer mehr an Bedeutung. In dieser Arbeit wird eine Analyse des Forschungsgebietes und des Marktes für Telepräsenzroboter vorgenommen sowie der Aufbau eines Telepräsenzroboters beschrieben, der als Basisplattform für weitergehende Forschung dient. Der Roboter ermöglicht dem Bediener mittels einer Stereokamera und einem Headtracking System das räumliche Sehen in der Umgebung des Roboters. In der Gegenrichtung ermöglicht ein Audiosystem die verbale Interaktion mit der Umgebung. Sensoren an der Plattform liefern dem Bediener Informationen über die Umgebung außerhalb seines Sichtfeldes.

**Hendrik Wiese**

**Title of the paper**

Development of a base platform for telepresence systems

**Keywords**

telepresence, robotics, stereovision, mapping, surround sound

**Abstract**

In times when flexibility regarding location, in business as well as in private terms, becomes more and more significant, being in more than one location simultaneously equally becomes more important. This thesis contains an analysis of the research area and the market of telepresence systems followed by a description of the development of a telepresence robot that acts as a basis for further research. The operator of the robot will be able to see in three dimensions by means of a stereovision camera that is coupled with a head mounted display and a head tracker. In the other direction it will be possible to verbally interact with the environment using a headset and a sound system that is mounted on the robot. Sensors on the platform provide information about the environment outside of the operator's field of view.

# Inhaltsverzeichnis

<b>Tabellenverzeichnis</b>	<b>5</b>
<b>Listings</b>	<b>6</b>
<b>Abbildungsverzeichnis</b>	<b>7</b>
<b>1. Einleitung</b>	<b>9</b>
1.1. Zielgruppe . . . . .	9
1.2. Persönliche Motivation . . . . .	10
1.3. Ziel . . . . .	10
1.4. Übersicht über diese Thesis . . . . .	11
<b>2. Recherche und Stand der Technik</b>	<b>12</b>
2.1. Forschungsprojekte im Bereich Telepräsenzrobotik . . . . .	12
2.2. Forschung an Teilkonzepten . . . . .	15
2.3. Kommerzielle Produkte . . . . .	17
<b>3. Anforderungen und Konzept</b>	<b>22</b>
3.1. Anforderungen . . . . .	23
3.2. Gesamtkonzept . . . . .	24
3.3. <a href="#">Bedienerstation</a> . . . . .	25
3.4. <a href="#">Mobileinheit</a> . . . . .	27
<b>4. Aufbau des Telepräsenzroboters</b>	<b>37</b>
4.1. Grundlegender Aufbau . . . . .	37
4.2. Im Department verfügbare Komponenten der <a href="#">Mobileinheit</a> . . . . .	39
4.3. Angefertigte Komponenten der <a href="#">Mobileinheit</a> . . . . .	45
4.4. <a href="#">Steuerungssystem der Mobileinheit (ME)</a> . . . . .	52
4.5. Sicherheitsaspekte . . . . .	53
4.6. Komponenten der <a href="#">Bedienerstation</a> . . . . .	54
4.7. Software . . . . .	57
<b>5. Zukunft des Systems</b>	<b>75</b>
5.1. Allgemeine Erweiterungen . . . . .	75

5.2. Erweiterung und Verbesserung der <a href="#">Mobileinheit</a> . . . . .	77
5.3. Erweiterungen der <a href="#">Bedienerstation</a> . . . . .	81
<b>6. Fazit und Ausblick</b>	<b>84</b>
<b>Literaturverzeichnis</b>	<b>87</b>
<b>Glossar</b>	<b>93</b>
<b>Abkürzungsverzeichnis</b>	<b>98</b>
<b>Symbolverzeichnis</b>	<b>100</b>
<b>A. Datenträger</b>	<b>101</b>

# Tabellenverzeichnis

2.1. Betrachtete <a href="#">Telepräsenzroboter (TPR)</a> -Projekte . . . . .	12
2.2. Teilkonzepte für Soft- und Hardwarekomponenten . . . . .	15
4.1. Kennzeichnungen in <a href="#">Abbildung 4.1</a> . . . . .	38
4.2. Kennzeichnungen in <a href="#">Abbildung 4.7</a> und <a href="#">Abbildung 4.8</a> . . . . .	47
4.3. Kennzeichnungen in <a href="#">Abbildung 4.15</a> . . . . .	70

# Listings

4.1. /etc/udev/rules.d/60-tpv.rules . . . . .	44
4.2. roscore.conf Konfiguration für Upstart . . . . .	60
4.3. Modifizierte 10.roslaunch.sh . . . . .	61
4.4. Konfiguration der Controller . . . . .	63
4.5. Übersetzung des Headtrackings . . . . .	64
4.6. Auszug aus sixaxis.py - Übersetzung der Joy-Nachrichten in Twist-Nachrichten	67
4.7. Auszug aus dead_mans_button.py . . . . .	68
4.8. Launchfile für Betriebsmodus der <a href="#">Mobileinheit</a> . . . . .	71
4.9. Launchfile für Betriebsmodus der <a href="#">Bedienerstation</a> . . . . .	73

# Abbildungsverzeichnis

2.1. MIT - MeBot . . . . .	13
2.2. Willow Garage - Texai Remote Presence System . . . . .	14
2.3. Vereinfachter TeleHead . . . . .	16
2.4. iRobot - AVA . . . . .	17
2.5. iRobot - AVA 500 . . . . .	18
2.6. InTouch Health - RP-7i . . . . .	19
2.7. InTouch Health - RP-Vita . . . . .	20
2.8. Suitable Technologies, Inc. - Beam . . . . .	21
3.1. Konzeptdiagramm . . . . .	24
3.2. Rolling-Shutter-Effekt bei einem schnell vorbeifahrenden Auto . . . . .	29
3.3. Rolling-Shutter-Effekt bei einem Flugzeugpropeller . . . . .	30
3.4. Drehachsen aus der Luftfahrt . . . . .	32
3.5. Die drei Achsen anhand der menschlichen Halswirbelsäule . . . . .	33
3.6. pitch→yaw . . . . .	34
3.7. yaw→pitch . . . . .	34
3.8. links: Achsen in Grundausrichtung; rechts: Pitch und Roll in Gimbal Lock . . . . .	35
4.1. <a href="#">Mobileinheit</a> in Frontansicht . . . . .	37
4.2. Schematische Darstellung der <a href="#">Mobileinheit</a> . . . . .	38
4.3. Schematische Darstellung der <a href="#">Bedienerstation</a> . . . . .	38
4.4. Pioneer 3-DX von Adapt MobileRobots . . . . .	40
4.5. Bumblebee von Point Grey Research Inc. . . . .	42
4.6. Webcam Logitech C310 . . . . .	43
4.7. <a href="#">Kopfeinheit</a> Seitenansicht links . . . . .	46
4.8. <a href="#">Kopfeinheit</a> Frontansicht . . . . .	46
4.9. Lage maximaler Belastung für einzelnen Servo . . . . .	48
4.10. Verbindungsschema der <a href="#">Bedienerstation</a> und <a href="#">Mobileinheit</a> . . . . .	51
4.11. Oculus Rift Development Kit . . . . .	54
4.12. SIXAXIS™ Controller . . . . .	56
4.13. Abstrakte Darstellung der Kommunikation im <a href="#">Robot Operating System (ROS)</a> nach [ <a href="#">Conley u. a. (2014b)</a> ] . . . . .	60
4.14. Kontrolle des Pioneer 3-DX mittels linkem Analogstick . . . . .	67

---

4.15. Plugin rqt_ops für die Überwachung der <a href="#">Mobileinheit</a> . . . . .	70
5.1. SPS200 von TSL Professional Products Ltd. . . . .	77
5.2. SpaceNavigator der 3Dconnexion GmbH, München . . . . .	79
5.3. Eine Karte der Umgebung . . . . .	80
5.4. Hexapod . . . . .	80
5.5. Virtuix Omni . . . . .	82

# 1. Einleitung

In Zeiten wachsender Ansprüche bezüglich Mobilität und Flexibilität gewinnen technische Telekommunikationsmittel an Bedeutung. Das Wachstum auf dem Mobilfunkmarkt seit Mitte der 1990er Jahre zeigt einen steigenden Bedarf an permanenter Erreichbarkeit und Vernetzung. Mobilfunk bietet die Möglichkeit, verbal und visuell über Distanz zu kommunizieren. Jedoch fehlt dem System die Fähigkeit, den Benutzer am Ort der Gegenstelle mit der Umwelt direkt interagieren zu lassen und sich dort anwesend zu fühlen. An dieser Stelle bekommen **Telepräsenzsysteme (TPS)** und **Telepräsenzroboter** ihre Berechtigung. **TPS** werden aufgrund ihrer stationären Bauart primär in Videokonferenzen eingesetzt. **TPR** vermitteln ihrem Benutzer das Gefühl realer Präsenz am Zielort. Dies erreichen sie, indem sie sensorische Sinneseindrücke an den Benutzer übermitteln. Sie erlauben es ihm, am Ort des Gerätes physisch mit der Umgebung zu interagieren.

## 1.1. Zielgruppe

Unter die Zielgruppen von **TPS** oder **TPR** fallen Personen im Arbeitsleben, die viel reisen. Eine Herausforderung für sie besteht darin, vor Ort Kundenakquise zu betreiben und gleichzeitig mit dem Arbeitgeber jederzeit Rücksprache halten zu können. **TPS** und **TPR** ermöglichen einem Arbeitnehmer die Teilnahme an Konferenzen, Meetings und Präsentationen, ohne persönlich vor Ort sein zu müssen.

Sinnvoll kann der Einsatz von **TPR** für Kanalarbeiter sein, die aufgrund gesundheitlicher Gefahren oder baubedingter Einschränkungen bestimmte Bereiche der Kanalisation nicht erreichen können, in denen sie Arbeiten verrichten müssten. **TPR** können aufgrund ihrer Größe oder Spezialisierung in Bereiche vordringen, die für Menschen unzugänglich sind.

Die Mobilität nimmt im Durchschnitt mit steigendem Alter ab, jedoch fordert die Gesellschaft diese Mobilität weiter ein. Damit können ältere Mitmenschen für die Nutzung von **TPR** in Betracht gezogen werden. Sie können zu solchen Systemen greifen, um damit alltägliche Aufgaben wie Einkäufe oder den Gang zum Briefkasten zu erledigen, und Pflegepersonal kann mit Hilfe von **TPR** dem Pflegebedürftigen jederzeit zur Verfügung stehen, ohne permanent anwesend sein zu müssen.

Ärzte und Chirurgen erhalten durch TPR die Möglichkeit, ihre Arbeit über Distanz zu verrichten. Operative Eingriffe können damit über Entfernungen vorgenommen werden, ohne dass der Patient oder der Arzt diese Entfernungen dafür überwinden muss.

Neben zivilen Anwendungen spielen das Militär, die Polizei und der Katastrophenschutz als Zielgruppen eine große Rolle. TPR können in der Aufklärung Dienste leisten, die unter normalen Umständen lebensgefährlich für Soldaten sein können. Katastrophenhelfer können TPR nutzen, um in für Menschen unzugängliche Gebiete wie die radioaktiv kontaminierte Zone um das havarierte Kernkraftwerk in Fukushima/Japan vorzustoßen und dort Hilfe zu leisten und Arbeiten zu verrichten. Kampfmittelräumdienste können mittels eines TPR Entschärfungen von Sprengkörpern vornehmen, ohne sich selbst in Gefahr zu bringen. Darauf spezialisierte TPR erledigen diese Aufgaben unter Umständen effizienter als der Mensch selbst.

Allgemein lässt sich sagen, dass als Zielgruppe für mobile und stationäre TPS Menschen in Frage kommen, deren Anwesenheit an einem Ort erforderlich ist und die aus unterschiedlichen Gründen diesen Ort nicht oder nicht in hinreichender Zeit erreichen können.

## 1.2. Persönliche Motivation

Telepräsenz hat den Autor dieser Arbeit, so schlicht diese Formulierung klingen mag, tatsächlich schon in Kindertagen fasziniert. Vielleicht bedingt unter dem Begriff der *Telepräsenz* hatte der Gedanke, gleichzeitig zum Lernen in der Schule und zum Spielen zuhause sein zu können, dennoch schon damals ganz natürlich einen ungeheuren Reiz. Dieser Gedanke, nahe oder auch ferne Orte besuchen zu können, ohne sich wirklich selbst dort hin begeben zu müssen, motiviert den Autor, diesem Themenfeld seine eigenen Ideen und Konzepte hinzuzufügen.

## 1.3. Ziel

Die HAW Hamburg strebt Forschungen auf dem Gebiet der mobilen Telepräsenz an. Zu diesem Zweck wird eine erweiterbare Basisplattform für einen TPR benötigt. Das Ziel dieser Thesis ist der Aufbau dieser Plattform. Für sich genommen hält diese die grundlegenden Fähigkeiten eines solchen Systems vor. Darunter fallen beispielsweise ferngelenkte Bewegung und immersive Wahrnehmung der Umgebung des TPR durch den Bediener.

Für die **Immersion** ist es wichtig, dass eine realitätsnahe und verzögerungsfreie Übermittlung von Sinneseindrücken an den Bediener erfolgt. Es ist ein Konzept zu entwerfen, welches die Aufnahme, Übertragung und Wiedergabe dieser Sinneseindrücke ermöglicht.

Das zu entwickelnde System besteht aus einem mobilen und einem stationären Teil. Der mobile Teil des Systems ist von einer Person am stationären Systemteil zu bedienen und soll in der Lage sein, sich durch eigenen Antrieb in der Ebene fortzubewegen. Kommunikation zwischen Bediener und Personen in der Umgebung der mobilen Einheit ist zu ermöglichen und Sicherheitsaspekte bezüglich Diebstahlschutz und Unfallvermeidung sind zu beleuchten. Das Gesamtsystem ist auf Erweiterbarkeit hinsichtlich Forschung im Bereich der **Immersion** und der autonomen Navigation auszulegen.

## 1.4. Übersicht über diese Thesis

In **Kapitel 2** dieser Thesis wird eine Übersicht über am Markt verfügbare mobile Telepräsenzgeräte gegeben, die verschiedenen Teilgebieten der Telepräsenz zuzuordnen sind. Darüber hinaus benennt und beschreibt das Kapitel einige Forschungsprojekte verschiedener Institute weltweit, die sich mit der Entwicklung von Telepräsenzrobotern oder der Forschung auf dem Gebiet der Telepräsenz widmen. Projekte zu Teilaspekten von Telepräsenzrobotern beziehungsweise Telepräsenz allgemein finden Betrachtung. **Kapitel 2** stützt sich hauptsächlich auf wissenschaftliche Arbeiten aus Quellen wie dem **IEEE** sowie Produktbroschüren und Websites von Herstellern betrachteter Systeme.

Das **Kapitel 3** widmet sich den Anforderungen und der Ausarbeitung des Konzepts der für die HAW Hamburg entwickelten Basisplattform für einen **TPR**. Es werden spezifisch die Anforderungen an einzelne Komponenten des Systems betrachtet sowie allgemeine Anforderungen an das Gesamtsystem formuliert.

**Kapitel 4** beschreibt den tatsächlichen Aufbau des Systems. Die einzelnen Komponenten der Hard- und Software werden vorgestellt sowie Vorteile und Nachteile beleuchtet. Es wird Programmcode für die Bedienschnittstelle und die Steuerung der **Kopfeinheit (KE)** aufgeführt und dokumentiert. Designentscheidungen werden begründet.

In **Kapitel 5** werden Themenvorschläge für Weiterentwicklungen und Ausarbeitungen vorgebracht. Teilaspekte, die in dieser Arbeit keine Berücksichtigung gefunden haben, deren Durchführung explizit ermöglicht wird, werden aufgeführt.

Des Weiteren befindet sich in **Kapitel 6** ein Fazit über das entwickelte System. Die Erfüllung der in **Kapitel 3** aufgeführten Anforderungen sowie der Aufgabenstellung wird betrachtet. Erfahrungen während der Ausarbeitung werden erörtert.

## 2. Recherche und Stand der Technik

Telepräsenz (TP) beschreibt das Gefühl, sich an einem fernen Ort zu befinden, ohne persönlich dort anwesend zu sein. Telepräsenzsysteme ermöglichen es, dieses Gefühl zumindest im Ansatz zu erleben. Bei diesen Systemen wird unterschieden zwischen ortsfesten und mobilen Geräten. Stationäre TPS beschränken sich auf die bidirektionale Übertragung von visuellen und audiovisuellen Sinneseindrücken sowie verbaler Kommunikation. Dem gegenüber ermöglichen mobile TPR dem Benutzer, sich am Ort der ME zu bewegen und abhängig von der Ausstattung des Systems mit der Umgebung zu interagieren.

Einsatzfähige TPS und TPR gibt es am Markt bereits zahlreich. Institute und Universitäten betreiben des Weiteren Forschungsprojekte im Bereich der TP. Unter anderem arbeiten Forscher daran, die Immersion zu steigern. Parallel dazu wird an der Verbesserung der Kommunikation und Interaktion zwischen TPR und vor Ort anwesenden Personen geforscht. Dies bezieht nonverbale Komponenten wie bewusste Gestik, Mimik und unbewusste Körpersprache mit ein.

In den folgenden Sektionen werden einige der mobilen TPR und Forschungsprojekte sowie Teilprojekte aus dem Bereich der TP und eine repräsentative Auswahl bereits erhältlicher Geräte vorgestellt.

### 2.1. Forschungsprojekte im Bereich Telepräsenzrobotik

Renommierte Forschungsinstitute befassen sich in Projekten mit der Entwicklung von TPR. Eine Auflistung der in dieser Arbeit betrachteten Systeme zeigt Tabelle 2.1.

Tabelle 2.1.: Betrachtete TPR-Projekte

Bezeichnung	Institut/Unternehmen	Besonderheiten
MeBot	MIT	Gestikübertragung
Texai	Willow Garage	nutzt ROS und Standardhardware
Beaming	Universität de Barcelona	Virtual Reality, Eyetracker

Das [Massachusetts Institute of Technology \(MIT\)](#) beschäftigte sich 2009 unter Leitung des Masterstudenten Sigurður Örn Aðalgeirsson mit der Entwicklung des vergleichsweise kleinen [TPR MeBot](#) ([Abbildung 2.1](#)). Primäres Ziel des [TPR MeBot](#) und der Master Thesis war der Nachweis, dass die Qualität der Kommunikation mit der Einbeziehung nonverbaler Komponenten (Gesten) wesentlich verbessert werden kann. Zu diesem Zweck wurden dem Roboter Gliedmaßen gegeben, die in einem gewissen Rahmen die Gestik des Bedieners widerspiegeln können. [[Örn Aðalgeirsson \(2009\)](#)]



Abbildung 2.1.: MIT - MeBot  
[[MIT Media Lab \(2014\)](#)]

Das [Texai Remote Presence System](#) ([Abbildung 2.2](#)) ist ein prototypischer Konferenz-TPR, der von der Firma [Willow Garage](#) aus Kalifornien, USA entwickelt wird. Ein aus diesem Projekt abgeleitetes System kam 2012 im Auftrag der aus dem Projekt entstandenen Firma [Suitable Technologies, Inc.](#) in kommerzialisierter Form unter dem Namen [Beam](#) auf den Markt. Im Gegensatz zum kommerziellen und höher integrierten Produkt [Beam](#) basiert [Texai](#) auf handelsüblichen PC-Komponenten sowie Sensorik aus der Industrie. Für die Steuerung der [ME](#) nutzt [Texai](#) das [Robot Operating System \(ROS\)](#). Die Kommunikationsübertragung erfolgt durch eine kommerzielle Videokonferenzsoftware. [[Willow Garage \(2014d\)](#)]

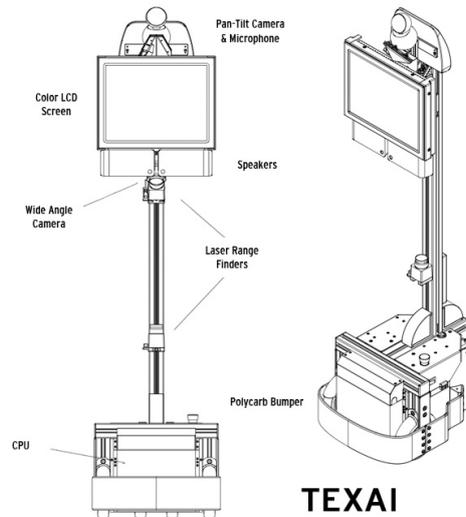


Abbildung 2.2.: Willow Garage - Texai Remote Presence System  
 [Willow Garage (2014c)]

Die *Universität de Barcelona* entwickelt ein [TPS](#) mit Namen *Beaming*. Bei diesem Konzept handelt es sich um ein System mit dem Ziel, die Immersion des Bedieners zu maximieren. Der Bediener soll das Gefühl bekommen, sich tatsächlich am Ort der Mobileinheit zu befinden. Ungewöhnlich an dem System der Universität Barcelona ist der Aufbau der [Bedienerstation \(OPS\)](#) und der [ME](#). Erstere besteht im wesentlichen aus einem Stereovisionssystem mit Headtracking und einem Motion-Capturing-Anzug, der dem Bediener die Übertragung natürlich wirkender Bewegungen und damit nonverbale Kommunikation und direkte Interaktion ermöglicht. Mit Hilfe eines Eyetrackers ist dem System darüber hinaus die Aufnahme von Augenbewegungen des Bedieners möglich. Das beinhaltet sowohl die Blickrichtung als auch Lidschlag und Pupillengröße. Bei der Mobileinheit handelt es sich um eine Vielzahl von Ansätzen. Darunter fallen rein virtuelle Alter Egos und physisch existente stationäre, anthropomorphe Roboter, die die Gestik und Mimik des Bedieners wiedergeben können. Die Universität Barcelona führt mit Hilfe von *Beaming* Entwicklungen an einem automatischen Übersetzer interkulturell unterschiedlicher, nonverbaler Kommunikation durch. Das System ist fähig, aus der in asiatischen Kulturen üblichen Begrüßungsgeste der Verbeugung die in westlichen Ländern eher verwendete Grußgeste des Winkens und umgekehrt zu erzeugen. [Steptoe u. a. (2012)]

## 2.2. Forschung an Teilkonzepten

Neben Unternehmen, Instituten und Hochschulen, die an Gesamtkonzepten um das Thema TP forschen, existieren zahlreiche Projekte, die sich mit Teilgebieten der TP beschäftigen. Einige dieser Projekte werden in diesem Kapitel kurz vorgestellt. [Tabelle 2.2](#) führt eine Übersicht der hier betrachteten Teilkonzepte auf.

Tabelle 2.2.: Teilkonzepte für Soft- und Hardwarekomponenten

Bezeichnung	Institut/Unternehmen
Autonomous Person Following for Telepresence Robots	Georgia Institute of Technology
Improved Human Interaction in Telepresence Robot using Real-time Face Segmentation	Nanyang Technological University
The effect of head movement on sound localization in an acoustical telepresence robot	NTT Communication Science Laboratories

Das *Georgia Institute of Technology* arbeitet im Projekt *Autonomous Person Following for Telepresence Robots* daran, einem TPR das autonome Folgen einer natürlichen Person beizubringen. Die Forschung des Instituts geht dabei über das Folgen der Person hinaus. Das System und damit der Bediener kann während des Followings mit der zu folgenden Person interagieren. [[Cosgun u. a. \(2013\)](#)]

Die Forscher im Projekt *Improved Human Interaction in Telepresence Robot using Real-time Face Segmentation* der *Nanyang Technological University* in Singapore arbeiten an einem Algorithmus zur Ausblendung des Hintergrundes in einer Videokonferenz. Die beteiligten Wissenschaftler der Nanyang TU sind der Auffassung, dass durch die Entfernung des Hintergrundes die Qualität der Wahrnehmung der Kommunikation verbessert werden kann. [[GU u. a. \(2013\)](#)]

Die *NTT Communication Science Laboratories* in Kanagawa, Japan erforschen die Auswirkungen der Kopfbewegung eines TPR auf die Fähigkeit der akustischen Klangquellenortung des Benutzers. Die Forscher des Projekts *The effect of head movement on sound localization in an acoustical telepresence robot: TeleHead* stellten fest, dass sich die stationäre Haltung und die Wahrnehmung von Klangquellen aus unterschiedlichen Richtungen und in unterschiedlichen Lautstärken durch Drehung des Kopfes positiv auf die Genauigkeit der Ortung einer Schallquelle auswirken. Darüber hinaus konnten die Wissenschaftler keinen signifikanten Verlust der Qualität bei der Klangortung nachweisen, wenn dem Subjekt nur eine dieser Quellen zur Verfügung steht. In weiteren Experimenten ermittelten sie die minimale

Latenz zwischen eigener Kopfbewegung und der entsprechenden Bewegung des Telehead. Nicht diese Verzögerung ist für die Ortung einer Klangquelle wichtig, sondern der daraus resultierende Unterschied zwischen somatosensorisch wahrgenommener Pose des eigenen Kopfes und der akustisch wahrgenommenen Lage der den Roboterkopf umgebenden Klangquellen. Die Wissenschaftler entwickelten anhand dieser Erkenntnisse den *TeleHead*. Dabei handelt es sich um einen ballähnlichen Roboterkopf ([Abbildung 2.3](#)). Der Kopf entspricht dank vereinfachter Ohrmuscheln den akustischen Eigenschaften eines menschlichen Kopfes hinreichend genau. Mit Hilfe seines Aufbaus erhöht er die Genauigkeit der Lokalisierung von Klangquellen in der Umgebung des Roboters. [[Toshima und Aoki \(2006\)](#)] [[Toshima und Aoki \(2007\)](#)] [[Toshima und Aoki \(2009\)](#)]



Abbildung 2.3.: Vereinfachter TeleHead  
[[Toshima und Aoki \(2009\)](#)]

## 2.3. Kommerzielle Produkte

AVA von *iRobot* ist eine für [TPR](#) geeignete kommerzielle Basisplattform, die mit verschiedenen Aufbauten versehen werden kann. *iRobot* liefert eine Reihe von fertigen Aufsätzen und die Schnittstelleninformationen an Dritthersteller, damit diese ihre eigenen Aufbauten auf AVA implementieren können. [Abbildung 2.4](#) zeigt AVA mit einigen beispielhaften Aufbaukonzepten.



Abbildung 2.4.: iRobot - AVA  
[[iRobot \(2014b\)](#)]

AVA nutzt ein [holonomisches Fahrwerk](#) und kann somit vor, zurück und Kurven fahren, sich auf der Stelle drehen und sich seitwärts in alle Richtungen bewegen. Das Gerät stellt Umgebungssensorik bereit, die für [Simultaneous Localization And Mapping \(SLAM\)](#) geeignet ist. Dank intelligenter Navigation besitzt AVA die Fähigkeit den optimalen Weg zu einem vorgegebenen Ziel innerhalb des bekannten Areals zu ermitteln. AVA kann in diesem Gebiet das vorgegebene Ziel autonom anfahren und dabei Hindernissen ausweichen, ohne dass der Benutzer eingreifen muss. Es nutzt diese Fähigkeit unter anderem zum Anfahren der eigenen Ladestation und zum Folgen von Personen. [[iRobot \(2014c\)](#)]

Eines der von iRobot hergestellten Systeme ist der in [Abbildung 2.5](#) gezeigte *AVA 500*. Er basiert auf der Basisplattform AVA und ist ein klassischer Konferenz-TPR. Das System ist in der Lage, autonom zu navigieren und im Rahmen seiner physischen Fähigkeiten Ziele anzufahren. Alleinstellungsmerkmal von AVA 500 gegenüber den anderen vorgestellten Konferenz-TPR ist die aus der Ferne höhenverstellbare Kopfeinheit. Die Höhenverstellung ermöglicht es dem Bediener, sich in Konferenzen auf Augenhöhe mit anderen Teilnehmern zu begeben. [[iRobot \(2014a\)](#)]



Abbildung 2.5.: iRobot - AVA 500  
[[Robotics Business Review \(2014\)](#)]

Die [Abbildung 2.6](#) zeigt den *RP-7i* der Firma *InTouch Health* aus Santa Barbara, Kalifornien. Dabei handelt es sich um einen medizinischer [TPR](#). Das System ermöglicht dem bedienenden Arzt, aus der Ferne mit seinen Patienten in Kontakt zu treten. Der *RP-7i* ist mit einem großen Display ausgestattet, auf dem sich der Arzt seinem Patienten zeigen kann. Darüber hinaus besitzt der *RP-7i* ein Erweiterungssystem, mit dem eine Auswahl an medizinischem Gerät an ihm angebracht und aus der Ferne bedient werden kann. [[InTouch Health \(2014a\)](#)]



Abbildung 2.6.: InTouch Health - RP-7i  
[[InTouch Health \(2014a\)](#)]

Neben dem RP-7i vertreibt das Unternehmen InTouch Health zusammen mit der Firma iRobot den *RP-Vita* ([Abbildung 2.7](#)). Das Gerät basiert auf der Basisplattform AVA von iRobot.



Image courtesy of InTouch Health

Abbildung 2.7.: InTouch Health - RP-Vita  
[[Business Wire \(2013\)](#)]

Ebenso wie AVA 500 und im Gegensatz zu RP-7i kann RP-VITA in dem ihm bekannten Gebiet autonom navigieren. Das Gerät lässt sich mit einem Tablet aus dem Hause *Apple* und der Anwendung *InTouchHealth* bedienen. Mittels eines redundanten Netzwerks von Servern überall auf dem Globus wird der zuverlässige Verbindungsaufbau zwischen **OPS** und **ME** sichergestellt. Die Anwendungsschicht der Netzwerkarchitektur unterstützt unter anderem den Videocodec H.264 zur hochauflösenden Bildübertragung sowie Verschlüsselung nach dem AES256 Standard. [[InTouch Health \(2014b\)](#)]

Das Unternehmen Suitable Technologies vertreibt mit Beam ein System, das auf dem von Willow Garage entwickelten Prototypen-TPR Texai basiert. Beam vereint, wie in [Abbildung 2.8](#) zu erkennen, entsprechend seiner prototypischen Vorlage ein Display samt Lautsprecher und Kamera mit einem differentiell angetriebenen Fahrwerk. Die Größe des Displays beträgt 17 Zoll mit einem Seitenverhältnis von 4:3. Den Herstellerangaben zufolge ist das Fahrwerk in der Lage, Beam auf eine Geschwindigkeit von  $0,9 \text{ m s}^{-1}$  zu beschleunigen. Wie die Mehrheit der betrachteten Systeme besitzt Beam die Fähigkeit, die eigene Ladestation, sofern in Sicht- und Reichweite, autonom anzufahren. [[Suitable Technologies, Inc. \(2014b\)](#)]



Abbildung 2.8.: Suitable Technologies, Inc. - Beam  
[[Suitable Technologies, Inc. \(2014a\)](#)]

Die Kommunikation durch und mit Beam basiert auf derzeit als zuverlässig betrachteten Verschlüsselungsstandards wie TLS/SSL, AES-256 und HMAC-SHA1. Beam verwendet zwei Weitwinkelkameras zur Aufnahme der Umgebung sowie sechs Mikrofone, die Echos und Hintergrundgeräusche herausfiltern können. [[Suitable Technologies, Inc. \(2014b\)](#)]

### 3. Anforderungen und Konzept

In [Kapitel 2](#) wurde anhand einiger Beispiele gezeigt, dass sowohl der Markt als auch der Forschungssektor der Telepräsenz Fortschritte gemacht hat. Das Konzept der Telepräsenzrobotik und die Vertiefung der [Immersion](#) stößt bei Unternehmen und Forschern auf breites Interesse.

Dieses Kapitel beinhaltet die Anforderungen an das System und die Ausarbeitung des Gesamtkonzepts. Darüber hinaus beinhaltet es die Konzepte für die [OPS](#) und die [ME](#). Es werden Problemlösungen betrachtet sowie Konzeptentscheidungen begründet.

Für den Betrieb eines Robotersystems auf ebenen Untergründen gibt es zwei Fahrwerkstypen:

- holonom
- differentiell

Ein holonomisches Fahrwerk besitzt drei Freiheitsgrade. Diese ermöglichen es, in der Ebene gerade Bewegungen sowie Kurvenbewegungen in jeder Richtung auszuführen und sich auf der Stelle zu drehen. Fahrwerkskonzepte, die diese Fähigkeit besitzen, setzen beispielsweise auf Kugeln oder sogenannte *Omni-Wheels*. Durch Kombination der drei Freiheitsgrade sind komplexe Bewegungen möglich. Beispielsweise kann sich ein holonomisches Fahrwerk um seine vertikale Achse drehen, während es sich in einer geraden Linie bewegt.

Differentielle Fahrwerke besitzen zwei Freiheitsgrade. Der Freiheitsgrad der Seitwärtsbewegung fehlt. Aufgrund dieser Einschränkung können diese Fahrwerke vorwärts, rückwärts und Kurven fahren, sowie sich auf der Stelle drehen. Seitwärtsbewegung und komplexere Bewegungsmuster sind nicht möglich. Diese Fahrwerke benötigen zwei separat angetriebene Räder und ein drittes, passives Rad als Stütze.

## 3.1. Anforderungen

Neben den grundlegenden Fähigkeiten eines TPR, wie das direkte Manövrieren und Navigieren aus der Distanz sowie die bidirektionale Übertragung von Bild und Ton, werden an das in dieser Arbeit entwickelte System weitere Anforderungen gestellt:

- Bewegliche KE mit Kamera, ausgelegt für Erfassung eines natürlichen Sichtfeldes
- Headtracking, Stereovision & Head-Mounted Display (HMD)
- Vorbereitung für Raumklangaufnahme
- Kapazitäten für autonome Navigation
- Erweiterbarkeit der Soft- & Hardware bezüglich Themen aus dem Umfeld dieser Arbeit (Beispiele in Kapitel 5)
- Sicherheit bezüglich Unfallvermeidung (*Safety*) & Diebstahlschutz (*Security*)

Die HAW Hamburg gibt das differentielle Fahrwerk vor. Es handelt sich dabei um einen *Pioneer 3-DX* des Herstellers *Adept MobileRobots* aus Amherst, New Hampshire, USA.

Die ME soll mit einer beweglichen KE ausgestattet werden. Auf dieser wird ein Kamerasystem für Stereovision montiert. Die KE wird hinreichend Freiheitsgrade besitzen, um die Blickrichtung des Bedieners auf die Kamera zu übertragen.

Die HAW Hamburg stellt dem TPR ein stereovisuelles HMD des Typs *Rift* der Firma *Oculus VR* aus Irvine, Kalifornien, USA zur Verfügung. Dieses HMD ist mit der notwendigen Sensorik für das Headtracking ausgestattet. Die Verwendung der Rift ist Vorgabe der Hochschule.

Zusätzlich zur visuellen Aufnahme soll die ME in der Lage sein, die Raumakustik in der Umgebung zu erfassen. Der Raumklang in Verbindung mit der Stereovision soll dem Bediener eine verstärkte Immersion ermöglichen. Dazu steht seitens der HAW Hamburg mit einem Raumklangmikrofon des Typs *SPS200* der Firma *TSL Professional Products Ltd.* aus Marlow, UK die entsprechende Hardware zur Verfügung. Es ist vorgegeben, die Installation des SPS200 zu ermöglichen.

Die Steuerung der ME ist so zu entwerfen, dass sie über das lokale Netzwerk erfolgt. Steuerbefehle sind an der OPS von geeigneter Eingabehardware entgegenzunehmen und an die ME zu übertragen. Informationen über den Betriebszustand der ME sind dem Benutzer zur Verfügung zu stellen.

Autonome Navigation erfordert erhöhte Rechenleistung gegenüber der direkten Fernsteuerung, bei der Fahrtrichtung und Geschwindigkeit manuell vorgegeben werden. Das **Steuerungssystem der ME (MESS)** soll auf autonome Navigation ausgelegt sein. Dies beinhaltet simultane Lokalisation und Kartenerstellung (**SLAM**) und Hinderniserkennung sowie -vermeidung. Ausreichend Rechenkapazität ist auf der **ME** bereitzustellen.

Der Roboter soll neben der Erweiterbarkeit der Software Raum und Kapazitäten für zusätzliche Hardware besitzen. Diese Zusatzhardware kann beispielsweise Sensorik für autonome Navigation beinhalten.

Da ein sich im öffentlichen Raum bewegendes Gerät unweigerlich mit Menschen in Kontakt kommt, sind Aspekte der Sicherheit bezüglich Diebstahlschutz von Komponenten zu prüfen und Verletzungen von Passanten vorzubeugen.

## 3.2. Gesamtkonzept

Das Gesamtsystem besteht aus der **OPS** und der **ME**. Der Bediener wird die **OPS** in sitzender Position bedienen. Er hat während der Steuerung der **ME** das **HMD** sowie das Headset aufgesetzt und hält die Fernsteuerung in der Hand.

**Abbildung 3.1** zeigt den Aufbau des Konzepts als Diagramm. Mit der grün gestrichelten Linie werden Komponenten gruppiert. Es handelt sich einerseits um die **KE** und andererseits um das Headset. Die grünen Pfeile zeigen die Verbindungen der Komponenten und geben die primäre Richtung des Informationsflusses an.

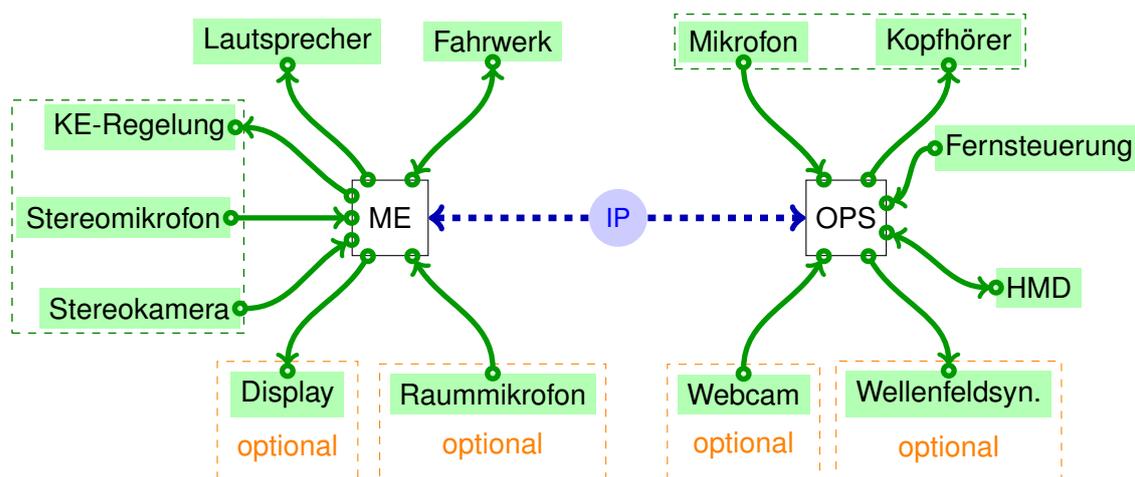


Abbildung 3.1.: Konzeptdiagramm

Das **HMD** zeigt dem Bediener die Bilder der auf der **KE** montierten Kamera. Die Kamera bewegt sich durch die Mechanik der **KE** im Idealfall simultan zur Bewegung des Kopfes des Bedieners.

Auf der **KE** ist ein Stereomikrofon zur Aufnahme von Sprache und Geräuschen in der Umgebung der **ME** montiert. Die aufgenommene Klangkulisse wird an die **OPS** übertragen und dort auf dem Headset ausgegeben. Neben der Ausgabe der akustischen Umgebung der **ME** kann das Headset akustischer Signale zur Informationsübermittlung wiedergeben. Darüber hinaus besitzt das Headset ein Mikrofon, mit dem die Stimme des Bedieners aufgenommen wird. Diese wird digitalisiert an die **ME** übertragen und dort auf den Lautsprechern ausgegeben.

Die Steuerung der **ME** erfolgt über ein entsprechend ausgelegtes, mit der **OPS** gekoppeltes Eingabegerät. Dieses Eingabegerät bietet die Möglichkeit, grundlegende Fahrbefehle zur Richtung und Geschwindigkeit zu erteilen. Darüber hinaus ermöglicht das Gerät, der **ME** weitere Befehle zu geben. Näheres zu möglichen Befehlen ist in [Kapitel 5](#) zu finden.

### 3.3. Bedienerstation

Die **OPS** besteht aus nachfolgend aufgeführten Komponenten:

- PC
- Headset
- Gamepad
- VR-System/**HMD**

Der Zweck der einzelnen Komponenten findet im Verlauf dieses Kapitels Erläuterung.

#### Komponenten

Die Wahl für die Basisstation der **OPS** fällt aus Gründen der leichten und gewohnten Installation und Bedienung auf einen handelsüblichen x86/64bit kompatiblen PC. Geräte dieser Architektur werden vielfach an anderer Stelle im Netzwerk der HAW Hamburg betrieben. Eine Aufnahme der **OPS** in das Netzwerk bereitet daher keine Probleme. Andere Architekturen wie ARM oder Apple sind denkbar, werden jedoch wegen fehlender oder nicht offiziell unterstützter Kompatibilität zur verwendeten Software nicht empfohlen.

Der PC ist mit Programmen für die Fernsteuerung der **ME** ausgestattet. Diese Software erlaubt es dem Bediener, die **ME** zu bewegen und sie mit der Umgebung interagieren zu lassen. Sie enthält Komponenten zur Kommunikation mit Personen im Umfeld der **ME**. Soweit lizenzrechtlich möglich ist die Software quelloffen ausgeführt und darf angepasst werden.

Die Wahl des Headsets ist dem Bediener überlassen. Um bestimmten Produkten den Vorzug zu geben, ist die Auswahl am Markt verfügbarer Systeme schlicht zu groß und die Kriterien für die Auswahl zu abstrakt gesteckt. Das Headset benötigt Komponenten zur Aufnahme und Wiedergabe von Schall, bevorzugt im Frequenzspektrum menschlicher Stimme. Dieses Kriterium erfüllen alle verwendbaren Headsets. Bei dieser Arbeit wird ein vorhandenes Headset verwendet.

Zur Steuerung der **ME** wird ein **Gamepad** verwendet. Diese handlichen Geräte bieten eine Vielzahl an Tasten, die für erweiterte Funktionen verwendet werden können. Näheres zu möglichen Erweiterungen ist im Kapitel **Kapitel 5** beschrieben. Das Gamepad findet Verwendung aufgrund seines zweckmäßigen Aufbaus. Gegenüber einer Tastatur ist es kompakt. Die Auslegung auf die Eingabe von Bewegungsbefehlen ohne **Graphical User Interface (GUI)** ist ein Vorteil gegenüber einer Maus. Die Bedienelemente sind auch mit aufgesetztem **HMD** leicht zu finden.

Voraussetzung für die Steuerung der **ME** ist ein Gamepad mit einem Analog-Stick. Dieses Eingabeelement dient dazu, innerhalb eines Einheitskreises nahezu kontinuierlich jede mögliche Position zu beschreiben und damit von reiner Geradeausfahrt bis zur Drehung auf der Stelle fließend wechseln zu können. Mittels dieses Bedienelements wird der **ME** die Fahrtrichtung und Geschwindigkeit vorgegeben.

Der Zweck des **HMD** besteht in der räumlichen Wahrnehmung digitaler Bilder und der Steuerung der **KE**. Das **HMD** besitzt zwei vor den Augen seines Trägers platzierte Linsen. Auf dem dahinter liegenden Display wird ein Bild der zugehörigen Kamera dargestellt. Das Bild ist passend zu den Linsen des **HMD** verzerrt. Die Linsen entzerren das Bild für den Träger. Sie spannen es auf diese Weise auf ein Sichtfeld, das dem des Menschen hinreichend entspricht. Die zwei Kameras nehmen ihre Umgebung aus dem Augenabstand des Menschen entsprechend unterschiedlichen Perspektiven auf. Die Anzeige eines Bildes für jedes Auge des Trägers ermöglicht räumliche Wahrnehmung der erfassten Umgebung. Die Tiefenwahrnehmung und die Steuerung der Blickrichtung der Kameras durch das **Headtracking** führen zu einer gesteigerten Immersion.

## Software

Die Software auf der [OPS](#) benötigt für die geforderten Grundfähigkeiten des Systems die nachfolgend aufgeführten Fähigkeiten:

- Aufnahme von Bewegungsbefehlen aus dem [Gamepad](#)
- Übertragung von Bewegungsbefehlen an die [ME](#)
- Aufnahme von Headtracking-Daten aus dem [HMD](#)
- Übertragung der Headtracking-Daten an die [ME](#)
- Aufnahme von Audiodaten über das [Headset](#)
- Übertragung der Audiodaten an die [ME](#)
- Ausgabe der Audiodaten von der [ME](#) auf Lautsprechern
- soweit möglich quelloffen

### 3.4. Mobileinheit

Die [ME](#) des [TPR](#) besteht aus folgenden Komponenten:

- elektronisch steuerbares Fahrwerk
- [MESS](#) mit Schnittstelle zum Fahrwerk
- Hardware für Ein- und Ausgabe von Audio und Video
- bewegliche [KE](#)
- passive Tragkonstruktion für die [KE](#)
- Stereokamera und Mikrofon auf der [KE](#)

## Fahrwerk

Das differentielle Fahrwerk der [ME](#) ist nach Vorgabe der Hochschule das Pioneer 3-DX. Details zu diesem System sind in [Abschnitt 4.2](#) aufgeführt.

## Steuerungssystem der ME

Das **MESS** basiert auf x86/64 Architektur. Diese ermöglicht den Einsatz von Standardsoftware. Wie bei der **OPS** wäre auch als **MESS** der Einsatz einer anderen Architektur denkbar, jedoch aus Kompatibilitätsgründen nicht empfohlen. Die Steuerung übernimmt das **ROS** der Firma Willow Garage. Als grundlegendes Betriebssystem für **ROS** wird Ubuntu 12.04 LTS empfohlen [Willow Garage (2014b)]. Außerhalb dieser Arbeit gesammelte Erfahrungen mit **ROS** untermauern die Empfehlung zur Verwendung von Ubuntu 12.04 LTS. Dieses System findet daher Verwendung.

## Stereokamera

Auf der **KE** wird ein Kamerasystem montiert. Zwei Kameras dienen der Aufnahme stereoskopischer Bilder für die räumliche Darstellung der Umgebung über das **HMD**. Sie liefern Bilder, die in ihrer relativen Lage zueinander dem durchschnittlichen menschlichen Augenabstand entsprechen. Es werden unterschiedliche Systeme auf Eignung getestet:

- eine integrierte Stereokamera mit zwei Objektiven
- zwei getrennte Kameras mit jeweils einem Objektiv

Der Nachteil zweier getrennter Kameras wird unter Umständen eine nicht simultane Aufnahme der Bilder sein. Gerade in Verbindung mit einem stereovisuellen **HMD** wie der Rift ist die simultane Aufnahme jedoch erforderlich. Bei schnellen Kopfbewegungen kann es andernfalls zu Differenzen zwischen beiden Augen des Bedieners kommen. Während das ein Auge bereits das neue Bild sieht, bekommt das andere Auge noch das vorherige Bild präsentiert. Je nach Geschwindigkeit der Kopfbewegung können diese Bilder völlig unterschiedlich sein.

Bewegt sich in einem solchen Fall ein Objekt auf die **ME** zu oder von ihr weg, so ist der Effekt möglicherweise als störend zu empfinden. Das Objekt erscheint auf einem Auge weiter weg als auf dem anderen. Gleiches gilt für die Bewegung entlang der vertikalen Bildachse. Dabei erscheint das Objekt auf einem Auge vertikal höher oder tiefer als auf dem anderen Auge. Diese Fehldarstellung wird ebenfalls als störend eingeschätzt.

Schwerwiegender ist eine Bewegung entlang der horizontalen Bildachse, bei der sich ein Objekt quer durch die Sichtfelder der Kameras bewegt. Durch zeitliche Differenz bei der Aufnahme kann das Objekt dem Benutzer näher oder weiter entfernt erscheinen, als es tatsächlich ist. Eine Bewegung quer durch die Sichtfelder ist wahrscheinlicher, da diese bereits beim Drehen des Kopfes um die vertikale Achse auftritt. Ein schneller Schwenk

des Blickfeldes, verbunden mit dem genannten Verzögerungseffekt kann somit zu falschen Tiefeneindrücken führen.

Zusätzliche Hardware zur simultanen Abfrage und Kombination beider Kamerabilder kann dieses Problem lösen. Die praktische Lösung dieses Problems ist nicht Bestandteil dieser Arbeit, mögliche Wege werden jedoch in [Kapitel 5](#) aufgezeigt.

Des Weiteren ist die Art des Blendenverschlusses der Kamera wichtig. Es sollte sich hierbei um einen *global shutter* handeln. Die alternative Variante des Verschlusses wäre ein *rolling shutter*. Das Bild wird beim rolling shutter Zeile für Zeile aufgenommen. Das führt bei schnell bewegten Objekten dazu, dass sich diese in einer Zeile an einer anderen Position befinden als in der nächsten. Dieser Effekt kann bei der Aufnahme von schnell vorbeifahrenden Autos wie in [Abbildung 3.2](#) oder Flugzeugpropellern wie in [Abbildung 3.3](#) beobachtet werden.



Abbildung 3.2.: Rolling-Shutter-Effekt bei einem schnell vorbeifahrenden Auto  
[[Axel1963 \(Wikipedia\)](#) (2014)]



Abbildung 3.3.: Rolling-Shutter-Effekt bei einem Flugzeugpropeller  
[Richmilliron (Wikipedia) (2014)]

Bei einer derart fehlerhaften Aufnahme kippen Objekte bei schneller Kopfbewegung um die Yaw-Achse. Bei schnellen Bewegungen um die horizontale Pitch-Achse erscheinen Objekte länger oder kürzer. Schnelle Bewegungen um die Roll-Achse hätten einen ähnlichen Effekt zur Folge wie die Drehung um die Yaw-Achse, allerdings verbunden mit einer Verbiegung der Objekte. Der global shutter nimmt zum Zeitpunkt des Auslösens das gesamte Bild simultan auf und vermeidet diesen Effekt.

## Display

Zur Anzeige des Gesichts des Bedieners kann an der KE vor der Stereokamera ein Display montiert werden. Zur KE passende Teile zur Befestigung des Displays sind konstruiert, jedoch nicht angefertigt.

Die Position des Displays zwischen den Objektiven wird als optimal betrachtet. Der Gesprächspartner kann dadurch dem Bediener auf dem Display in die Augen und gleichzeitig in Richtung der Kamera sehen. Beim Bediener wird auf diese Weise der Eindruck vermieden, sein Gegenüber schaue an ihm vorbei.

## Raumklang-Mikrofon

Das Wiedergabesystem für den aufgenommenen Raumklang ist wie die OPS fest im Raum installiert. Im Gegensatz zum Headset ändert es seine Orientierung nicht. Aus diesem Grunde ist das Raumklang-Mikrofon unbeweglich an der ME zu montieren. Klangquellen im Umfeld der ME werden dadurch auf dem Wiedergabesystem aus der korrekten Richtung wiedergegeben. Eine Klangquelle hinter der ME wird aus Sicht des Bedieners an der OPS ebenfalls von hinten zu hören sein.

Das Mikrofon und das Wiedergabesystem für den Raumklang sind optional ausgeführt. Daher wird das Mikrofon im Rahmen dieser Arbeit nicht montiert. Sein Einsatz ist in einer weiteren Abschlussarbeit denkbar. Eine Halterung für das Mikrofon ist konstruiert, jedoch nicht angefertigt.

## Gelenksystem für Stereokamera

Ein Gelenksystem ermöglicht die Drehung der Stereokamera auf der ME in drei Freiheitsgraden. Das Gelenksystem muss so gestaltet sein, dass es der Bewegung des menschlichen Kopfes beim Umschauen weitestgehend entspricht. In dieser Arbeit wird das Gelenksystem als KE bezeichnet.

Das menschliche Genick vereint sieben Wirbel, von denen jeder drei Freiheitsgrade besitzt. Rechnerisch ergibt dies 21 Freiheitsgrade. Jedoch ist nicht jeder einzelne Freiheitsgrad aktuiert. Diesen Umstand macht sich die Ausgestaltung der KE zunutze. Sie wird auf drei aktuierte Freiheitsgrade beschränkt.

Ein Versuch zeigt, dass sich für eine natürliche Beweglichkeit der KE eine bestimmte Reihenfolge der drei Achsen ergibt. Im Fachjargon der Luftfahrt ausgedrückt entspricht diese Reihenfolge:

1. Pitch (Nicken)
2. Yaw (Gieren)
3. Roll (Rollen)

Abbildung 3.4 zeigt die Bedeutung der drei Achsen anhand eines Flugzeugmodells. Die Pitch-Achse beschreibt die Neigung des Flugzeugs. Durch Rotation um diese Achse hebt oder senkt sich die Nase des Flugzeugs. Dreht sich das Flugzeug um die Roll-Achse, kippt es zur Seite.

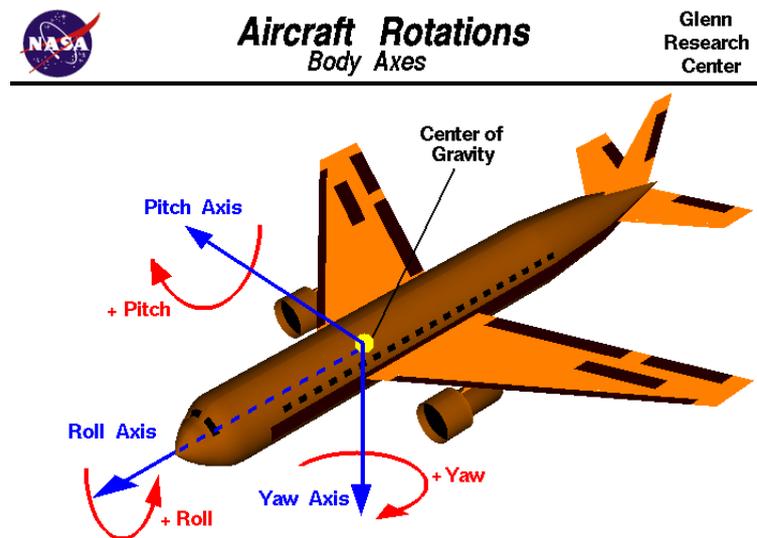


Abbildung 3.4.: Drehachsen aus der Luftfahrt  
[NASA Glenn Research Center (2014)]

Die [Abbildung 3.5](#) zeigt die drei Achsen angewendet auf die menschliche Halswirbelsäule. Die oberen zwei Teilbilder zeigen die Bewegung um die Pitch-Achse, die mittlere Zeile stellt Bewegung um die Yaw-Achse und die untere Zeile um die Roll-Achse dar. Die Illustration der Anordnung der Rotationsachsen ([Abbildung 3.5](#), [Abbildung 3.6](#), [Abbildung 3.7](#)) wurde mit Hilfe der Software *DAZ Studio 4.6* erzeugt.

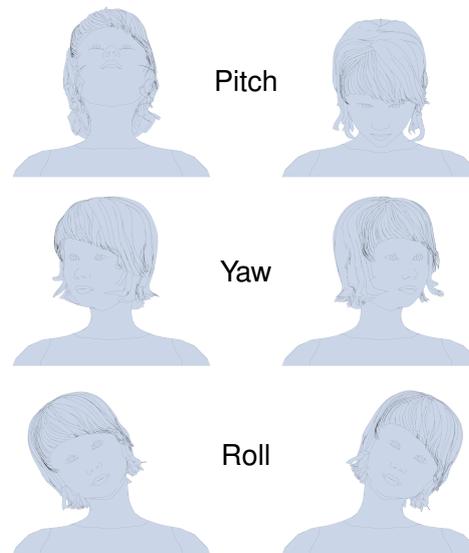


Abbildung 3.5.: Die drei Achsen anhand der menschlichen Halswirbelsäule

Der Grund für die Relevanz der Anordnung ist das Verhalten der jeweils nachgestellten Achse bei Verdrehung ihrer Vorgängerachse. Die Yaw-Achse muss der Pitch-Achse nachgeordnet sein, damit der Bediener mit in den Nacken gelegtem Kopf zu den Seiten schauen kann. Wäre die Yaw-Achse die erste Achse in der Anordnung, so würde dies dazu führen, dass der Bediener zwar den Kopf in den Nacken legen, diesen in dieser Pose jedoch nicht zur Seite drehen kann. Stattdessen würde die KE die Blickrichtung im Gegensatz zum Bediener mit starrem Blick nach oben um seine vertikale Körperachse drehen, während der Bediener tatsächlich zur Seite blickt.

Abbildung 3.6 zeigt die natürliche Anordnung der Achsen, bei der zunächst das Drehen um die Pitch-Achse und nachfolgend um die Yaw-Achse erfolgt. In dieser Kombination ist es möglich, mit in den Nacken gelegtem Kopf zu den Seiten zu schauen. Die Roll-Achse in der Anordnung ermöglicht eine Haltung ähnlich der in [Abbildung 3.7](#).

In [Abbildung 3.7](#) ist zu sehen, dass bei inkorrektter Anordnung zwar der gerade Blick zur Decke mit der korrekten Anordnung übereinstimmt. Eine inkorrekte Anordnung der Achsen führt jedoch bei weiterer Drehung um die Yaw-Achse dazu, dass sich das Blickfeld lediglich um die vertikale Körperachse verdreht, anstatt zur jeweiligen Seite zu kippen. Die Yaw-Achse verbleibt parallel zur vertikalen Körperachse. Ein Blick zur Seite ist nicht mehr möglich.



Abbildung 3.6.: pitch→yaw



Abbildung 3.7.: yaw→pitch

Abgesehen von der natürlichen Bewegung der **KE** resultiert der Einfluss einer Achse auf die nachfolgenden Achsen unter gewissen Umständen in einer **Singularität**, dem sogenannten *Gimbal Lock*. Dieser Fall tritt ein, wenn sich eine der drei Achsen in dieselbe Ebene bewegt wie eine andere. Die Überlagerung zweier Achsen führt dazu, dass ein Freiheitsgrad verloren geht.

Die in Solidworks erzeugte [Abbildung 3.8](#) zeigt im linken Abschnitt die Ausrichtung der drei Achsen in Grundorientierung. In dieser Lage sind Drehungen um alle drei Achsen möglich. Dreht sich die Yaw-Achse um  $90^\circ$ , wie im rechten Abschnitt zu sehen, so überlagern sich die Achsen Pitch und Roll. Rotationen um Pitch und Roll haben in dieser Stellung denselben Effekt. Eine Rotation um die ursprüngliche Roll-Achse ist nicht mehr möglich.

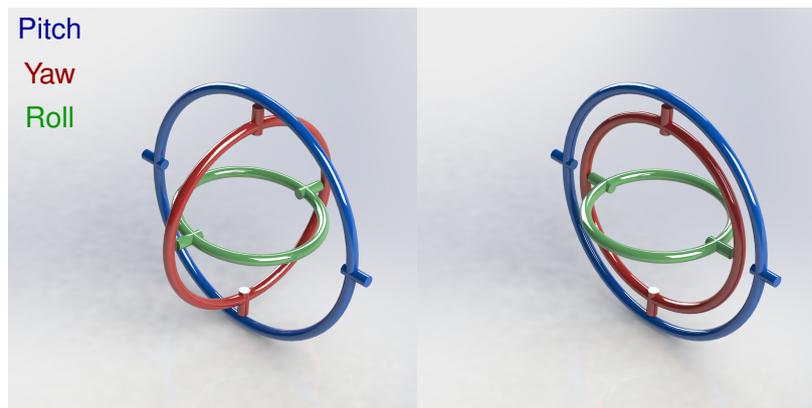


Abbildung 3.8.: links: Achsen in Grundausrichtung; rechts: Pitch und Roll in Gimbal Lock

In der gewählten Achsreihenfolge Pitch-Yaw-Roll tritt diese Singularität auf, wenn der Bediener  $90^\circ$  zur Seite schaut. Dadurch bewegt sich die Roll-Achse in die gleiche Ebene wie die Pitch-Achse, wodurch beide dieselbe Bewegung bewirken. Der Arbeitsraum des menschlichen Genicks ist nicht darauf ausgelegt, den Kopf  $90^\circ$  zur Seite zu drehen [[Teaching\\*TheHumanBrain \(2014\)](#)]. Ein darauf aufbauender Blick über die Schulter herab entlang der Körperseite ist nicht vorgesehen. Diese Singularität lässt sich daher vernachlässigen.

Der Aufbau der Mechanik der KE erfolgt individuell für diesen Einsatzzweck. Pan-Tilt-Systeme für das aktuierte Drehen und Schwenken von Kameras sind fertig verfügbar. Diese Systeme bieten zwei Freiheitsgrade mit den Achsen Yaw und Pitch. Der zusätzliche Freiheitsgrad der KE ermöglicht die Darstellung rudimentärer Gestik. Skepsis oder Unverständnis kann beispielsweise durch schräg legen, Abwägen durch wechselseitiges Neigen des Kopfes vermittelt werden. Der dritte Freiheitsgrad wird für die natürliche Bewegung daher als notwendig erachtet. Ein weiteres Defizit fertiger Systeme ist die fehlende Rückmeldung über die aktuelle Orientierung des Kamerakopfes. Die Software für die optische Aufbereitung der Bilder benötigt diese Information. Aus diesen Gründen fällt die Entscheidung zugunsten einer Eigenkonstruktion, die diese Nachteile nicht besitzt. Die Fertigung der zugehörigen Teile ist dank des an der Hochschule verfügbaren 3D-Druckers einfach und kostengünstig durchzuführen.

## Software

Die Steuerung der beweglichen Komponenten und die Aufnahme der Sensordaten der **ME** übernimmt eine zu entwickelnde Software. Die erforderlichen Grundfähigkeiten und -aspekte der Software bestehen aus nachfolgend genannten Funktionalitäten:

- Aufnahme von Bewegungsbefehlen von der **OPS**
- Übersetzung von Bewegungsbefehlen in Steuerbefehle für die Fahrplattform
- Aufnahme von Headtracking-Daten von der **OPS**
- Übersetzung der Headtracking-Daten in Steuerbefehle für die **KE**
- Ausgabe von Audio- und Videodaten von der **OPS** auf die Ausgabehardware
- Aufnahme von Audio- und Videodaten über die Eingabehardware
- Übertragung von Audio- und Videodaten an die **OPS**
- soweit möglich quelloffen

Darüber hinaus ist die Software so auszulegen, dass sie außerhalb dieser Arbeit folgende Fähigkeiten erlangen kann:

- **SLAM** und autonome Navigation
- Stereovision über eine Stereokamera bzw. zwei Einzelkameras
- Raumklangaufnahme
- Erweiterbarkeit bzgl. Sensorik und Aktorik

## 4. Aufbau des Telepräsenzroboters

Das vorige Kapitel beschäftigt sich mit den Anforderungen an das Gesamtsystem und dessen Komponenten. Darüber hinaus behandelt es die Ausarbeitung des Konzepts. In diesem Kapitel erfolgt die Beschreibung der technischen Umsetzung dieses Konzepts in Form der **ME** und der **OPS** sowie vorhandene Komponenten. Die Konstruktion der Teile der **KE** wird mit Hilfe der CAD-Software Solidworks 2013 der Firma Dassault Systèmes aus Frankreich durchgeführt.

### 4.1. Grundlegender Aufbau

Abbildung 4.1 zeigt die **ME** in Frontansicht. Die Kennziffern werden in **Tabelle 4.1** benannt.

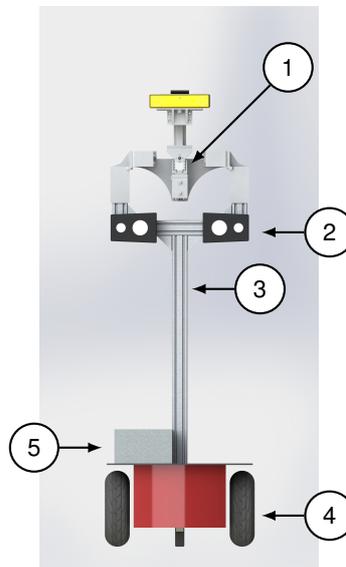
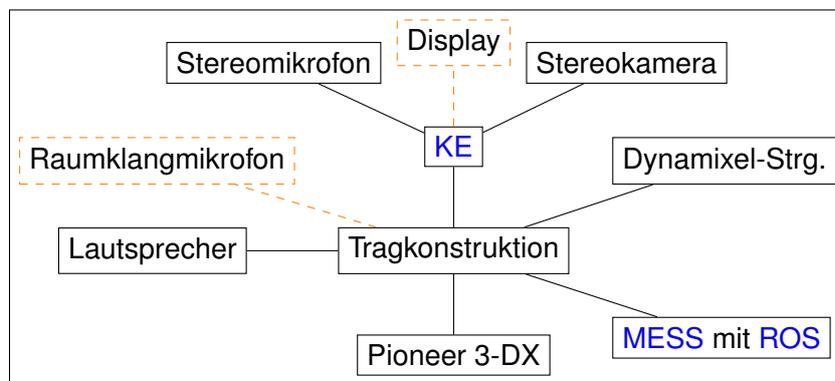
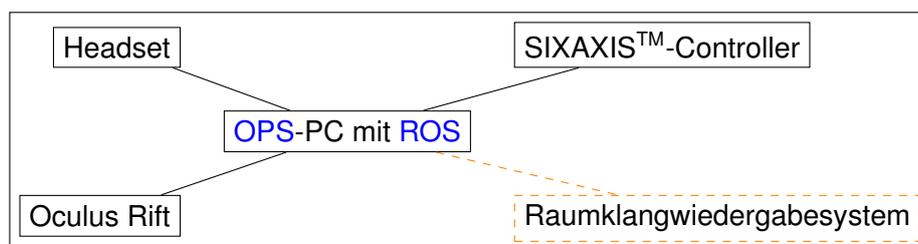


Abbildung 4.1.: Mobileinheit in Frontansicht

Tabelle 4.1.: Kennzeichnungen in [Abbildung 4.1](#)

1	KE mit Stereokamera und Stereomikrofon
2	USB-Aktivlautsprecher
3	Tragkonstruktion
4	Fahrwerk Pioneer 3-DX
5	MESS Intel DH57JG Mini-ITX

Schematisch stellen [Abbildung 4.2](#) und [Abbildung 4.3](#) den Aufbau von **ME** und **OPS** dar. Details zu den einzelnen Komponenten folgen im Verlauf dieses Kapitels.

Abbildung 4.2.: Schematische Darstellung der **Mobileinheit**Abbildung 4.3.: Schematische Darstellung der **Bedienerstation**

Das Kamerasystem nimmt stereoskopische Bilder der Umgebung der **ME** auf, um sie auf dem **HMD** darzustellen. Es steht eine Stereokamera des Typs *Bumblebee* der Firma *Point Grey Research Inc.* aus Richmond, Kanada zur Verfügung. Der Einsatz von zwei Webcams wird alternativ geprüft. Die Webcams sind vom Typ *C310* des Herstellers *Logitech AG* aus Apples, Schweiz.

Damit Bewegungen der Kamera simultan zur Kopfbewegung des Bedieners ausgeführt werden können, wird die elektrisch aktuierte **KE** (1) konstruiert und angefertigt. Zur Konstruktion wird die CAD-Software Solidworks 2013 verwendet.

Ein Aufbau ist erforderlich, der **KE** und Lautsprecher (2) trägt. Die Tragkonstruktion (3) wird aus Aluminiumprofilen aufgebaut und auf dem Fahrwerk (4) fixiert. Die Masse der Tragkonstruktion ist gering gehalten, um einen tiefen Schwerpunkt der **ME** zu bewahren. Dies vermeidet Kippen bei Beschleunigung.

Betriebssystem auf **MESS** (5) und **OPS** ist ein 64bit Linux. Das Betriebssystem ist für sich nicht ohne Weiteres in der Lage, die proprietäre Hardware der **ME** zu steuern. Daher findet mit **ROS** ein auf Linux basierendes System von Softwarebibliotheken zur Robotersteuerung Verwendung. **ROS** konnte bereits in zahlreichen anderen Projekten außerhalb dieser Arbeit seine Fähigkeiten zur Steuerung autonomer Robotersysteme unter Beweis stellen [[Willow Garage \(2014a\)](#)]. Außerhalb der Arbeit gesammelte Erfahrungen des Autors bestätigen den Einsatz des Systems.

Für die Kommunikationsübertragung wird eine Voice-over-IP-Software eingesetzt. Das Programm basiert auf dem [Session Initiation Protocol \(SIP\)](#)-Protokoll. Es läuft auf dem **MESS** und der **OPS**. Auf dem **MESS** ist es so konfiguriert, dass es eingehende Anrufe automatisch entgegen nimmt. Die SIP-Software besitzt keine Fähigkeiten zur Fernsteuerung der **ME**. Sicherheitsaspekte bezüglich der Authentifizierung und Autorisierung von Anrufern werden daher in dieser Arbeit nicht betrachtet. Es ist für die Weiterentwicklung zu beachten, dass ein nicht autorisierter Anrufer das Gerät unter Umständen zum Mithören nutzen kann.

## 4.2. Im Department verfügbare Komponenten der **Mobileinheit**

Eine detaillierte Beschreibung der Komponenten der **ME** erfolgt in diesem Kapitel. Es werden Vor- und Nachteile beleuchtet und Spezifikationen genannt. Für die Nutzung von nicht durch die Hochschule vorgegebene Komponenten werden Begründungen gegeben.

### Fahrwerk: *Pioneer 3-DX*

Das Pioneer 3-DX ([Abbildung 4.4](#)) ist ein differentielles Fahrwerk. Es wird mittels RS232-Schnittstelle gesteuert.



Abbildung 4.4.: Pioneer 3-DX von Adapt MobileRobots  
[[Smashing Robotics \(2014\)](#)]

Der Pioneer 3-DX stellt Drehimpulsgeber an den angetriebenen Rädern zur Verfügung. Diese sind für autonome Navigation einsetzbar. Zur Hinderniserkennung besitzt der Pioneer 3-DX insgesamt acht Ultraschall-Distanzsensoren. [[Adept MobileRobots \(2014b\)](#)]

Aufgrund seiner Luftbereifung kann der Pioneer 3-DX Stufen von 2,5 cm sowie Spalten von 5 cm überwinden. Mit einer Höchstgeschwindigkeit von  $1,6 \text{ m s}^{-1}$  ist er hinreichend schnell. Er kann ein Gewicht von 23 kg tragen. Seine elektrische Energie bezieht der Pioneer 3-DX aus bis zu drei 12 V Bleigel-Akkumulatoren. Die Akkumulatoren können, sofern mindestens einer im Gerät verbleibt, im laufenden Betrieb getauscht werden. [[Adept MobileRobots \(2014b\)](#)]

Um dem Betrieb der [KE](#) und des [MESS](#) ausreichend Energie zur Verfügung zu stellen, ist der Einsatz von mindestens zwei Bleigel-Akkumulatoren mit jeweils 7,2 Ah maximaler Ladung erforderlich. Der eingesetzte Akkumulator ist ein *LC-R127R2PG* der Marke *Panasonic*.

## Servomotoren

Die für die KE verwendeten Servomotoren sind vom Typ *Dynamixel AX-12A* der Firma *Robotis* aus Seoul, Südkorea. Sie liefern laut Herstellerangabe bei einer Betriebsspannung von 12 V ein Stillstandsmoment von 1,5 Nm und eine maximale Drehzahl von  $59 \text{ min}^{-1}$  [Robotis (2014)]. Eine Betriebsspannung von 12 V wird nicht empfohlen, da die Motoren erfahrungsgemäß trotz Einhaltung der Betriebsspezifikation bei dieser Spannung dazu neigen, unter Last durchzubrennen. Aus diesem Grunde wird auf eine Betriebsspannung von 9 V gesetzt. Eine herabgesetzte Betriebsspannung resultiert in der Verringerung des Stillstandsmoments und der maximalen Drehzahl der Motoren.

Die Spannungsversorgung ist in Form eines Tiefsetzstellers ausgeführt. Die Ausgangsspannung wird mittels eines Trim-Potentiometers angepasst. Dadurch sind Versuche mit höheren Spannungen möglich. Die angegebenen 9 V werden für die Motoren als sicher eingestuft.

Für den Proof of Concept (PoC) ohne montierte Kameras sind diese Motoren bei einer Betriebsspannung von 9 V ausreichend stark und schnell. Es wird empfohlen, bei der Weiterentwicklung des TPR stärkere Motoren des selben Herstellers zu verwenden. Eine Anpassung und erneute Fertigung der betroffenen, in Abschnitt A auf Seite 101 aufgelisteten Elemente der KE wird dadurch gegebenenfalls erforderlich. Als Alternative ohne Anpassungen der Elemente der KE wäre der Servo *Dynamixel AX-18A* zu nennen. Der Motor liefert nach Herstellerangaben bei einer Betriebsspannung von 12 V ein Stillstandsmoment von 1,8 Nm. Er kann eine maximale Drehzahl von  $93 \text{ min}^{-1}$  erreichen [Robotis (2014)]. Dieser Servo passt in dieselben Aufnahmen wie der Dynamixel AX-12A. Der Grund für den Einsatz der Dynamixel AX-12A liegt in der zum Zeitpunkt der Verfügbarkeit an der Hochschule.

## Prüfung Stereokamera

Für die Verwendung als Stereokamera stehen zwei Systeme zur Verfügung. Diese werden im Verlauf des Aufbaus auf ihre Eignung getestet.

- monolithische Stereokamera Typ *Bumblebee* von Point Grey Research Inc., Kanada
- zwei USB-Webcams Typ *C310* von Logitech AG, Schweiz

### Bumblebee

Die in Abbildung 4.5 gezeigte Stereokamera ist eine *Bumblebee* des Herstellers *Point Grey Research Inc.* aus Richmond, Kanada. Ein Vorteil der *Bumblebee* gegenüber zwei

getrennten Kameras liegt in simultaner Aufnahme beider Bilder. Die simultane Auslösung bei zwei oder mehr Kameras kann durch einen Triggereingang erfolgen, sofern die Kameras diesen zur Verfügung stellen. Ein weiterer Vorteil ist die Montierbarkeit. Die Kamera wird mittels einer Schraube des Typs 1/4"-20 UNC mit einer Länge von 20 mm befestigt und mit zwei Zylinderstiften EN ISO 8734-A-3x10-St gegen Verdrehung gesichert. Nachteil der Kamera ist das vergleichsweise hohe Gewicht von 342 g.



Abbildung 4.5.: Bumblebee von Point Grey Research Inc.  
[[Point Grey Research Inc \(2014\)](#)]

Die Inbetriebnahme der Bumblebee schlug fehl. Der Grund hierfür ließ sich im Rahmen dieser Arbeit nicht ermitteln. Darüber hinaus ist sie für die in dieser Arbeit entwickelte KE zu schwer. Sie scheidet damit als Kamera für den Betrieb auf der KE aus.

### Logitech C310

Im zweiten Versuch werden zwei Webcams des Typs *C310* der Firma *Logitech AG* aus Apples, Schweiz getestet ([Abbildung 4.6](#)). Die Wahl fällt auf diesen Typ aufgrund der technischen Daten und des Anschaffungspreises. Darüber hinaus ist die Kamera kompatibel zu Linux.



Abbildung 4.6.: Webcam Logitech C310  
[[Logitech AG \(2014\)](#)]

Laut Produktbeschreibung arbeitet die Kamera mit einem horizontalen Sichtfeld von  $60^\circ$ . Der eingebaute Chip liefert eine hinreichend hohe Auflösung von  $1280 \times 720$  Pixeln. Angeschlossen und mit Energie versorgt werden die Kameras mittels USB 2.0. Die aufgenommene Bildrate liegt bei 30 FPS je Kamera. [[Logitech AG \(2014\)](#)]

Vorteil der Webcams gegenüber der Stereokamera ist eine höhere Auflösung und eine geringere Masse. Ein Nachteil ist die Montierbarkeit. Die Kamera sieht keine Montage durch Verschraubung vor. Eine provisorische Montage wird durchgeführt. Die Konstruktion eines zur Kamera passenden Halters wird empfohlen. Die Kameras werden ebenfalls als Übergangslösung betrachtet. Der Einsatz von passenden Industriekameras wird empfohlen. Der Halter für die Webcams ist daher nicht Teil dieser Arbeit.

## Inbetriebnahme des Kamerasystems

Die Inbetriebnahme der Webcams erfolgt problemlos. Der [Kernel](#) des Betriebssystems erkennt die Kameras ohne Installation zusätzlicher Treibersoftware.

Die Reihenfolge, in der Linux beim Bootvorgang Geräte erkennt, ist nicht deterministisch. Abhängig davon, welche Kamera Linux zuerst findet, kann die linke Kamera als `/dev/video0` und die rechte als `/dev/video1` im System bereitgestellt sein, oder umgekehrt. Die Verknüpfung des Pfades mit einer bestimmten Kamera ist nicht sicher vorherzusagen.

Zur Lösung dieses Problems besitzt Linux ein System zur Verwaltung von Gerätedateien, genannt *udev*. Mit Hilfe von Regeldateien kann *udev* Geräte an festen Stellen im Gerätepfad von Linux verfügbar machen. Diese Regeln werden genutzt, um die Kameras im Pfad `/dev/stereocam/left` und `/dev/stereocam/right` bereitzustellen. Eine deterministische Einbindung der Kameras wird anhand eines festen, individuellen Kriteriums erreicht. Jeder USB-Port des Systems besitzt eine eindeutige ID. Anhand dieser ID wird ermittelt, welche Kamera für das linke und das rechte Bild zuständig ist. Die Regeln für *udev* sind in der Datei `/etc/udev/rules.d/60-tpr.rules` festgelegt. [Listing 4.1](#) zeigt den Inhalt dieser Datei.

Listing 4.1: `/etc/udev/rules.d/60-tpr.rules`

```
KERNEL=="hidraw*", SUBSYSTEM=="hidraw", MODE="0664", \
    GROUP="plugdev"
KERNEL=="video*", KERNELS=="2-1.1", MODE="0664", \
    GROUP="plugdev", SYMLINK="stereocam/left"
KERNEL=="video*", KERNELS=="2-1.2", MODE="0664", \
    GROUP="plugdev", SYMLINK="stereocam/right"
```

Die linke Kamera hängt an der USB-Bus-ID 2-1.1, die rechte an der ID 2-1.2. Diese Portvergabe ist einzuhalten. Die zugriffsberechtigte Benutzergruppe wird über diese Datei auf *plugdev* festgelegt. Die erste Zeile der Datei legt die Benutzergruppe für das [HMD](#) ebenfalls auf *plugdev* fest.

Befestigt sind die Kameras für den Testlauf auf der [KE](#) jeweils durch den mitgelieferten Clip. Dieser ist mechanisch kompatibel zu dem auf die Bumblebee ausgelegten Kameraträger. Da die C310 an diesem Clip händisch vertikal schwenkbar ist, werden die zwei Kameras manuell aufeinander ausgerichtet. Diese Ausrichtung wird voraussichtlich wiederholt erfolgen müssen. Sofern der Einsatz der Kameras fortgesetzt werden soll, wird die Herstellung einer passenden Halterung empfohlen. Für den [PoC](#) wird das Provisorium beibehalten.

Aufgrund der Unterschiede im Sichtfeld der Kameras und der Rift sind die aufgenommenen Bilder mittels entsprechender Filter auf die Optik der Rift anzupassen. Diese Aufgabe ist Teil einer parallel erarbeiteten Bachelor Thesis.

### 4.3. Angefertigte Komponenten der **Mobileinheit**

Ein Teil der **ME** besteht aus den in der vorigen Sektion genannten käuflich erhältlichen Komponenten. Die verbliebenen Komponenten werden konstruiert und angefertigt. Darunter fallen die Tragkonstruktion für die **KE**, die **KE** selbst sowie der Kabelbaum.

#### Tragkonstruktion

Der Bediener bedient die **OPS** hauptsächlich in sitzender Position. Auf dieser Grundlage wird eine Tragkonstruktion für die **KE** entworfen. Die Tragkonstruktion erreicht eine Augenhöhe über dem Boden von etwa 1,2 m. Diese Höhe entspricht zufriedenstellend genau der Augenhöhe eines durchschnittlichen Menschen, während er auf einem handelsüblichen Bürostuhl sitzt.

Das Fahrwerk Pioneer 3-DX hat eine Höhe von 215 mm. Hinzu kommt die Höhe der **KE** selbst, in einer Lage, in der die Kamera parallel zum Boden in Fahrtrichtung nach vorn blickt. Der vertikale Abstand zwischen Sattellebene<sup>1</sup> ([Abbildung 4.7](#) und [Abbildung 4.8](#)) und Mittelpunkt der Kameraobjektive in dieser Lage beträgt etwa 195 mm. Es verbleibt somit eine Gesamthöhe der Tragkonstruktion von etwa  $1200 \text{ mm} - (215 \text{ mm} + 195 \text{ mm}) \approx 790 \text{ mm}$ .

Aufgrund von Variationen in Sitzhöhe und Körpergröße des Operators wird eine Genauigkeit auf 100 mm für die Höhe der Tragkonstruktion als ausreichend angenommen. Dementsprechende Fertigungstoleranz für die zentrale vertikale Strebe wird akzeptiert. Die Tragkonstruktion besteht aus  $40 \text{ mm} \times 40 \text{ mm}$  Aluminiumprofilen der Firma *Rexroth* in unterschiedlichen Längen. Dieses Maß wird gewählt, um einen Kompromiss aus Platzbedarf, Gewicht und Stabilität zu erreichen. Die Tragkonstruktion trägt die Komponenten **KE**, die Lautsprecher für die Sprachausgabe sowie optional das Raumklang-Mikrofon.

#### Kopfeinheit

Die **KE** ist ein in drei Freiheitsgraden bewegliches Trägersystem für die Stereokamera und das Stereomikrofon. Eine entsprechende Konstruktion ermöglicht das Nachempfinden der natürlichen Kopfbewegung.

Die visuelle Wahrnehmung des Bedieners kann durch Abweichungen in der Bewegung seines eigenen Kopfes und der Bewegung der **KE** unangenehm beeinträchtigt werden. Stimmt die Trajektorie der Kameras nicht mit der Trajektorie der Augen des Bedieners überein, kann dies als Schwanken der Umgebung wahrgenommen werden. Dies führt in

---

<sup>1</sup>Aufsetzpunkt der Aluprofile in der Schulter

Abhängigkeit von der Verfassung des Bediener unter Umständen zu Symptomen der **Motion Sickness**. Die Ausführung der KE in drei Achsen soll diesem Effekt entgegenwirken und die Komplexität des Aufbaus gering halten.

**Abbildung 4.7** und **Abbildung 4.8** zeigen die Konstruktion der KE in der Front- und Seitenansicht mit Kennzeichnung der Rotationsachsen. In **Tabelle 4.2** sind die Kennziffern der Abbildungen benannt.

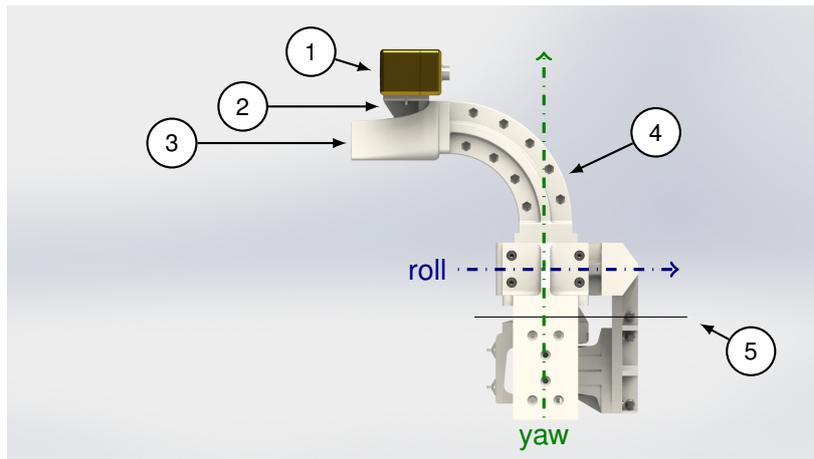


Abbildung 4.7.: Kopfeinheit Seitenansicht links

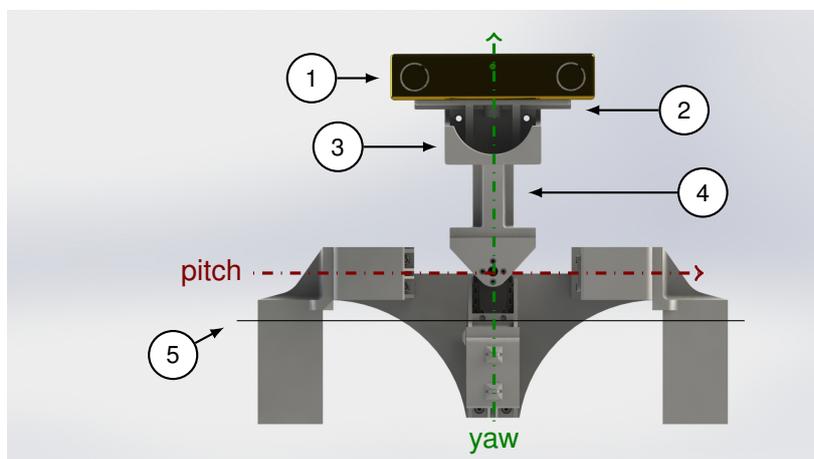


Abbildung 4.8.: Kopfeinheit Frontansicht

Tabelle 4.2.: Kennzeichnungen in [Abbildung 4.7](#) und [Abbildung 4.8](#)

1	Kamera <i>Bumblebee</i>
2	Kameraträger
3	Displayträger
4	starrer Hals
5	Sattalebene

Die Achsverteilung erfordert den Aufbau eines Systems aus Trägern. Das System erreicht in der vorgesehenen Kombination ein Schneiden aller drei Achsen in einem Punkt. Um diesen Punkt bewegt sich der Kamerakopf.

Die Einzelteile der **KE** sind mittels Schrauben des Typs M4x16 nach DIN 6912 miteinander verschraubt, die Servos geben Verschraubungen mit Kreuzschlitzschrauben der Größe M2 in unterschiedlichen Längen vor und die Flansche der Teile sind weitestgehend zueinander kompatibel ausgeführt. Einige Teile der **KE** finden mehrfach Verwendung.

In [Abbildung 4.7](#) und [Abbildung 4.8](#) ist der optionale Displayhalter (3) unterhalb der Kamera (1) zu sehen. In der Grundausführung ist dieser nicht montiert. Der Halter ist dergestalt konstruiert, dass er zwischen die Verbindungsplatten des Kameraträgers (2) und der Halseinheit (4) passt. Zur Montage ist die Verwendung von Zylinderschrauben mit Innensechskant nach DIN 6912 in M4x20 anstelle von M4x16, erforderlich. Das Display, das Stereomikrofon und die Verkabelung sind in den Abbildungen nicht dargestellt.

Die Teile der **KE** werden in einem an der Hochschule verfügbaren 3D-Drucker des Typs *Mojo* des Herstellers *Stratasys, Ltd.* aus Edina, Minnesota, USA gefertigt. Sie bestehen aus dem Kunststoff ABS. Die Teile sind primär auf die Herstellung im 3D-Druckverfahren **FFF** optimiert. Persönliche Erfahrungen zeigen, dass für den Druck in diesem Verfahren der Aufbau des jeweiligen Teils mit mindestens einer größeren Fläche optimal ist. Die Fläche wird während des Drucks als Standfläche genutzt.

Überhänge sind bei der Konstruktion zu vermeiden. Sie können nicht ohne Stützmaterial in den leeren Raum gedruckt werden. Überhängende Schrägen ab 45° zum Druckbett sind ohne Stützmaterial möglich.

Aus diesem Grunde ist beispielsweise das starre Halselement in zwei Teilen ausgeführt. Das Element besitzt ein kreuzförmiges Profil. Dadurch existiert keine Lage, in der es ohne Überhänge zu drucken wären. Die Teilung des Elements in zwei Teile entlang einer Mittelebene ermöglicht den Druck ohne Überhang. Auf der anderen Seite erfordert es die mechanische Verbindung beider Teile.

Entsprechendes gilt für Bohrungen. Sie sind bevorzugt senkrecht zum Druckbett auszuführen. Parallel zum Druckbett ausgeführte Bohrungen führen zu Überhängen.

Sie lassen sich bei der Konstruktion der KE nicht vermeiden. Für Fälle, in denen Überhänge unvermeidbar sind, stellt der genutzte Drucker Stützmaterial zur Verfügung. Das Stützmaterial wird nach dem Druck chemisch aus dem Erzeugnis ausgewaschen.

Die Größe und Stabilität der KE berücksichtigt das Gewicht der zunächst für den Einsatz als Stereokamera geplanten Bumblebee und gegebenenfalls nachträglich eingesetzter Industriekameras sowie Größe und Gewicht der eingesetzten Servomotoren. Damit sich die Kamera in einem näherungsweise der menschlichen Anatomie entsprechenden Abstand zu den Rotationsachsen der KE bewegt, ist sie exzentrisch angebracht. Aus diesem Grunde ist die KE mit einem starren Halsteil (4) versehen. Der Kameraträger ist an dem Halsteil montiert. Die statische und dynamische Auslegung der Mechanik der KE erfolgte anhand von Erfahrungswerten und Schätzungen bezüglich der in erster Näherung ermittelten Belastungsfälle. Für die nachfolgend näherungsweise berechneten Belastungen und Momente wird die Montage der Bumblebee als Grenzfall angenommen, da die Gesamtmasse der zwei Webcams geringer ist.

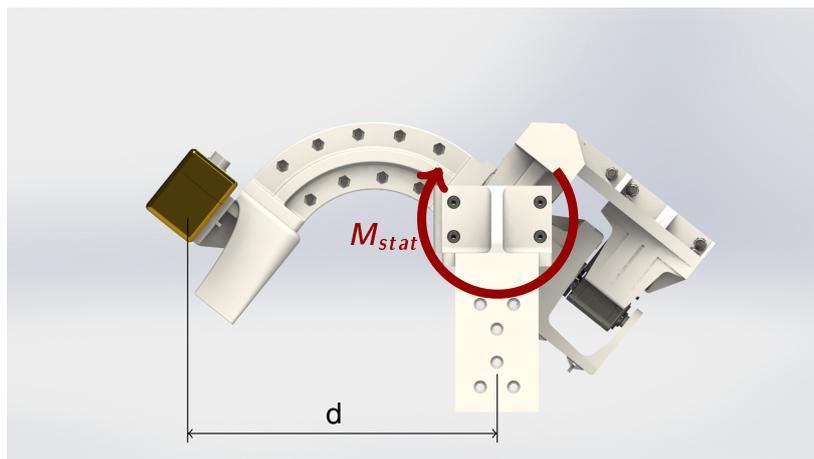


Abbildung 4.9.: Lage maximaler Belastung für einzelnen Servo

Die höchste Last erfährt der Servo am rechten Schultergelenk, sobald die Kamera aus der in [Abbildung 4.9](#) gezeigten Lage in die Lage aus [Abbildung 4.7](#) gebracht werden soll. Unbeweglich stellt sich in dieser Lage das statische Drehmoment  $M_{stat}$  am Motor ein, das wie folgt errechnet wird:

$$\begin{aligned} M_{stat} &= G_{Kamera} \cdot d \\ &= 3,5 \text{ N} \cdot 0,190 \text{ m} \\ &\approx 0,7 \text{ Nm} \end{aligned}$$

Das Eigengewicht der KE und des Yaw-Servos werden in der ersten Näherung vernachlässigt. Die Kamera wird als Massepunkt mit Masse  $m_{Kamera} = 0,342 \text{ kg}$  in einer Entfernung zur Pitch-Achse von  $d = 190 \text{ mm}$  angenommen. Sobald Bewegung einsetzt, kommt das Massenträgheitsmoment  $J_{Kamera}$  der Kamera hinzu. Dieses wird ebenfalls als exzentrischer Massepunkt der Masse  $m_{Kamera}$  angenähert. Das Trägheitsmoment  $J_{Kamera}$  errechnet sich unter diesen Annahmen wie folgt:

$$\begin{aligned} J_{Kamera} &= m_{Kamera} \cdot d^2 \\ &= 0,342 \text{ kg} \cdot (0,190 \text{ m})^2 \\ &= 0,01 \text{ kg m}^2 \end{aligned}$$

Das Moment  $M_{ges}$  darf das Stillstandsmoment  $M_{max}$  des Servos nicht überschreiten. Hieraus lässt sich nun die maximal mögliche Winkelbeschleunigung  $\alpha$  annähern:

$$\begin{aligned} M_{max} &= 1,5 \text{ Nm} \\ M_{ges} &\approx 0,01 \text{ kg m}^2 \cdot \alpha + 0,7 \text{ Nm} \\ M_{ges} &< 1,5 \text{ Nm} \\ \Rightarrow 0,01 \text{ kg m}^2 \cdot \alpha + 0,7 \text{ Nm} &< 1,5 \text{ Nm} \\ \Rightarrow 0,01 \text{ kg m}^2 \cdot \alpha &< 0,8 \text{ Nm} \\ \Rightarrow \alpha &< \frac{0,8 \text{ Nm}}{0,01 \text{ kg m}^2} \\ \Rightarrow \alpha &< 80 \text{ s}^{-2} \end{aligned}$$

Theoretisch kann der Servo den mit der Bumblebee versehenen Kameraträger demnach mit bis zu  $80 \text{ s}^{-2}$  beschleunigen. Dieser Wert wird in praktischen Versuchen nicht erreicht. Reibung in den Lagern und die Masse der KE spielen eine nicht vernachlässigbare Rolle. Darüber hinaus bezieht sich die Herstellerangabe des Stillstandsmoments der Servos auf eine Betriebsspannung von 12 V.

Es wird daher angenommen, dass die Servomotoren nahe an ihrem Stillstandsmoment bei 9 V Betriebsspannung betrieben werden und wiederholt empfohlen, die Motoren durch stärkere Typen zu ersetzen.

## Verbindung zwischen Bedienerstation und Mobileinheit

Die Verbindung der Geräte untereinander findet mittels WLAN statt, damit sich die ME vollständig von externen Leitungen befreit bewegen kann. Das zwischen den Geräten verwendete WLAN ist zunächst das interne WLAN der Fakultät Technische Informatik der HAW Hamburg.

Beide Geräte sind grundsätzlich in jedem WLAN nach IEEE 802.11 einsetzbar, sofern Zugangsberechtigung hergestellt werden kann. Für eine flüssige Übertragung von Bild und Ton muss ausreichend Bandbreite zur Verfügung stehen. Diese variiert stark, je nach verwendetem Audio-/Video-Codec. Als Mindestanforderung für einen flüssigen Betrieb wird ein WLAN nach IEEE 802.11g empfohlen. Die Auswahl des Codecs und die daraus resultierende erforderliche Bandbreite ist nicht Teil dieser Arbeit.

## Leitungsnetz

Die einzelnen Komponenten der beiden Einheiten sind anhand des in Abbildung 4.10 dargestellten Leitungsschemas miteinander verbunden. Bei der OPS nicht dargestellt sind Tastatur, Maus, Monitor und Energieversorgung. Darüber hinaus ist die Energieversorgung der einzelnen separat versorgten Komponenten, wie beispielsweise der Oculus Rift, nicht dargestellt.

Der KE-Controller und die Steuerungseinheit der ME werden vom Akkumulator des Pioneer 3-DX mit Energie versorgt. Der Pioneer 3-DX stellt eine Betriebsspannung von 12 V zur Verfügung. Diese kann an sechs Schraubklemmen unter der Serviceklappe abgegriffen werden. Die Betriebsspannung für die Servomotoren der KE von 9 V erzeugt ein Tiefsetzsteller des Typs LM2596.

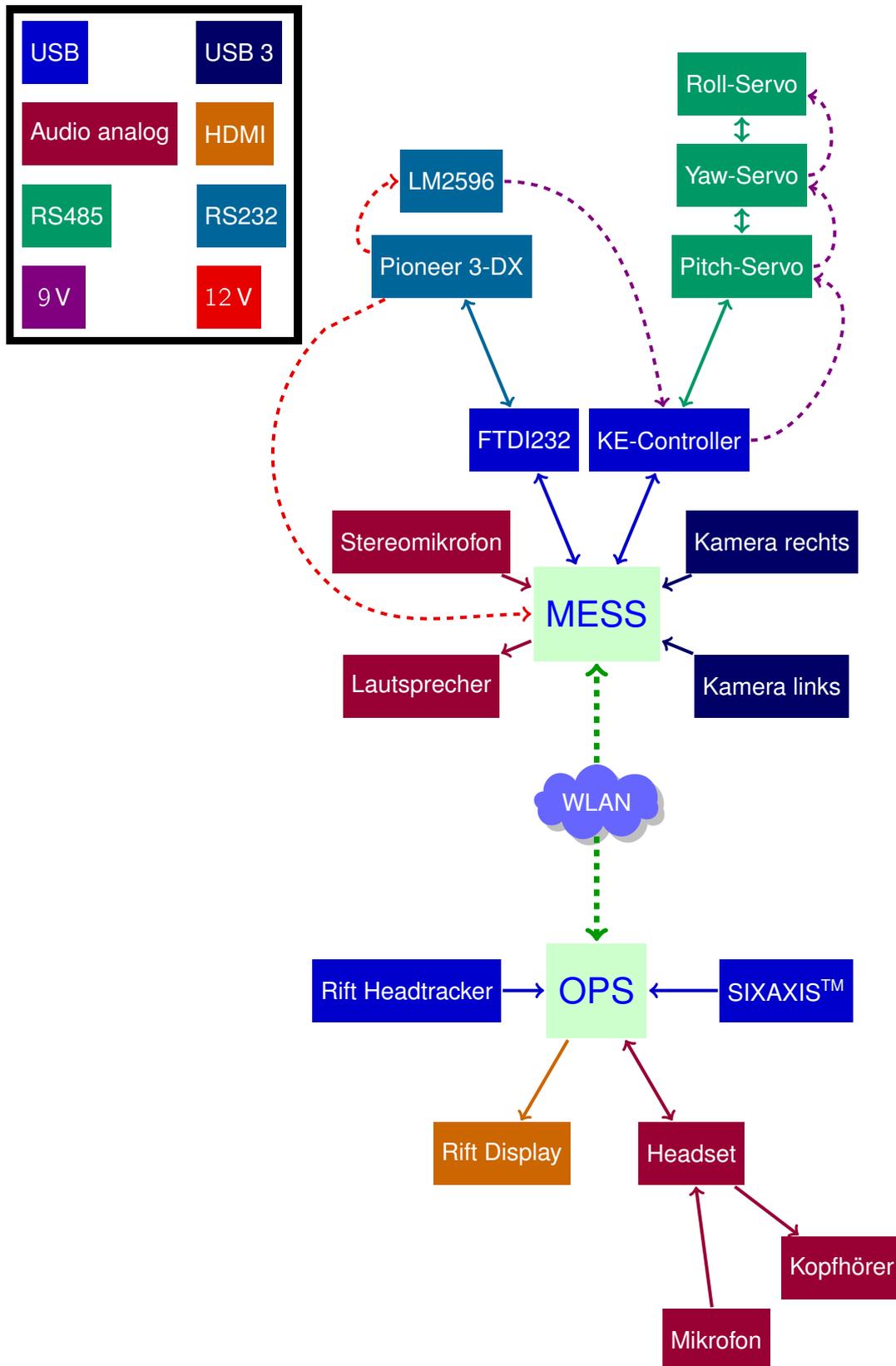


Abbildung 4.10.: Verbindungsschema der **Bedienerstation** und **Mobileinheit**

## 4.4. Steuerungssystem der ME

Das **MESS** ist flexibel austauschbar ausgeführt. Aus den Beständen der Hochschule findet ein Mini-ITX-System auf Basis eines Intel DH57JG Boards Anwendung. Es hat folgende Spezifikationen:

<b>CPU</b>	Quadcore Intel Core i5 3.33GHz
<b>Mainboard</b>	Intel DH57JG; Formfaktor Mini-ITX
<b>RAM</b>	2GB
<b>HDD</b>	32GB SSD

Es besitzt folgende für den Einsatz qualifizierende Ein- und Ausgänge:

<b>Audio-I/O</b>	Soundchip nach AC'97 Standard
<b>Seriell</b>	6x USB 2.0
<b>Netzwerk</b>	1000Base-TX Ethernet
<b>PCIe</b>	1 Slot PCIe 16x

Der PCIe Slot ist erforderlich für unter Umständen eingesetzte Erweiterungskarten mit USB 3.0 oder [IEEE1394A](#)/Firewire Anschlüssen. Diese werden für Industriekameras mit höheren Auflösungen und Bildraten benötigt, bei denen der USB 2.0 Standard für die Datenübertragung nicht ausreicht. Das Intel DH57JG liefert diese Anschlüsse nicht mit.

Für die Aufnahme und Ausgabe von Audiosignalen ist ein Soundadapter erforderlich. Das Board besitzt einen solchen. Die sechs USB-Anschlüsse stellen ausreichend Möglichkeiten zur Verfügung, zusätzliche Hardware anzuschließen. Darunter fallen die **FTDI**-Adapter für den Betrieb des Pioneer 3-DX und des Servo-Controllers.

Die Netzwerkschnittstelle ist für den lokalen Testbetrieb geeignet und findet im Testbetrieb Verwendung. Damit sich die **ME** in ihrer Umgebung ohne Einschränkung durch Leitungen frei bewegen kann, wird im Produktivbetrieb für die Netzwerkschnittstelle auf WLAN gesetzt.

Zusätzliche wird folgende Hardware verbaut:

<b>Audioaufnahme</b>	Stereomikrofon von Sony
<b>Audioausgabe</b>	Stereolautsprecher
<b>WLAN</b>	USB WLAN-Stick

## 4.5. Sicherheitsaspekte

Die ME wird sich in belebten Umgebungen bewegen. Daher ist dem Aspekt der Sicherheit bezüglich Diebstahlschutz (Security) und Verletzungsgefahr (Safety) Rechnung zu tragen. Schutz gegen Verletzungen ist dahingehend gewährleistet, dass das System per Software eine Totmanneinrichtung und einen Not-Aus-Schalter besitzt. Die Motoren der ME sind aktiv, solange der Bediener die Totmantaste auf dem Gamepad gedrückt hält. Der Bediener kann dadurch die Motoren im Zeitraum der Netzwerklatenz kraftlos schalten, indem er die Totmantaste los lässt.

Das MESS überwacht laufend die Verbindung zur OPS und deaktiviert die Motoren innerhalb von 5 s, sobald es einen Verbindungsverlust feststellt. Die maximale Latenz ist einstellbar (siehe Seite 67). Sporadische Latenzen von über 5 s können dabei dazu führen, dass die ME die Motoren kurzzeitig aufgrund fehlerhaft detektierten Verbindungsverlustes deaktiviert. Verzögerungen dieser Dauer werden im lokalen Netzwerk nicht erwartet. Treten diese wider Erwarten auf, ist das Netzwerk für den ordnungsgemäßen Betrieb der ME als ungeeignet einzustufen. Die Latenzen im Internet übersteigen unter Umständen diese Grenze. Der Einsatz des Prototypen über das Internet wird nicht empfohlen, da die Latenzen für die Datenübertragung des Headtracking und der Bilddaten als zu hoch eingestuft werden. An dieser Stelle besitzt das System Optimierungspotential.

Auf der ME ist ein Not-Aus-Taster montiert, der ebenfalls die Totmanneinrichtung bedient und mittels dessen die Motoren ebenso kraftlos geschaltet werden können. Aufgrund der auf der ME ausschließlich genutzten Niederspannungen kann auf ein vollständiges Abschalten der Elektronik verzichtet werden. Im unwahrscheinlichen Falle eines Brandes wird der Einsatz von Löschequipment empfohlen. Ein am Korpus der ME angebrachtes Schild weist darüber hinaus darauf hin, die ME im Betrieb nicht zu berühren.

Bezüglich der Sicherheit gegen Diebstahl sind alle Teile der ME mit dem Korpus verbunden, wodurch die schnelle Entwendung einzelner Teile hinreichend erschwert wird. Gegen den Diebstahl der kompletten ME existiert kein wirksames Konzept.

## 4.6. Komponenten der Bedienerstation

Die OPS basiert auf vorhandenen Komponenten. Sie besteht aus einem PC mit folgender Ausstattung:

<b>CPU</b>	Dual Core Pentium D 3.0 GHz
<b>RAM</b>	4GB
<b>HDD</b>	80GB

Es werden an die OPS folgende Komponenten angeschlossen:

<b>Headset</b>	vorhandenes Headset mit 3,5 mm Stecker nach <a href="#">EIA RS-453</a> Standard
<b>Eingabegerät f. Bewegung</b>	SIXAXIS™-Controller von Sony
<b>HMD</b>	Rift von Oculus VR

### Head-Mounted Display

Das [Virtual Reality \(VR\)](#) System *Rift* ([Abbildung 4.11](#)) der Firma *Oculus VR* ist ein ursprünglich auf Videospiele ausgelegtes [HMD](#). Dieses soll mittels seiner für Headtracking zuständigen Sensorik die [KE](#) der [ME](#) steuern und auf seinem integrierten Display die von der Stereokamera aufgenommenen Bilder anzeigen.



Abbildung 4.11.: Oculus Rift Development Kit  
[[Sebastian Stabinger \(Wikipedia\) \(2014\)](#)]

Die von der Kamera aufgenommenen Bilder besitzen die relevanten Eigenschaften Breite, Höhe sowie Verzerrung aufgrund der Brennweite der Kamera und Linsenverzeichnung. Diese Eigenschaften entsprechen nicht zwingend denen des [HMD](#). Die Optik des [HMD](#) ist fest eingestellt. Ein Ausgleich der optischen Unterschiede ist erforderlich. Andernfalls besteht die Möglichkeit, dass der Bediener Entfernungen in der Umgebung der [ME](#) falsch wahrnimmt. Ein Konferenzzimmer wirkt wie eine Turnhalle, ein kurzer Flur wie ein langer Tunnel. Umgekehrt kann der Bediener das Gefühl bekommen, vor einer Wand zu stehen, obwohl diese real mehrere Meter entfernt ist.

Linsenverzeichnung kann unter Umständen negativen Einfluss auf die Wahrnehmung haben. In der Realität gerade Linien erscheinen verkrümmt. Diesem Umstand muss im Rahmen einer anderen Arbeit bei der Auswahl der geeigneten Optik Rechnung getragen werden. Technische Daten der gewählten Optik können in dem Zusammenhang in der Software Verwendung finden, um Linsenverzeichnung zu beseitigen. Zum Ausgleich dieser optischen Unterschiede kann die Optik der Kamera an das [HMD](#) angepasst werden. Alternativ kann eine Transformation der Bilder mittels Software vorgenommen werden.

Diese Arbeit beschränkt sich an dieser Stelle auf die Nutzung einer bereits vorhandenen Umsetzung, um die genannten optischen Unterschiede softwareseitig zu beheben. Möglichkeiten zur Lösung dieser Probleme werden in [Kapitel 5](#) aufgeführt.

## Eingabegerät für Bewegung

Für die Eingabe von Bewegungsbefehlen wird ein *SIXAXIS<sup>TM</sup>-Controller* ([Abbildung 4.12](#)) der Firma *Sony* verwendet. Dieser Controller kann mittels USB oder Bluetooth betrieben und über bereits vorhandene Software mit dem [ROS](#) gekoppelt werden. Das Gerät bietet mit seinen zwei analogen [Sticks](#) die Möglichkeit, Fahrgeschwindigkeit und Richtung stufenlos vorzugeben. Prinzipiell genügt für die Eingabe von Richtung und Geschwindigkeit ein einzelner Stick, womit der zweite Stick für weitere Entwicklungen frei wäre. Neben den Sticks besitzt das Gerät 17 druckempfindliche Eingabetasten, die ebenfalls für Weiterentwicklung zur Verfügung stehen. Das Gerät wird vom Betriebssystem der [OPS](#) beim Anschließen mit USB unmittelbar als Eingabegerät des Typs *Joystick* erkannt und kann ohne weiteres im [Paket](#) `joystick_driver` genutzt werden. Einsatzgrund für dieses Gerät ist neben seinen Fähigkeiten die Verfügbarkeit an der Hochschule.



Abbildung 4.12.: SIXAXIS™ Controller  
[Evan-Amos (Wikipedia) (2014)]

## 4.7. Software

Für die Steuerung der **ME** und **OPS** ist die Einrichtung einer Software erforderlich. Sie muss in der Lage sein, Eingaben seitens des Bedieners entgegen zu nehmen und ihm Informationen zu liefern. Die Ansteuerung von nicht zum normalen PC-Betrieb gehörender Hardware ist eine weitere Anforderung an die Software. Die Software sollte erweiterbar sein, um unspezifizierte Hardware steuern zu können und neuen Ein- und Ausgabekonzepten eine Basis zu bieten. Zusätzlich zur Erweiterbarkeit mit Blick auf Hardware soll die Software im Hinblick auf rein Software bezogene Fähigkeiten erweiterbar ist. Darunter fällt beispielsweise die Implementierung neuer Wegfindungsalgorithmen und die Erweiterung der Bildverarbeitung (Objekterkennung, **Pick and Place**).

Das System des in dieser Arbeit entwickelten **TPR** verlangt konkret folgende Fähigkeiten von seiner Software:

- Eingaben mittels des SIXAXIS™-Controllers entgegennehmen und diese Eingaben in Steuerbefehle für die Plattform Pioneer 3-DX umwandeln
- Daten des Headtrackers des **HMD** entgegennehmen und diese Daten in Steuerbefehle für die Servos der **KE** umwandeln
- visuelle Aufnahme der Umgebung der **ME** entgegennehmen und diese Bilddaten für die Darstellung auf dem **HMD** verarbeiten
- die umgewandelten Bilddaten auf dem **HMD** anzeigen
- Sensorinformationen der Plattform Pioneer 3-DX verarbeiten
  - diese Sensorinformationen dem Bediener zur Verfügung stellen
  - diese Sensorinformationen für nicht-visuelle Wahrnehmung der Umgebung (*awareness*) nutzen
- Daten über den Betriebszustand der **ME** dem Bediener zur Verfügung stellen
- verbale Kommunikation zwischen Umgebung der **ME** und dem Bediener ermöglichen
- autonome Navigation und **SLAM** ermöglichen

## Gemeinsamkeiten zwischen Mobileinheit und Bedienerstation

Beide Systeme sind bezüglich ihrer Software weitgehend identisch. Unterschiede in der Konfiguration werden gesondert auf Seite 60 aufgeführt. Die Gemeinsamkeiten werden in diesem Teil beschrieben.

### Betriebssystem Ubuntu 12.04 LTS

Ubuntu in der Version 12.04 LTS ist mit allen Standardeinstellungen für den Desktop-Betrieb installiert. Das beinhaltet die grafische Oberfläche für eine bequeme spätere Weiterentwicklung. Der für den Betrieb von MESS und OPS angelegte Benutzer trägt den Namen `tpr`. Er ist Mitglied in der Benutzergruppe `sudo`. Die Mitgliedschaft in der Gruppe stattet ihn indirekt mit administrativen Rechten des Superusers aus. Dies ist erforderlich, damit der Benutzer bei der Weiterentwicklung bei Bedarf fehlende (ROS-)Pakete nachinstallieren und nötigenfalls Systemeinstellungen ändern kann. Des Weiteren ist der Benutzer Mitglied in den Gruppen `dialout`, `tty` und `plugdev`. Er erhält dadurch Zugriff auf die für den Betrieb des HMD, des Fahrwerks und der KE notwendigen Schnittstellen. Die Logindaten für den Benutzer lauten auf beiden Systemen wie folgt:

<b>Login:</b>	<code>tpr</code>
<b>Passwort:</b>	<code>tpr</code>

Während des Bootvorgangs und Starts der grafischen Oberfläche werden einige zusätzliche Prozesse automatisch gestartet, die für den Betrieb als TPS erforderlich sind:

- Jitsi
- TPR-Stack im jeweiligen Betriebsmodus

Der SIP-Client Jitsi startet aus der Einstellung *Startprogramme* der grafischen Oberfläche von Ubuntu.

### Robot Operating System

Das ROS ist ein auf Linux basierendes Framework für die Steuerung autonomer Robotersysteme. Es stellt Pakete und Treiber zur Verfügung, um unterschiedliche Hardware steuern und regeln zu können. Die Steuerung einer Kaffeemaschine ist mit ROS ebenso denkbar wie die eines Hochregallagers oder einer Fertigungsstraße. ROS wird beispielsweise im Projekt *ROS Industrial* (ROS-Industrial Consortium (2014)) für industrielle Aufgaben eingesetzt. Des Weiteren arbeitet an ROS eine Schar freiwilliger Entwickler. Sie

kümmern sich um Neu- und Weiterentwicklung von **Paketen**, leisten Hilfestellungen und beantworten Fragen.

Somit erfüllt **ROS** bezüglich der Ansteuerung unterschiedlicher Hardware die Anforderungen an die Software für den Einsatz auf dem **TPR**. Darüber hinaus ist es aufgrund seiner quelloffenen ausgeführten Struktur erweiterbar. Das **Git-Repository** von Willow Garage enthält außerdem eine umfangreiche Sammlung von **Paketen** und **Stacks** für erhältliche Peripherie. Aus diesen Gründen fiel die Entscheidung zugunsten von **ROS**.

**ROS** basiert auf einer Reihe von Prozessen, **Nodes** genannt. Die Prozesse kommunizieren miteinander über Kanäle namens **Topics**. Die Datenübertragung über diese Kanäle findet mittels **Nachrichten** statt. Grundlage für diese Kommunikationsschnittstelle bildet das **IP**-Protokoll. Damit sich **Nodes** gegenseitig finden, muss auf einem Gerät im Subnetz ein zentraler Kernprozess laufen, genannt `roscore`. Dieser dient als Adressservice. Die Kommunikation zwischen den **Nodes** findet direkt statt. Nachrichten von **Nodes** auf demselben Gerät werden nicht über das externe Netzwerk übertragen. Für die Kommunikation zwischen **Nodes** auf demselben Gerät wird die **Loopback**-Schnittstelle genutzt. Der Austausch von Nachrichten im **ROS** erfolgt über TCP/IP. Die Kommunikation mit dem Kerndienst geschieht über **Extensible Markup Language Remote Procedure Call (XMLRPC)**. Es ist bis auf die im folgenden Abschnitt beschriebene Einstellung der IP-Adresse keine weitere Konfiguration erforderlich, um die Datenverteilung über das Netzwerk zu ermöglichen.

Da **ROS** auf beiden Geräten zum Einsatz kommt, wird der **Workspace** für beide Geräte der einfachen Wartung wegen grundlegend identisch ausgeführt. Der **Workspace** enthält auf beiden Geräten alle **Nodes** für die Betriebsmodi als **OPS** und **MESS**. Eine Auswahl des Betriebsmodus erfolgt anhand der aus dem **Workspace** gestarteten Prozesse. Ein weiterer Grund für die einheitliche Ausführung des **Workspace** ist die Möglichkeit des direkten Betriebs einzelner Komponenten wie der **KE** oder des Fahrwerks an der **OPS**. Dieses Vorgehen erleichtert die spätere Weiterentwicklung der Software.

**Abbildung 4.13** zeigt abstrahiert die Kommunikationskanäle zwischen den **Nodes** untereinander und dem Kerndienst. Das **TCPROS** genannte Protokoll ist eine Transportschicht für **Nachrichten** und **Services**. Es nutzt TCP/IP **Sockets** für die Datenübertragung. [Conley u. a. (2014a)]

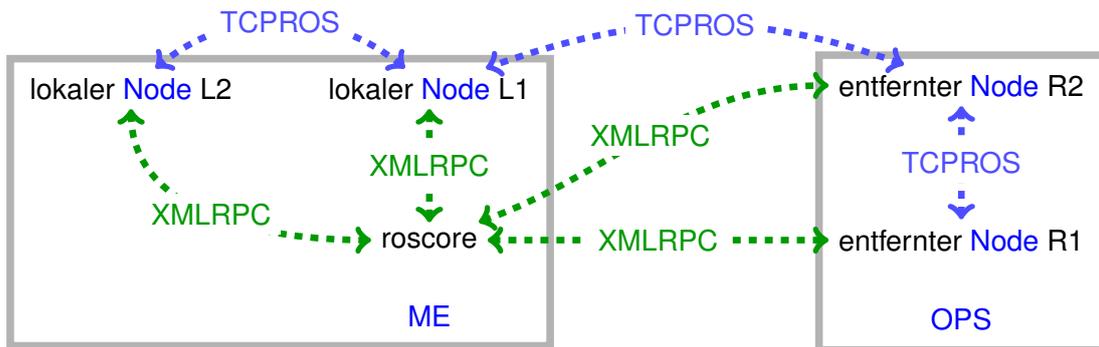


Abbildung 4.13.: Abstrakte Darstellung der Kommunikation im ROS nach [Conley u. a. (2014b)]

Das zur Basisinstallation des ROS verwendete Paket aus dem APT-Repository von Willow Garage trägt den Namen `ros-hydro-desktop-full`. Die Installation des Gesamtpaketes liegt in der Vorbereitung auf spätere Weiterentwicklung begründet.

## Unterschiede zwischen Mobileinheit und Bedienerstation

Zwischen der Software der OPS und des MESS bestehen Unterschiede. Sie ergeben sich aus den zwei Betriebsmodi. Diese Unterschiede werden in diesem Abschnitt beschrieben.

### Konfiguration der Mobileinheit

Auf dem MESS läuft der Kerndienst des ROS, `roscore`. Dieser Prozess wird als **Daemon** durch den Init-Prozess `upstart` gestartet. Eine Startkonfiguration für `roscore` mittels `upstart` wird angelegt. Diese Konfiguration befindet sich im Pfad `/etc/init/roscore.conf`. Die Datei ist in Listing 4.2 aufgeführt.

Listing 4.2: `roscore.conf` Konfiguration für Upstart

```
# roscore - ROS core service

description "ROS core"
author "Hendrik Wiese <hendrik.wiese@haw-hamburg.de">

start on runlevel [2345]
stop on runlevel [016]

respawn
```

```
script
exec su tpr -c "source /opt/ros/hydro/setup.bash && roscore"
end script
```

Damit das **MESS** mit der **OPS** kommunizieren kann, sind einige Einstellungen vorzunehmen. Der **ROS**-Kerndienst `roscore` läuft auf dem **MESS**, um die Einrichtung weiterer **OPS** zu erleichtern. Dem Kerndienst ist mitzuteilen, dass er auf einem externen **Socket** auf sich verbindende **Nodes** der **OPS** lauschen soll. Zu diesem Zweck ist vor dem Start des Dienstes eine Umgebungsvariable zu setzen. Dies erfolgt über folgende Befehle:

```
$ export ROS_MASTER_URI=0.0.0.0:11311
$ sudo restart roscore
```

Durch Ausführung dieser Befehle wird die IP-Adresse und der Port festgelegt, auf denen der Kerndienst lauschen soll. Die Adresse `0.0.0.0` kombiniert alle Netzwerkschnittstellen des Systems. Port `11311` ist als Standard-Port für die Kommunikation der **Nodes** mit `roscore` definiert. Damit diese Information beim Booten erhalten bleibt, wird sie in der Basiskonfiguration des **ROS** auf dem **MESS** hinterlegt. Diese Einstellung wird in der Datei `/opt/ros/hydro/etc/catkin/profile.d/10.roslaunch.sh` ([Listing 4.3](#)) vorgenommen.

Listing 4.3: Modifizierte `10.roslaunch.sh`

```
# roslaunch/env-hooks/10.roslaunch.sh

if [ ! "$ROS_MASTER_URI" ]; then
    export ROS_MASTER_URI=http://0.0.0.0:11311
fi
```

In der ursprünglichen `10.roslaunch.sh` steht anstelle der IP `0.0.0.0` der Eintrag `localhost`. Dadurch lauschte der Kerndienst ausschließlich auf der internen Loopback-Schnittstelle. Die Adresse `0.0.0.0` öffnet den Kerndienst auf allen Netzwerkschnittstellen.

### Konfiguration der **Bedienerstation**

**ROS** ist über den Verbleib des Kerndienstes in Kenntnis zu setzen. Diese Einstellung wird analog zu [Listing 4.3](#) auf der **OPS** festgelegt. Der Befehl zum temporären Konfigurieren der **ROS**-Core Adresse lautet:

```
$ export ROS_MASTER_URI=http://tpr-me.local:11311
```

Permanent eingestellt wird die Adresse auf der **OPS** ebenfalls in der Datei `/opt/ros/hydro/etc/catkin/profile.d/10.roslaunch.sh`.

Das **MESS** registriert sich im lokalen Netzwerk mit seinem Hostname `tpr-me.local`. Dadurch wird es unabhängig von der ihm durch den DHCP zugeteilten IP-Adresse erreichbar.

## Stack für Mobileinheit und Bedienerstation

Der **Stack** wird gemeinsam für beide Einheiten implementiert. Zwei unterschiedliche **Launchfiles** steuern den jeweiligen Betriebsmodus des Stacks.

Das Erstellen des **Workspace** erfolgt entsprechend dem Tutorial [Jain (2014)]. Im Unterschied zum genannten Tutorial heißt der **Workspace** `tpr` anstelle von `catkin_ws`. Er befindet sich im Pfad `~/tpr` im Heimatordner des Benutzers `tpr`. Der vollständige Pfad zum **Workspace** lautet: `/home/tpr/tpr`.

Zusätzlich zum Basispaket sind folgende Pakete installiert bzw. aus dem **Git-Repository** *GitHub* in den **Workspace** geklont:

<b>Servosteuern</b>	<code>ros-hydro-dynamixel-motor</code>
<b>Headtracking</b>	<i>OTL/Oculus</i> auf <i>GitHub</i>

Die zwei Betriebsmodi für **ME** und **OPS** werden repräsentiert durch die folgenden zwei **Launchfiles**:

<b>Betriebsmodus OPS</b>	<code>~/tpr/src/base/launch/ops.launch</code>
<b>Betriebsmodus ME</b>	<code>~/tpr/src/base/launch/me.launch</code>

## Einrichtung des Servotreibers

Der in **ROS** enthaltene **Stack** für den Betrieb der Dynamixel-Servos beinhaltet ein Paket namens `dynamixel_tutorial`. Dieses Paket enthält eine vorkonfigurierte Umgebung für den Betrieb zweier Servomotoren. Die Umgebung wird in das für die Umsetzung des Headtrackings zuständige Paket übertragen und auf den Betrieb der drei Servomotoren der **KE** angepasst.

Die Konfiguration der zu aktivierenden Controller wird in der Datei `~/tpr/src/headunit/config/ke_controllers.yaml` (Listing 4.4) vorgenommen. Diese Konfigurationsdatei wird im **Launchfile** des als *Controller-Spawner* bezeichneten **Nodes** aus

dem Dynamixel-Stack eingebunden. Der Eintrag `pitch_controller` steht hier stellvertretend für die zwei weiteren Controller `yaw_controller` und `roll_controller`. Sie sind, abgesehen von der Motor-ID, identisch konfiguriert.

Listing 4.4: Konfiguration der Controller

```
pitch_controller:
  controller:
    package: dynamixel_controllers
    module: joint_position_controller
    type: JointPositionController
  joint_name: pitch_joint
  joint_speed: 3.0
  motor:
    id: 3
    init: 512
    min: 0
    max: 1023
```

Der [Stack](#) stellt je Controller zwei Nachrichtenkanäle zur Verfügung, auf denen die Controller Kommandos für die Positionierung der Servos empfangen kann. Sie liefern außerdem regelmäßig Informationen über die tatsächliche Lage des jeweiligen Servos sowie dessen Temperatur, das anliegende Moment und den Betriebszustand.

Der Status des Servos wird regelmäßig auf dem Kanal `/<Controllername>/state` als Nachricht des Typs `dynamixel_msgs/JointState` übermittelt.

### Aufnahme Headtracking

Die Aufnahme der Headtracking-Daten übernimmt der [Stack](#) `oculus`. Er enthält den [Node](#) `oculus_driver`. Dieser [Node](#) sendet mit einstellbarer Frequenz die bereits durch die Firmware der Rift vorverarbeiteten und gefilterten Daten des Lagesensors auf den Nachrichtenkanal `/oculus/orientation`. Die Orientierung wird übermittelt als [Quaternion](#) in einer Nachricht des Typs `geometry_msgs/Quaternion`. Der [Node](#) wird im [Launchfile](#) für den Betriebsmodus [OPS](#) konfiguriert und gestartet.

### Umsetzung Headtracking

Eines der für die Steuerung der [ME](#) notwendigen, zu entwickelnden Pakete ist die Schnittstelle zwischen [HMD](#) und [KE](#). Diese Schnittstelle ist als eigenständiger [Node](#)

ausgeführt. Der **Node** ist in Python verfasst und trägt den Namen `headunit.py`. Seitens des **Nodes** bestehen Abhängigkeiten von anderen **Paketen**:

<b>rospy, roscpp</b>	Basispakete
<b>tf</b>	enthält Funktion für die Transformation eines Quaternions in Eulerwinkel
<b>std_msgs</b>	enthält Nachrichtentyp <i>Float64</i> für Übertragung der Eulerwinkel
<b>geometry_msgs</b>	enthält Nachrichtentyp <i>Quaternion</i>

Das Anlegen des Pakets mit Namen `headunit`, das den **Node** beinhalten wird, erfolgt über folgenden Befehl:

```
~/tpr/src$ catkin_create_pkg headunit rospy roscpp tf \
  std_msgs geometry_msgs
```

Dadurch wird die für die Übersetzung durch das **ROS-Build-Werkzeug** `catkin` notwendige Verzeichnisstruktur sowie die erforderlichen Paketdateien angelegt. Ein weiteres Verzeichnis für das **Node-Script** wird erzeugt:

```
~/tpr/src/headunit$ mkdir scripts
```

In diesem Ordner wird das **Node-Script** selbst angelegt und mit den erforderlichen Rechten zur Ausführung versehen:

```
~/tpr/src/headunit/scripts$ touch headunit.py
~/tpr/src/headunit/scripts$ chmod u+x headunit.py
```

Die in [Listing 4.5](#) auszugsweise dargestellte Klasse `Headunit` kapselt die gesamte Funktionalität der Kinematik der **KE**. Der Rift-Treiber liefert die Orientierung des Headtrackers als **Quaternion**. In der Klasse `Headunit` erfolgt unter anderem die Umrechnung des **Quaternions** in die Eulerwinkel  $p(\text{itch})$ ,  $y(\text{aw})$ ,  $r(\text{oll})$  für die drei Servomotoren.

Listing 4.5: Übersetzung des Headtrackings

```
# [...]
2 class Headunit:
    def turn_head(self, p, y, r):
4     if self.sleeping and not self.told_sleeping:
        rospy.loginfo("Headunit: Hey, I'm sleeping, friend! \
            Leave me alone!")
6     self.told_sleeping = True
    elif not self.sleeping:
8
```

```
10     y = numpy.clip(y, self.yaw_min, self.yaw_max)
    # Arbeitsraum pitch wird um den tatsächlichen Yaw \
    Winkel eingeschränkt, um Gimbal Lock Effekte zu \
    verringern.
12     pitch_min_reduced = min(0, self.pitch_min + abs(y))
    pitch_max_reduced = max(0, self.pitch_max - abs(y))

14     p = numpy.clip(p, pitch_min_reduced, \
    pitch_max_reduced)
    r = numpy.clip(r, self.roll_min, self.roll_max)

16
18     self.pitch_pub.publish(p)
    self.yaw_pub.publish(y)
    self.roll_pub.publish(r)

20
22     def rift_orientation_callback(self, quat):
    if not self.sleeping:
        (p,y,r) = euler_from_quaternion([quat.x, quat.y, \
        quat.z, quat.w])
24     self.turn_head(-p,y,-r) # Invertierte Roll- und \
        Pitch-Achse aufgrund gedreht eingebauter Servos
# [...]
```

Bei der Methode `rift_orientation_callback` in [Listing 4.5](#) handelt es sich um die durch den [ROS-Subscriber](#) aufgerufene Methode nach Eingehen einer Nachricht mit der Orientierung der Rift. Die Methode nutzt eine Funktion aus dem [Paket `tf.transformations`](#), um das [Quaternion](#) in Eulerwinkel zu transformieren. Die Methode `turn_head` dreht die Kopfeinheit in die übergebene Orientierung. Dabei werden die drei Winkel auf gegebene Intervalle beschränkt. Der Pitch-Winkel wird darüber hinaus auf ein Intervall beschränkt, das sich aus dem Winkel der Yaw-Achse ergibt. Diese Vorgehensweise reduziert in der Praxis auftretende Effekte in der Nähe des in [Abbildung 3.4](#) genannten Gimbal Locks.

Nach der Transformation und Beschränkung werden die drei Eulerwinkel entsprechend ihrer Zuordnung auf die folgenden drei [Topics](#) gesendet:

- `/pitch_controller/command`
- `/yaw_controller/command`
- `/roll_controller/command`

Der Dynamixel-Stack aus dem Paket `dynamixel-motor` setzt diese Befehle in Bewegung des jeweiligen Servos um.

### Aufnahme von Fahrbefehlen

Für die Aufnahme der Fahrbefehle ist an der OPS das Gamepad angeschlossen. Der Betrieb des Gamepads wird gewährleistet durch das Paket `joystick_drivers`. Dieses Paket beinhaltet einen Node, der die vom Betriebssystem gelieferten Daten des Gamepads in Nachrichten vom Typ `sensor_msgs/Joy` übersetzt und an den Topics `/joy` sendet.

Die Verwendung dieses Pakets setzt den Anschluss des Gamepads über USB voraus. Soll in späteren Arbeiten für die Verbindung die funkbasierte Technologie Bluetooth Verwendung finden, so ist das Paket `ps3joy` zu verwenden [Kolbe (2014)]. Für diese Arbeit wird der einfacheren Einrichtung und höheren Zuverlässigkeit wegen die kabelgebundene Verbindung über USB genutzt.

Der Topic des Nodes `RosAria` benötigt Nachrichten vom Typ `geometry_msgs/Twist`. Die Übersetzung der Nachrichten vom Typ `sensor_msgs/Joy` geschieht in einem sogenannten *Teleoperation-Node*. Dieser Node muss individuell für das verwendete Gamepad entwickelt werden. Der Node für die Verwendung des SIXAXIS™-Controllers ist in Python verfasst. Die Implementierung aus Listing 4.6 nutzt für die Steuerung des Fahrwerks den linken Analogstick des Gamepads (Abbildung 4.14). Bewegungen entlang der Y-Achse des Stickgen führen zu einer Vorwärts- oder Rückwärtsbewegung der ME. Bewegungen entlang der X-Achse führen zu Rotation um die vertikale Körperachse der ME. Die Kombination beider Achsen sorgt für eine Kurvenfahrt. Je nach Auslenkung des Stickgen bewegt sich die ME langsamer oder schneller.

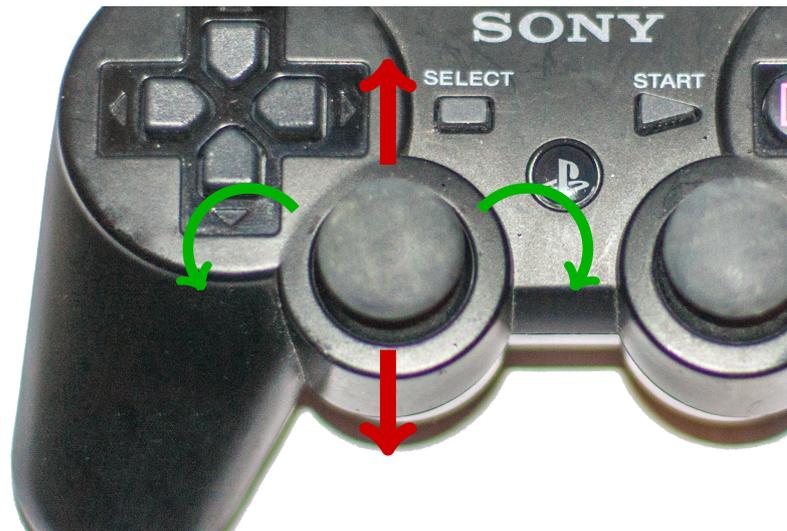


Abbildung 4.14.: Kontrolle des Pioneer 3-DX mittels linkem Analogstick

`self.speed` und `self.tr` werden in der Initialisierung der Klasse durch die Parameter `~/max_velocity` bzw. `~/max_turnrate` gefüllt. Sind die Parameter im `Launchfile` nicht genannt, erhalten sie ihre Standardwerte von  $0,5 \text{ m s}^{-1}$  für `self.speed` und  $1,0 \text{ s}^{-1}$  für `self.tr`.

Listing 4.6: Auszug aus `sixaxis.py` - Übersetzung der Joy-Nachrichten in Twist-Nachrichten

```

1
class ps3Control():
3     # ...
   def ps3input(self, msg):
5         trans = msg.axes[1]
           ang = msg.axes[0]
7
           tw = Twist()
9         tw.linear.x = trans * self.speed
           tw.angular.z = ang * self.tr
11
           self.cmdvel_pub.publish(tw)

```

### Totmanneinrichtung

Von dem `Node` des Pakets `joystick_drivers` wird zusätzlich zu den Achsen der `Sticks` der Zustand aller Knöpfe des Gamepads in der `Nachricht` vom Typ `sensor_msgs/`

Joy übermittelt. Diese Information nutzt die Totmanneinrichtung, um den Zustand der Totmanntaste abzufragen. Die ME wird aktiviert, indem der Bediener auf dem Gamepad die Taste für die Totmanneinrichtung gedrückt hält. Fahrbefehle werden abgewiesen, wenn die ME nicht aktiv ist. Standardmäßig ist die Funktion der Totmanntaste auf die Taste des Gamepads mit dem X gelegt. Sie lässt sich mit dem Parameter `~/dead_mans_button_index` umkonfigurieren. Der Node `dead_mans_button.py` läuft auf dem MESS.

Listing 4.7 zeigt die Klasse `DeadMansButton`. Sie nimmt den Status der Totmanntaste in der Methode `joy_state` entgegen. Diese wird vom ROS-Subscriber in Zeile 10 aufgerufen, sobald eine `sensor_msgs/Joy`-Nachricht eintrifft. Der Status wird von der Klasse auf dem Topic `~/dead_mans_button_okay` als Nachricht des Typs `std_msgs/Bool` mit einer Frequenz von 50Hz veröffentlicht. Mit dieser Frequenz wird außerdem die Zeitdifferenz zur letzten eingetroffenen Nachricht der Totmanntaste aus dem Joy-Node überprüft. Übersteigt diese Differenz 5 s, wird davon ausgegangen, dass die Verbindung des MESS zum Gamepad unterbrochen wurde. Als Resultat erfolgt die Veröffentlichung einer nicht gedrückten Totmanntaste. Die Zeitdifferenz kann mit dem Parameter `~/dead_mans_button_delay` geändert werden. Eine Zeitüberschreitung kann sowohl den Ausfall des Netzwerks, der OPS, der Verbindung zwischen OPS und Gamepad oder des Gamepads selbst als Ursache haben.

Listing 4.7: Auszug aus `dead_mans_button.py`

```
class DeadMansButton():
2   def __init__(self):
    rospy.init_node("dead_mans_button")
4
    self.state = False
6    self.dmb_time = rospy.Time.now()
    self.dmb_state = False
8    self.dmb_idx = \
        rospy.get_param("~/dead_mans_button_index", -5)
    self.dmb_pub = \
        rospy.Publisher("~/dead_mans_button_okay", Bool)
10   rospy.Subscriber("/joy", Joy, self.joy_state)

12   def joy_state(self, msg): # wird durch Joy-Node aufgerufen
    self.dmb_state = msg.buttons[self.dmb_idx] == 1
14   self.dmb_time = rospy.Time.now()

16   def run(self, rate = 50.0):
    r = rospy.Rate(rate)
18   d = rospy.get_param("~/dead_mans_button_delay", 5.0)
```

```
20     while not rospy.is_shutdown():
21         dt = rospy.Time.now() - self.dmb_time
22
23         dmb_state = self.dmb_state if dt.to_sec() < d else \
24             False
25
26         # im Falle des Abreissens der Verbindung nach d \
27             Sekunden stoppen
28         self.dmb_pub.publish(dmb_state)
29
30         r.sleep()
```

Auf dem **MESS** wird dieser Status entgegen genommen und entsprechend reagiert. Abhängig davon, ob die Totmanntaste gedrückt ist oder nicht, wird der Schlafmodus der **ME** deaktiviert oder aktiviert.

### Steuerung der Fahrplattform

Für die Steuerung der Fahrplattform wird das fertige **Paket** `rosaria` genutzt. Dem Tutorial *How to use ROSARIA* [Hedges (2014a)] folgend wird das **Paket** in den Workspace eingebunden.

Der mit dem **Paket** `rosaria` installierte **Node** `RosAria` bindet sich an den **Topic** `~/cmd_vel`. Mittels Nachrichten vom Typ `geometry_msgs/Twist` werden Fahr- und Drehgeschwindigkeit an den **Node** übergeben. Der **Node** überträgt diese an die vom Hersteller des Pioneer 3-DX zur Verfügung gestellte Steuerungsbibliothek `ARIA`. Diese führt die Ansteuerung der Elektronik durch.

Neben dem abonnierten Kanal `~/cmd_vel` stellt der **Node** eine Anzahl eigener Kanäle bereit. Sie übermitteln beispielsweise die folgenden Daten:

<b>Odometrie</b>	<code>~/pose als nav_msgs/Odometry</code>
<b>Ultraschall</b>	<code>~/sonar als sensor_msgs/PointCloud</code>
<b>Batteriespannung</b>	<code>~/battery_voltage als std_msgs/Float32</code>

Darüber hinaus bietet der **Node** zwei **Services**, mit denen die Fahrmotoren ein- und ausgeschaltet werden können (`enable_motors` bzw. `disable_motors`). [Hedges (2014b)]

## Überwachung der ME

Für die Überwachung der ME von der OPS wird ein Plugin für RQT erstellt (Abbildung 4.15). Bei RQT handelt es sich um ein Framework für die Entwicklung von grafischen Oberflächen in ROS unter Zuhilfenahme der Softwarebibliothek Qt. Das in Python geschriebene Plugin beinhaltet einen Bereich für Log-Meldungen des ROS. Der Log-Bereich ist in Abbildung 4.15 nicht dargestellt. Er befindet sich im Pluginfenster rechts neben den in der Abbildung gezeigten Kontrollen. Das Plugin enthält des Weiteren Anzeigen für Lage (1), Temperatur (2), Last (3) und Geschwindigkeit (4) der Servos der KE sowie die Spannung des oder der Akkumulatoren (5). RQT weist zum Zeitpunkt der Ausarbeitung dieser Arbeit Instabilitäten auf, die dazu führen, dass das Plugin gelegentlich abstürzt.

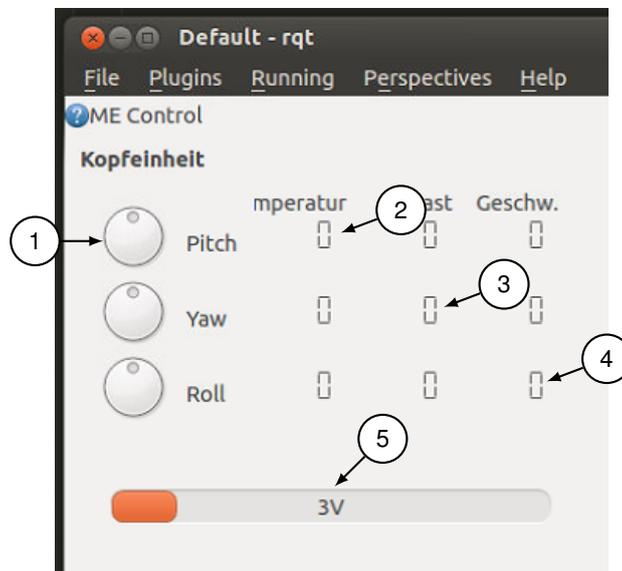


Abbildung 4.15.: Plugin rqt\_ops für die Überwachung der Mobileinheit

Tabelle 4.3.: Kennzeichnungen in Abbildung 4.15

1	Lageanzeigen der Kopfeinheit
2	Temperaturanzeigen der Servos
3	Lastanzeigen der Servos
4	Geschwindigkeitsanzeigen der Servos
5	Batteriespannung

### Starten des Stacks, Betriebsmodus & Parameter

Im Betriebsmodus als **ME** (Listing 4.8) werden durch das **Launchfile** folgende Komponenten gestartet:

<b>Servomotoren</b>	Eingebundene <b>Launchfiles</b> <code>ke_manager.launch</code> und <code>ke_spawner.launch</code>
<b>KE</b>	<b>Node</b> <code>headunit</code>
<b>Webcam links</b>	<b>Node</b> <code>left</code>
<b>Webcam rechts</b>	<b>Node</b> <code>right</code>
<b>Bildverarbeitung</b>	<b>Node</b> <code>iproc_left</code> und <code>iproc_right</code>
<b>Pioneer 3-DX</b>	Eingebundenes <b>Launchfile</b> <code>engine.launch</code>

Der **Node** `headunit` erhält Parameter für die Konfiguration des Arbeitsraumes der **KE**. Die in Listing 4.8 eingetragenen Werte sind die Standardwerte, die ebenfalls in Abwesenheit dieser Parameter angenommen würden. Die Frequenz, mit der der **Node** `right` die Orientierung an den Nachrichtenkanal und damit an die Kopfeinheit schickt, ist vom Standardwert 10 Hz auf 50 Hz erhöht. Dies gewährleistet eine stetige simultane Bewegung. Niedrigere Frequenzen sorgen für zunehmend abgehackte Bewegung.

Die **Nodes** `iproc_left` und `iproc_right` benötigen zur Entfernung von Linsenverzerrung eine Kalibrierung der Kameras. Diese wird für beide Kameras individuell nach [MeloneeWise u. a. (2014)] durchgeführt.

Listing 4.8: Launchfile für Betriebsmodus der **Mobileinheit**

```
<launch>
  <include file="$(find
    headunit)/launch/ke_manager.launch" />
  <include file="$(find
    headunit)/launch/ke_spawner.launch" />
  <node name="headunit" pkg="headunit"
    type="headunit.py" output="screen">
    <param name="pitch_min" value="-75" />
    <param name="pitch_max" value="+90" />
    <param name="yaw_min" value="-50" />
    <param name="yaw_max" value="+50" />
    <param name="roll_min" value="-45" />
    <param name="roll_max" value="+45" />
  </node>
  <node name="left" pkg="usb_cam" type="usb_cam_node"
    ns="/camera">
```

```
        <param name="video_device"
            value="/dev/stereocam/left" />
        <param name="camera_info_url"
            value="package://base/cam_calibration_left.yaml"
            />
    </node>
    <node name="right" pkg="usb_cam"
        type="usb_cam_node" ns="/camera">
        <param name="video_device"
            value="/dev/stereocam/right" />
        <param name="camera_info_url"
            value="package://base/cam_calibration_right.yaml"
            />
    </node>

    <node name="iprocc_left" pkg="image_proc"
        type="image_proc" ns="/camera/left" />
    <node name="iprocc_right" pkg="image_proc"
        type="image_proc" ns="/camera/right" />
    <include file="$(find base)/launch/engine.launch" />
</launch>
```

Das [Launchfile](#) für den Betriebsmodus der [OPS](#) startet folgende Komponenten:

**Oculus Rift Headtracking**    [Node](#) rift  
**Oculus Rift Viewer**        [Node](#) image\_distort\_viewer  
**Visualisierung von ME-Daten** [Node](#) me\_control  
**Joystick/Gamepad**         [Node](#) joy

Bei dem Oculus Rift Viewer handelt es sich um ein Programm, das die für die Rift verzerrten Kamerabilder auf dem Display der Rift anzeigt. Es erhält über Parameter des [Launchfiles](#) Informationen auf welchen [Topics](#) und in welchem Format die Kamerabilder übertragen werden.

Listing 4.9: Launchfile für Betriebsmodus der [Bedienerstation](#)

```
<launch>
  <node pkg="oculus_driver" type="oculus_node"
    name="rift">
    <param name="frequency" value="50" />
  </node>
  <node pkg="oculus_viewer"
    type="image_distort_viewer"
    name="image_distort_viewer" output="screen">
    <param name="image_transport" value="raw"/>
    <param name="left_image"
      value="camera/left/image_rect_color" />
    <param name="right_image"
      value="camera/right/image_rect_color" />
  </node>
  <node pkg="rqt_ops" type="me_control"
    name="me_control" respawn="true" args="-b
    pyside" />
  <node pkg="joy" type="joy_node" name="joy" />
</launch>
```

### Audio- und Videoübertragung

Der Empfang und die Ausgabe von Audiodaten von der OPS übernimmt eine von ROS unabhängige Software mit Namen *Jitsi*. Bei Jitsi handelt es sich um eine quelloffen verfügbare Kommunikationssoftware. Sie unterstützt zahlreiche Protokolle, unter anderem das für Videotelefonie vorgesehene SIP. Jitsi ist ebenfalls dafür zuständig, den vom Stereomikrofon der ME aufgenommenen Schall digitalisiert an die OPS zu übermitteln. Dabei ist zu beachten, dass SIP im Normalfall einen sogenannten *Registrar*, einen Server benötigt, auf dem sich der Benutzer mit seinem SIP-Account registriert und anmeldet. Jitsi ist in der Lage, ein *Registrarless SIP* zu betreiben, bei dem die Gegenstelle im lokalen Netzwerk liegt. Für Registrarless SIP wird kein zusätzlicher Server benötigt. Als Kontakt für den Anruf der Gegenstelle wird der Benutzername und die (IP-)Adresse der Gegenstelle verwendet; beispielsweise `tpr@192.168.1.172` oder `tpr@tpr-me.local`.

Jitsi auf der ME ist so konfiguriert, dass es eingehende Anrufe automatisch annimmt. Der für die Übermittlung der Audiodaten gewählte Codec wird von Jitsi anhand der zur Verfügung stehenden Bandbreite automatisch aus einer Reihe im Programm aktivierter Codecs gewählt. Ebenso verfährt Jitsi bei der Wahl des Video-Codecs.

## 5. Zukunft des Systems

Der [TPR](#) ist so ausgelegt, dass Spielraum für zukünftige Entwicklung vorhanden ist. Viele gute, im Zeitrahmen der Arbeit jedoch nicht realisierbare Ideen werden der Kreativität und dem Forscherdrang nachfolgender Studierender überlassen. In diesem Kapitel finden Anregungen und nicht mehr realisierte Ideen einen Platz.

### 5.1. Allgemeine Erweiterungen

Das Gesamtsystem beschränkt sich zum Zeitpunkt dieser Arbeit auf grundlegende Funktionen. Es werden daher in dieser Sektion Anregungen gegeben, inwiefern in weiteren Abschlussarbeiten das Gesamtsystem erweitert werden könnte.

#### Autonomie

In seinem derzeit realisierten Zustand ist der [TPR](#) nicht autonom. Er reagiert auf einfache Fahrbefehle. Er ist nicht fähig, Hindernissen auszuweichen oder selbständig einen gewählten Zielpunkt anzufahren. Jedoch besitzt er die Kapazität, um das Sensorequipment und die notwendigen Softwarealgorithmen für autonome Navigation zu betreiben.

Der Einsatz eines Laserscanners des Typs *04-LX* von *Hokuyo* empfiehlt sich, um [SLAM](#) zu betreiben. Die Software für den Betrieb dieses Scanners ist als funktionsfähiges [Paket](#) verfügbar. Der Anschluss des Scanners ist über USB problemlos möglich und die notwendigen Berechtigungen im System sind mit der Zugehörigkeit des Benutzers `tpr` zu den Gruppen `tty` und `dialout` gegeben. Offen bleibt die Einrichtung des [Stacks](#). Informationen zur Inbetriebnahme finden sich unter [[Leibs u. a. \(2014\)](#)].

Der Pioneer 3-DX besitzt eine Phalanx aus Ultraschall-Sensoren, die für Hinderniserkennung genutzt werden können. Der zum Betrieb des Fahrwerks genutzte [Stack](#) `rosaria` liefert die gemessenen Distanzen der Sensoren als Punktwolke an das [ROS](#). Die Auswertung dieser Informationen im Navigationsstack bleibt offen. Die Daten können ggfs. zusätzlich zum Hokuyo Laserscanner als Quelle für ein *Obstacle*-Plugin des Navigationsstacks

genutzt werden, um die Hinderniskarte für den Wegfindungsalgorithmus zu füllen. [EitanMardenEpstein u. a. (2014)]

Neben den Ultraschall-Sensoren besitzt der Pioneer 3-DX eine integrierte **Odometrie**. Sie ermöglicht es, eine relative Lokalisation des Roboters innerhalb seines Weltkoordinatensystems vorzunehmen. Zusammen mit **SLAM** ist die **Odometrie** eine hinreichend zuverlässige Vorgehensweise, um der **ME** mitzuteilen, wo sie sich befindet. Autonome Navigation kann beispielsweise dienlich sein, um dem Roboter aktives Folgen einer Person beizubringen. Darüber hinaus kann diese Eigenständigkeit genutzt werden, um die Ladestation anzufahren, wenn der Ladestand der Batterien einen bestimmten Wert unterschreitet.

### **Stereovision, Head-Mounted Display und Kopfeinheit**

Die Stereokamera *Bumblebee* ist für den Betrieb in Kombination mit dem Developer-Kit der *Oculus Rift* nicht geeignet. Die Auflösung der Rift selbst ist für eine dauerhaft angenehme Immersion des Bedieners zu gering. Das zweite Developer Kit der Oculus Rift enthält ein Display mit höherer Auflösung und weitere Verbesserungen in der Darstellung. [Oculus VR (2014)] Es wird daher empfohlen, die Oculus Rift durch ein aktuelles Modell zu ersetzen.

Da sich die im Inventar des Departments vorhandene Bumblebee nicht in Betrieb nehmen ließ und die eingesetzten Webcams in ihren technischen Fähigkeiten beschränkt sind, empfiehlt sich der Einsatz höher auflösender, leichter und schnellerer Industriekameras. Die Optik der eingesetzten Kamera stimmt nicht zwingend mit der Optik des **HMD** überein. Insofern ist es erforderlich, die optischen Unterschiede zu beheben. Für diesen Zweck stehen abstrakt betrachtet zwei Möglichkeiten zur Verfügung:

1. Anpassung der physikalischen Optik der Kamera, indem Objektive verwendet werden, die das Sichtfeld der Kamera auf das entsprechende Sichtfeld des **HMD** bringen. Der Einsatz von Industriekameras mit entsprechenden Objektiven wird empfohlen. Begründet wird dies neben der höheren Auflösung mit der im Gegensatz zu den Webcams problemlosen Austauschbarkeit der Objektive.
2. Anpassung der optischen Unterschiede zwischen Kamera und **HMD** mittels Software.

Eine Kombination beider Ansätze ist möglich und je nach Verfügbarkeit passender Objektive gegebenenfalls zwingend erforderlich.

## 5.2. Erweiterung und Verbesserung der Mobileinheit

Die ME ist auf Erweiterbarkeit ausgelegt. Einige Vorschläge zur Erweiterung und Verbesserung für weitere Arbeiten werden in dieser Sektion aufgeführt.

### Raumklangaufnahme

Für die räumliche akustische Aufnahme der Umgebung der ME steht ein Raumklangmikrofon SPS200 zur Verfügung. Eine passende Halterung ist konstruiert.

Das SPS200 ist ein softwaregesteuertes Klangfeldmikrofon. Es ist in der Lage, Raumklang nahezu ohne Phasenverschiebung aufzunehmen und mittels der entsprechenden Software in Mono, Stereo und Surround-Codecs umzuwandeln. Näheres zu der vom SPS200 genutzten Technologie ist auf der Website des Herstellers zu erfahren. [TSL Professional Products Ltd. (2014)]



Abbildung 5.1.: SPS200 von TSL Professional Products Ltd.  
[TSL Professional Products Ltd. (2014)]

Das SPS200 wiegt ca. 200 g und wird mit einer eigens dafür vorgesehenen Spannungsversorgung zur Erzeugung einer Phantomspannung von 48 V betrieben. Dieses Gerät ist für die Inbetriebnahme des Mikrofons auf der ME zu montieren.

## Display

Als Display kann ein mit dem quelloffenen Betriebssystem Android™ ausgestattetes Smartphone verwendet werden. Die Entwicklung von Applikationen für diese Geräte weist geringere Hürden gegenüber Geräten mit geschlossener Architektur auf und der Anschluss ist über USB problemlos möglich. Kenntnisse in der Programmiersprache Java sind Voraussetzung. Die erforderlichen Werkzeuge sind kostenlos und quelloffen erhältlich.

Zu dieser Arbeit gehört beispielhaft eine konstruierte Halterung für das Smartphone Samsung Galaxy SII GT-I9100. Die Halterung kann zwischen Kameraträger und Halsteil montiert werden.

Als Alternative zur Nutzung eines Smartphones als aktives Display mit eigenem Betriebssystem wäre der Einsatz eines passiven Displays möglich. Ein solches Display könnte aus einem kleinen 3 Zoll Monitor bestehen, der direkt an das MESS angeschlossen wird.

## Greifer

In ihrer derzeitigen Form ist die ME nicht darauf ausgelegt, physisch mit ihrer Umgebung zu interagieren. Diese Interaktion wird erst möglich, wenn der ME Gliedmaßen gegeben werden. Diese können beispielsweise aus einem Roboterarm bestehen, an dessen Ende ein Greifer montiert ist. Die Bedienung des Greifers wäre zum Beispiel durch die Nutzung des zweiten Sticks auf dem Gamepad möglich. Alternativ denkbar wäre der Einsatz einer an der OPS angeschlossenen 3D-Maus beispielsweise des Typs *SpaceNavigator* (Abbildung 5.2) der Firma *3Dconnexion GmbH* aus München. Der Spacenavigator ermöglicht die analoge Eingabe von Vektoren in 6 Achsen, drei davon zur Rotation und drei weitere zur Translation.



Abbildung 5.2.: SpaceNavigator der 3Dconnexion GmbH, München  
[Wikipedia (2014)]

Mittels dieser Informationen ließe sich ein Endeffektor bedienen, der mit Hilfe einer inversen Kinematik den Arm und den Greifer der **ME** bewegt. Die Bewegung des Greifers selbst (greifen und loslassen) wäre über simultane analoge Eingaben über den SIXAXIS™-Controller denkbar, intuitiv beispielsweise über die Bedienung der Taste *L2*, die einer Greifbewegung der Hand des Bedieners am nächsten kommt.

Beim Einsatz eines Roboterarmes ist darauf zu achten, dass der Schwerpunkt der **ME** niedrig genug gehalten wird. Andernfalls wird das Gerät unter Umständen instabil.

### **Erfassung der Umgebung**

Erweiterte Sensorik ermöglicht, dass der Roboter Aktivität außerhalb des Sichtfeldes seiner Kopfkamera erfasst und lokalisiert. Zu dieser Sensorik kann die Erweiterung der Ultraschall-Sensoren nach hinten sowie ein Laserscanner mit 360° Abdeckung gezählt werden. Diese Informationen können genutzt werden, um dem Bediener beispielsweise Personen außerhalb des Sichtbereichs der **ME** zu signalisieren und ihm damit gefahrlose Rückwärtsbewegung zu ermöglichen. Die Darstellung dieser Informationen wäre über eine im **Head-Up-Display (HUD)** des **HMD** angezeigte Umgebungskarte wie in **Abbildung 5.3** denkbar. In eine solche Karte können für den Bediener relevante Informationen wie die eigene Position, Hindernisse oder Personen eingezeichnet werden.

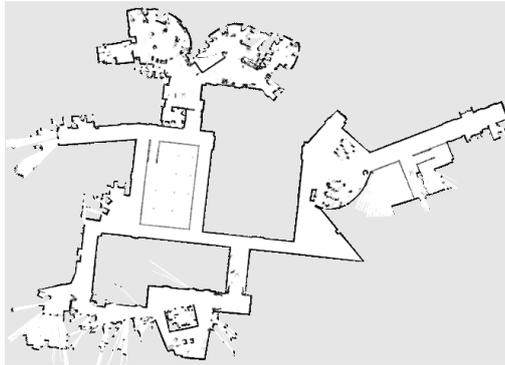


Abbildung 5.3.: Eine Karte der Umgebung  
[Stachniss (2014)]

### Verbesserungen der Kopfeinheit

Es existieren zwischen dem menschlichen Genick und der Mechanik der KE Differenzen bezüglich der genutzten Freiheitsgrade. Weitere Freiheitsgrade und inverse statt direkter Kinematik können diese Differenzen ausgleichen. Für den Umbau des Gelenks zur Vermeidung von Gimbal Locks könnte beispielsweise ein Kugelkopf entwickelt werden, der über Seilzüge oder Stangen aktuiert ist. Alternativ wäre eine Kombination von Hexapoden wie in [Abbildung 5.4](#) gezeigt als Hals vorstellbar. Hexapoden würden dem Gelenksystem weitere Freiheitsgrade hinzufügen.



Abbildung 5.4.: Hexapod  
[Physik Instrumente GmbH & Co. KG (2014)]

### 5.3. Erweiterungen der Bedienerstation

Die OPS ist ebenfalls auf Erweiterbarkeit ausgelegt. Da es sich um einen Standard-PC handelt, sind Software- und Hardwareerweiterungen möglich. Über die übliche Aufrüstbarkeit von Standard-PCs hinaus gehende Vorschläge für Erweiterungen werden nachfolgend kurz vorgestellt.

#### Sitz des Bedieners

Für eine verbesserte Immersion wäre eine Modifikation des Sitzes des Bedieners denkbar. Ein Drehstuhl könnte dergestalt modifiziert werden, dass seine Drehung sensorisch erfasst und als Rotationskommando an die ME geschickt wird. Der umgekehrte Weg wäre ebenso denkbar. Über eine Motorisierung des Drehstuhls könnte die Rotation des Roboters simultan auf den Stuhl übertragen werden, wodurch das System den Bediener physisch die Rotation verspüren lässt.

#### Betrieb als Onlinedienst

Die OPS ließe sich als Onlinedienst umsetzen, der die Bedienung der ME von außen durch autorisiertes Personal ermöglicht. Dieser Dienst könnte von Studierenden genutzt werden, denen aus besonderen Gründen eine persönliche Teilnahme an Veranstaltungen nicht möglich wäre. Auch Professoren oder Lehrbeauftragte könnten den Dienst nutzen, um Veranstaltungen stattfinden zu lassen, die ansonsten aufgrund ihrer Abwesenheit ausgefallen wären.

#### Einsatz in einem Motion-Capturing-System

Neben der Steuerung der ME mittels des SIXAXIS™-Controllers wäre die Steuerung über ein Motion-Capturing-System vorstellbar. Dieses System könnte die Bewegungen des Bedieners aufnehmen und seine Positionsänderung im Raum in Bewegungsbefehle für die ME umsetzen. Dieser Vorgang bewirkt unter Umständen eine Verbesserung der Immersion. Verschiedene Konzepte zur Aufnahme von menschlichen Bewegungsinformationen befinden sich derzeit in der Entwicklung. Einige sind bereits seit einigen Jahren im Einsatz für die Produktion von realistisch wirkenden Computeranimationen. Zur zweiten Kategorie zählen Systeme, bei denen der Akteur einen Anzug trägt, der mit Markern besetzt ist. Diese Marker werden von einem Computersystem ausgewertet und in Bewegungsinformationen für die zu animierenden Charaktere umgewandelt. Diese Informationen wären, sofern in

Echtzeit verarbeitet, ebenfalls nutzbar, um die ME zu steuern. Da der Bediener während der Bedienung zum Erreichen einer hohen Immersion die Rift tragen wird, beschränkt sich seine Sicht auf die ME. Daher muss der Bediener bei Annäherung an Hindernisse gewarnt werden, um Unfälle zu vermeiden.

Um dieses Problem zu lösen und damit einhergehend wesentlich weniger Platz in Anspruch zu nehmen, wäre der Einsatz eines Gerätes von der Art eines *Omni* der Firma *Virtuix* (Abbildung 5.5) denkbar. Bei diesem Gerät wird der Bediener von einem Gurt oder Bügel an Ort und Stelle gehalten. Der Boden dieses Geräts ist auf sehr geringe Reibung ausgelegt, wodurch die Schuhe des Bedieners leicht darauf gleiten. In diesem Gerät vollführte Laufbewegungen werden nicht in Translation, sondern in Bewegungsinformationen umgesetzt. Die Gefahr, gegen ein Hindernis zu laufen, besteht hierbei nicht.



Abbildung 5.5.: Virtuix Omni  
[Virtuix (2014)]

## Surroundsystem und Erweiterungen des Audiosystems

In Verbindung mit dem Raumklangmikrofon empfiehlt sich der Einsatz eines Wiedergabesystems für Raumklang. Dabei kann es sich beispielsweise um eine kommerzielle Surroundanlage handeln. Diese wäre in der Lage, den vom Raumklangmikrofon aufgenommenen Klang als Raumklang wiederzugeben und damit eine weitere Erhöhung der Immersion zu ermöglichen. Es empfiehlt sich unter Umständen ein Test der an der HAW Hamburg verfügbaren Anlage für Wellenfeldsynthese, die Klangquellen lokalisiert simulieren kann.

Neben der Wiedergabe von Raumklang ist die Verbesserung der Schallinformationen durch Filterung näherer Betrachtung wert. Die Servo- und die Antriebsmotoren der ME erzeugen charakteristische Geräusche, die durch entsprechende Filter aus dem Klangbild

der Umgebung der ME herausgefiltert werden können. Die Aufnahme, Analyse und Filterung dieser charakteristischen Geräusche wäre in einer Arbeit vorstellbar. Starke Umgebungsgeräusche, beispielsweise bei dem Besuch der Mensa bei durchschnittlichem Andrang, ließen sich mittels geeigneter Filter ebenfalls verringern. So wäre es für den Bediener sicherlich begrüßenswert, bei einer Konversation mit einem oder mehreren Gesprächspartnern den audiovisuellen Fokus auf die Konversation legen zu können. Möglich wäre dies beispielsweise durch die Einschränkung der Entfernung, in der sich eine Klangquelle befinden darf, um nicht aus dem Audiodatenstrom herausgefiltert zu werden. Dies würde die hinreichend genaue Lokalisation und Entfernungsmessung von Klangquellen erforderlich machen.

## 6. Fazit und Ausblick

Die **TP** ist ein vielfältiges Thema, das großes Interesse bei Unternehmen und Forschern geweckt hat. Das zeigen zahlreiche bereits verfügbare Produkte und viele Forschungsprojekte auf dem Gebiet. **TP** hat nach Einschätzung des Autors das Potential für eine ähnlich starke Entwicklung, wie sie Mobilfunk und das Internet in den vergangenen zwei Jahrzehnten erlebt haben. Gerade die Steigerung der Immersion bis hin zu einer subjektiv nahezu vollständigen Übertragung des Bewusstseins in die **ME** birgt enormes Potential, die Art, wie Menschen miteinander interagieren nachhaltig zu verändern. Die Übertragung von Mimik und Gestik auf die **ME** und Verbesserung der anthropomorphen Attribute der **ME** tragen dazu bei, dass **TPR** vielleicht eines Tages nicht mehr als mobile Alter Egos ihrer Bediener, sondern als die Bediener selbst wahrgenommen werden. Die untersuchten Forschungsprojekte auf dem Gebiet zeigen, dass das Interesse an dieser Entwicklung groß ist.

In dieser Arbeit wurde die Entwicklung und der Aufbau einer Basisplattform für einen erweiterbaren **TPR** zu Forschungszwecken verfolgt. Das System besteht aus **ME** und **OPS** und kann über das lokale Netzwerk ferngesteuert werden. Es ermöglicht bidirektionale Kommunikation und übermittelt stereovisuelle und akustische Sinneseindrücke an den Bediener. Aus vorhandenen Komponenten und mit Hilfe des 3D-Drucks wurde eine Plattform entwickelt, auf deren Basis Forschung im Bereich der **TP** vorgenommen werden kann.

Die entwickelte **KE** des Systems kann mit Einschränkungen dem menschlichen Kopf in seinen Bewegungen folgen. Die Freiheitsgrade lassen gewöhnliche Bewegungen des Kopfes zu. Komplexere und schnellere Bewegungen erfordern Optimierungen bezüglich Freiheitsgraden und Geschwindigkeit. Für den geforderten **PoC** verfolgt die **KE** die Bewegungen des Kopfes schnell genug.

Die *Oculus Rift* wird für die Aufnahme der Bewegungsinformationen verwendet. Der integrierte Headtracker der verfügbaren Version der Rift ist auf die Aufnahme der Orientierung des Kopfes beschränkt. Die von ihm gelieferte Orientierung lässt sich problemlos in drei Winkel umwandeln, die auf die Servomotoren der **KE** gegeben werden. Die Erfassung, Umrechnung und Weitergabe der Orientierungsdaten geschieht dank **ROS** problemlos und hinreichend verzögerungsfrei.

**ROS** vereinfachte die Entwicklung der Schnittstellen zwischen den unterschiedlichen Hardwarekomponenten massiv. Dank verfügbarer Treiber ließ sich die vorhandene Hardware mit Ausnahme der Stereokamera Bumblebee problemlos in Betrieb nehmen. Die Entwicklergemeinschaft um **ROS** stand, wo erforderlich, zu jedem Zeitpunkt mit Rat zur Verfügung.

Auf der **KE** ist ein Kamerasystem bestehend aus zwei Webcams montiert. Es nimmt zwei Bilder auf, welche zur Rift passend verzerrt und auf ihrem Display dargestellt werden. Die Kameras erfassen ein hinreichend weites Sichtfeld, welches in der Software auf die optischen Eigenschaften des **HMD** angepasst werden muss. Die Anpassung und Aufbereitung des Bildes erfolgt in einer anderen Arbeit. Die direkte Ausgabe der Bilder der Kameras auf der Rift funktioniert. Durch noch nicht vorhandene Kompensation der unterschiedlichen Sichtfelder ist die Tiefenwahrnehmung verfälscht. Dieser Effekt ist deutlich wahrnehmbar und sollte im Rahmen einer anderen Arbeit behoben werden.

Nach Auffassung des Autors funktioniert die **KE** und die Stereovision in der Ausführung zum Abschluss dieser Arbeit bereits gut genug, um damit weitere Forschung betreiben zu können. Jedoch hat sich gezeigt, dass die Verzögerung zwischen der Aufnahme der Bilder und der Anzeige selbiger auf dem **HMD** für eine starke Immersion zu hoch ist. Die Testläufe während der Entwicklung zeigten, dass dauerhafte Verwendung vom Bediener ein robustes Gemüt verlangt. Motion Sickness stellt sich sehr schnell ein.

Die auf der **ME** und an der **OPS** installierte Audiohardware ermöglicht bidirektionale Kommunikation in hinreichend guter Tonqualität. Dem freundlichen Gespräch im Aufzug oder der Bestellung einer Tasse Kaffee in der Kantine steht somit zumindest bezüglich der Sprachübermittlung nichts im Wege. Eine Möglichkeit zum sicheren Gütertransport ist nicht vorgesehen.

Mit dem Aufbau der Tragkonstruktion aus käuflich erhältlichen Aluminiumprofilen lässt sich weitere Hardware ohne großen Aufwand an der **ME** montieren. Darunter fällt beispielsweise das Raumklangmikrofon SPS200, dessen Vorbereitung zur Installation in den Anforderungen vorgesehen ist. Diese Anforderung wird dadurch erfüllt, dass der Arbeit die Konstruktion einer Halterung für das SPS200 in Form einer Solidworks-Datei beiliegt.

Das Antriebssystem Pioneer 3-DX gewährleistet die Mobilität der **ME**. Der Anschluss und die Inbetriebnahme des Geräts liefen problemlos vonstatten. Die Fernsteuerung durch den SIXAXIS™-Controller ist nach Meinung des Autors intuitiv und eingängig. Das differentielle Fahrwerk erlaubt Vorwärts- und Rückwärtsbewegung sowie Kurvenfahrt und Drehung auf der Stelle. Der Autor betrachtet diese Bewegungsart als sehr gute Verbindung einfacher Bedienung mit komplexer Mobilität.

Das **MESS** basiert auf einem Intel Core-i5 Prozessor mit 3.33 GHz und ist damit Erfahrungen außerhalb der Arbeit nach zu urteilen schnell genug für grundlegende autonome Navigation.

Da die **ME** offen ausgeführt ist, ist der Einsatz schnellerer x86-kompatibler Hardware möglich. Sie ermöglicht eine leistungsfähigere und schnellere autonome Navigation.

Dank des ausschließlichen Einsatzes von quelloffener Software und offenen Standards an relevanten Stellen ist die Forderung nach Erweiterbarkeit der Software im Rahmen der Möglichkeiten des Gesamtsystems erfüllt. Neue Hardware lässt sich softwareseitig einfach in Betrieb nehmen, sofern dabei auf technische Kompatibilität zum bestehenden System geachtet wird. Es existiert ein großes Kontingent an fertiger Treibersoftware für **ROS**, wodurch ein breites Spektrum an unterschiedlicher Hardware abgedeckt wird. Daher empfiehlt sich bei Auswahl neuer Hardware, im Vorwege Recherchen nach bereits vorhandenen Treibern durchzuführen. Andererseits lässt sich **ROS** aus Sicht des Autors leicht um neue Treiber erweitern.

Zu guter Letzt wurden Aspekte der Sicherheit für Passanten und das System selbst betrachtet und umgesetzt. Eine Totmanneinrichtung sorgt für Entkoppelung der Aktoren, wenn der Bediener die **OPS** verlässt oder wenn es zu einem Verbindungsabbruch kommt. Das System arbeitet zuverlässig, wenn die Voraussetzungen durch das lokale Netzwerk gegeben sind. Des Weiteren ist auf der **ME** ein Not-Aus-Schalter montiert, der bei Betätigen die Aktoren ebenfalls kraftlos schaltet. Ein gut sichtbar angebrachter Hinweis an der **ME** fordert dazu auf, die **ME** nicht zu berühren. Nach Ansicht des Autors sind Sicherheitsanforderungen bezüglich Unfallverhütung damit erfüllt. Dem Diebstahl einzelner Komponenten wird dadurch vorgebeugt, dass alle Komponenten fest mit der Tragkonstruktion verbunden sind. Ein schnelles, unauffälliges Entwenden einzelner Komponenten ist dadurch hinreichend erschwert. Der Diebstahl der gesamten **ME** kann aufgrund ihrer Mobilität nicht auf technischem Wege verhindert werden. Der Autor empfiehlt daher, sichtbar darauf hinzuweisen, dass dieses Gerät Eigentum der Hochschule ist. Personen im Besitz der **ME** sind unter Umständen zu überprüfen.

Der HAW Hamburg steht mit dem **TPR** ein solides System zur Verfügung, das ein breites Feld möglicher Forschung und Entwicklung eröffnet. Das System bietet weitreichende Möglichkeiten, es zu erweitern. Zahlreiche Vorschläge wurden gemacht, auf deren Basis weitere Arbeiten entstehen können.

# Literaturverzeichnis

- [Örn Aðalgeirsson 2009] AÐALGEIRSSON, Sigurður Örn: *MeBot: A robotic platform for socially embodied telepresence*, Massachusetts Institute of Technology, Master Thesis, 2009
- [Adept MobileRobots 2014a] ADEPT MOBILEROBOTS: *ARIA - MobileRobots Research and Academic Customer Support*. 2014. – URL <http://robots.mobilerobots.com/wiki/ARIA>. – Zugriffsdatum: 10.04.2014
- [Adept MobileRobots 2014b] ADEPT MOBILEROBOTS: *Pioneer P3-DX*. 2014. – URL <http://www.mobilerobots.com/ResearchRobots/PioneerP3DX.aspx>. – Zugriffsdatum: 01.04.2014
- [Axel1963 (Wikipedia) 2014] AXEL1963 (WIKIPEDIA): *Rolling-Shutter-Effekt*. 2014. – URL <http://de.wikipedia.org/w/index.php?title=Rolling-Shutter-Effekt&oldid=121796290>. – Zugriffsdatum: 20.03.2014. – Lizenz: Creative Commons Attribution-Share Alike 3.0 Unported
- [Business Wire 2013] BUSINESS WIRE: *InTouch Health and iRobot Announce First Customers to Install RP-VITATM, The New Face of Telemedicine | Business Wire*. 2013. – URL <http://www.businesswire.com/news/home/20130506005495/en/InTouch-Health-iRobot-Announce-Customers-Install-RP-VITATM>. – Zugriffsdatum: 29.03.2014. – Lizenz: alle Rechte am Bild liegen bei InTouch Health
- [Conley u. a. 2014a] CONLEY, Ken u. a.: *ROS/TCPROS*. 2014. – URL <http://wiki.ros.org/ROS/TCPROS>. – Zugriffsdatum: 07.04.2014
- [Conley u. a. 2014b] CONLEY, Ken u. a.: *ROS/Technical Overview*. 2014. – URL <http://wiki.ros.org/ROS/Technical%20Overview>. – Zugriffsdatum: 10.04.2014
- [Cosgun u. a. 2013] COSGUN, Akansel ; FLORENCIO, Dinei A. ; CHRISTENSEN, Henrik I.: *Autonomous Person Following for Telepresence Robots*. In: *International Conference on Robotics and Automation (ICRA)*. Karlsruhe, Germany : IEEE, 2013, S. 4335 – 4342
- [EitanMardenEpstein u. a. 2014] EITANMARDENEPSTEIN u. a.: *navigation/Tutorials/RobotSetup - ROS Wiki*. 2014. – URL <http://wiki.ros.org/navigation/Tutorials/RobotSetup>. – Zugriffsdatum: 07.04.2014

- [Evan-Amos (Wikipedia) 2014] EVAN-AMOS (WIKIPEDIA): *SIXAXIS*. 2014. – URL <https://en.wikipedia.org/w/index.php?title=Sixaxis&oldid=601394103>. – Zugriffsdatum: 10.03.2014. – Lizenz: public domain
- [GU u. a. 2013] GU, William ; SEET, Gerald ; MAGNENAT-THALMANN, Nadia: Improved Human Interaction in Telepresence Robot using Real-Time Face Segmentation. In: *Proceedings of the 32nd Chinese Control Conference*. Xi'an, China, 2013, S. 5902 – 5907
- [Hedges 2014a] HEDGES, Reed: *How to use ROSARIA*. 2014. – URL <http://wiki.ros.org/ROSARIA/Tutorials/How%20to%20use%20ROSARIA>. – Zugriffsdatum: 03.03.2014. – Lizenz: Creative Commons BY 3.0 Deutschland
- [Hedges 2014b] HEDGES, Reed: *ROSARIA*. 2014. – URL <http://wiki.ros.org/ROSARIA>. – Zugriffsdatum: 07.04.2014
- [InTouch Health 2014a] INTOUCH HEALTH: *RP-7i Robot | InTouch Health*. 2014. – URL <http://www.intouchhealth.com/products-and-services/products/rp-7i-robot/>. – Zugriffsdatum: 18.01.2014
- [InTouch Health 2014b] INTOUCH HEALTH: *RP-VITA Remote Presence Robot*. 2014. – URL <http://www.intouchhealth.com/products-and-services/products/rp-vita-robot/>. – Zugriffsdatum: 18.01.2014
- [iRobot 2014a] IROBOT: *iRobot Ava 500 Video Collaboration Robot, Telepresence Robot*. 2014. – URL <http://www.irobot.com/us/learn/commercial/ava500>. – Zugriffsdatum: 20.01.2014
- [iRobot 2014b] IROBOT: *iRobot Ava Mobile Robotics Platform*. 2014. – URL <http://www.irobot.com/us/learn/commercial/ava.aspx>. – Zugriffsdatum: 20.01.2014
- [iRobot 2014c] IROBOT: *iRobot Ava Platform - Features*. 2014. – URL <http://www.irobot.com/us/learn/commercial/ava/Features.aspx>. – Zugriffsdatum: 20.01.2014
- [Jain 2014] JAIN, Advait: *Creating a workspace for catkin*. 2014. – URL [http://wiki.ros.org/catkin/Tutorials/create\\_a\\_workspace](http://wiki.ros.org/catkin/Tutorials/create_a_workspace). – Zugriffsdatum: 03.03.2014. – Lizenz: Creative Commons BY 3.0 Deutschland
- [Kolbe 2014] KOLBE, Felix: *ps3joy - ROS Wiki*. 2014. – URL <http://wiki.ros.org/ps3joy>. – Zugriffsdatum: 01.04.2014
- [Leibs u. a. 2014] LEIBS, Jeremy u. a.: *hokuyo\_node - ROS Wiki*. 2014. – URL [http://wiki.ros.org/hokuyo\\_node](http://wiki.ros.org/hokuyo_node). – Zugriffsdatum: 13.04.2014

- [Logitech AG 2014] LOGITECH AG: *Logitech DE - Webcam HD C310 720p*. 2014. – URL <http://www.logitech.com/de-de/product/hd-webcam-c310?crid=34>. – Zugriffsdatum: 03.04.2014
- [MeloneeWise u. a. 2014] MELONEEWISE u. a.: *camera\_calibration/Tutorials/MonocularCalibration - ROS Wiki*. 2014. – URL [http://wiki.ros.org/camera\\_calibration/Tutorials/MonocularCalibration](http://wiki.ros.org/camera_calibration/Tutorials/MonocularCalibration). – Zugriffsdatum: 07.04.2014
- [MIT Media Lab 2014] MIT MEDIA LAB: *MeBot Overview*. 2014. – URL <http://robotic.media.mit.edu/projects/robots/mebot/overview/overview.html>. – Zugriffsdatum: 17.01.2014
- [NASA Glenn Research Center 2014] NASA GLENN RESEARCH CENTER: *Aircraft Rotations*. 2014. – URL <http://www.grc.nasa.gov/WWW/K-12/airplane/rotations.html>. – Zugriffsdatum: 05.02.2014
- [Oculus VR 2014] OCULUS VR: *The All New Oculus Rift Development Kit 2 (DK2) Virtual Reality Headset | Oculus Rift - Virtual Reality Headset for 3D Gaming*. 2014. – URL <http://www.oculusvr.com/dk2/>. – Zugriffsdatum: 07.04.2014
- [Physik Instrumente GmbH & Co. KG 2014] PHYSIK INSTRUMENTE GMBH & Co. KG: *PI Datenblatt | Spezifikationen: H-840 6-Achsen-Hexapod Kostengünstig, präzise und schnell; PI: Ultra Precision Motion Control*. 2014. – URL <http://www.physikinstrumente.de/de/produkte/prdetail.php?sortnr=700811>. – Zugriffsdatum: 24.03.2014
- [Point Grey Research Inc 2014] POINT GREY RESEARCH INC: *Point Grey Bumblebee 2*. 2014. – URL <http://ww2.ptgrey.com/stereo-vision/bumblebee-2>. – Zugriffsdatum: 05.02.2014
- [Richmilliron (Wikipedia) 2014] RICHMILLIRON (WIKIPEDIA): *Rolling-Shutter-Effekt*. 2014. – URL <http://de.wikipedia.org/w/index.php?title=Rolling-Shutter-Effekt&oldid=121796290>. – Zugriffsdatum: 20.03.2014. – Lizenz: Creative Commons Attribution-Share Alike 3.0 Unported
- [Robotics Business Review 2014] ROBOTICS BUSINESS REVIEW: *iRobot Ava 500 - Robotics Business Review*. 2014. – URL [http://www.roboticsbusinessreview.com/article/irobot\\_ava\\_500/](http://www.roboticsbusinessreview.com/article/irobot_ava_500/). – Zugriffsdatum: 20.01.2014
- [Robotis 2014] ROBOTIS: *:::ROBOTIS:::*. 2014. – URL [http://www.robotis.com/xe/dynamixel\\_en](http://www.robotis.com/xe/dynamixel_en). – Zugriffsdatum: 07.04.2014
- [ROS-Industrial Consortium 2014] ROS-INDUSTRIAL CONSORTIUM: *ROS Industrial*. 2014. – URL <http://rosindustrial.org/>. – Zugriffsdatum: 15.03.2014

- [Sebastian Stabinger (Wikipedia) 2014] SEBASTIAN STABINGER (WIKIPEDIA): *Oculus Rift - Developer Version - Front*. 2014. – URL [https://en.wikipedia.org/wiki/File:Oculus\\_Rift\\_-\\_Developer\\_Version\\_-\\_Front.jpg](https://en.wikipedia.org/wiki/File:Oculus_Rift_-_Developer_Version_-_Front.jpg). – Zugriffsdatum: 10.03.2014. – Lizenz: Creative Commons Attribution 3.0 Unported
- [Smashing Robotics 2014] SMASHING ROBOTICS: *Telepresence Robots Reviewed - Part 2 - Smashing Robotics*. 2014. – URL <http://www.smashingrobotics.com/telepresence-robots-reviewed-part-2/>. – Zugriffsdatum: 01.02.2014
- [Stachniss 2014] STACHNISS, Prof. Dr. C.: *Cyrrill Stachniss - Robotics Datasets*. 2014. – URL <http://www.informatik.uni-freiburg.de/~stachnis/datasets.html>. – Zugriffsdatum: 11.04.2014
- [Steptoe u. a. 2012] STEPTOE, William u. a.: *Beaming: An Asymmetric Telepresence System* / IEEE Computer Society. 2012. – Forschungsbericht
- [Suitable Technologies, Inc. 2014a] SUITABLE TECHNOLOGIES, INC.: *Suitable Technologies Beam*. 2014. – URL <https://www.suitabletech.com/beam/>. – Zugriffsdatum: 17.01.2014
- [Suitable Technologies, Inc. 2014b] SUITABLE TECHNOLOGIES, INC.: *Suitable Technologies Beam Remote Presence System*. 2014. – URL <https://www.suitabletech.com/beam/>. – Zugriffsdatum: 12.04.2014
- [Teaching\*TheHumanBrain 2014] TEACHING\*THEHUMANBRAIN: *Beweglichkeit der HWS - Zusammenwirken der Kopfgelenke*. 2014. – URL [http://teaching.thehumanbrain.info/theorie.php?theo=22#beweglichkeit\\_HWS](http://teaching.thehumanbrain.info/theorie.php?theo=22#beweglichkeit_HWS). – Zugriffsdatum: 03.03.2014. – Lizenz: Creative Commons BY-NC-ND 3.0 Deutschland
- [Toshima und Aoki 2006] TOSHIMA, Iwaki ; AOKI, Shigeaki: The effect of head movement on sound localization in an acoustical telepresence robot: TeleHead. In: *Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*. Beijing, China : IEEE, 2006, S. 872 – 877
- [Toshima und Aoki 2007] TOSHIMA, Iwaki ; AOKI, Shigeaki: Quantitative evaluation of delay time of head movement for an acoustical telepresence robot: TeleHead. In: *International Conference on Robotics and Automation*. Roma, Italy : IEEE, 2007, S. 4301 – 4306
- [Toshima und Aoki 2009] TOSHIMA, Iwaki ; AOKI, Shigeaki: The effect of head movement on sound localization in an acoustical telepresence robot: TeleHead. In: *The 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*. St. Louis, USA : IEEE, 2009, S. 193 – 198

- [TSL Professional Products Ltd. 2014] TSL PROFESSIONAL PRODUCTS LTD.: *TSL Products | SPS200 Software Controlled Microphone*. 2014. – URL <http://www.tslproducts.com/soundfield/sps200-software-controlled-microphone/>. – Zugriffsdatum: 05.02.2014
- [Virtuix 2014] VIRTUIX: *Virtuix Omni*. 2014. – URL [http://www.virtuix.com/wp-content/uploads/2013/06/color-locomotive\\_with-zdepth-e1370323677994.jpg](http://www.virtuix.com/wp-content/uploads/2013/06/color-locomotive_with-zdepth-e1370323677994.jpg). – Zugriffsdatum: 12.04.2014
- [Wiesemann und Theis GmbH. 2014] WIESEMANN UND THEIS GMBH.: *WuT Hintergrundinformationen RS485-Bussysteme*. 2014. – URL <http://www.wut.de/e-6www-11-apde-000.php>. – Zugriffsdatum: 03.04.2014
- [Wikipedia 2014] WIKIPEDIA: *3Dconnexion input-device with 6 degrees of freedom*. 2014. – URL <https://en.wikipedia.org/wiki/File:Space-Navigator.jpg>. – Zugriffsdatum: 10.03.2014. – Lizenz: Multi-license with GFDL and Creative Commons CC-BY-SA-2.5 and older versions (2.0 and 1.0)
- [Wikipedia 2014] WIKIPEDIA: *Git*. 2014. – URL <http://de.wikipedia.org/w/index.php?title=Git&oldid=129392643>. – Zugriffsdatum: 12.04.2014. – Lizenz: Creative Commons Shared Alike 3.0
- [Wikipedia 2014a] WIKIPEDIA: *Immersion (virtuelle Realität)*. 2014. – URL [http://de.wikipedia.org/w/index.php?title=Immersion\\_\(virtuelle\\_Realit%C3%A4t\)&oldid=126270572](http://de.wikipedia.org/w/index.php?title=Immersion_(virtuelle_Realit%C3%A4t)&oldid=126270572). – Zugriffsdatum: 02.04.2014
- [Wikipedia 2014b] WIKIPEDIA: *Institute of Electrical and Electronics Engineers*. 2014. – URL [http://de.wikipedia.org/w/index.php?title=Institute\\_of\\_Electrical\\_and\\_Electronics\\_Engineers&oldid=128755298](http://de.wikipedia.org/w/index.php?title=Institute_of_Electrical_and_Electronics_Engineers&oldid=128755298). – Zugriffsdatum: 31.03.2014
- [Wikipedia 2014c] WIKIPEDIA: *Odometrie*. 2014. – URL <http://de.wikipedia.org/w/index.php?title=Odometrie&oldid=126958792>. – Zugriffsdatum: 13.04.2014
- [Wikipedia 2014d] WIKIPEDIA: *Roll-Nick-Gier-Winkel*. 2014. – URL <http://de.wikipedia.org/w/index.php?title=Roll-Nick-Gier-Winkel&oldid=129318871>. – Zugriffsdatum: 12.04.2014. – Lizenz: Creative Commons Shared Alike 3.0
- [Willow Garage 2014a] WILLOW GARAGE: *Robots using ROS*. 2014. – URL <http://wiki.ros.org/Robots>. – Zugriffsdatum: 01.04.2014. – Lizenz: Creative Commons Attribution 3.0

- 
- [Willow Garage 2014b] WILLOW GARAGE: *ROS/Installation - ROS Wiki*. 2014. – URL <http://wiki.ros.org/ROS/Installation>. – Zugriffsdatum: 01.04.2014. – Lizenz: Creative Commons Attribution 3.0
- [Willow Garage 2014c] WILLOW GARAGE: *Texai Hardware Specifications*. 2014. – URL <http://www.willowgarage.com/pages/texai/hardware>. – Zugriffsdatum: 17.01.2014
- [Willow Garage 2014d] WILLOW GARAGE: *Texai Remote Presence System*. 2014. – URL <http://www.willowgarage.com/pages/texai/overview>. – Zugriffsdatum: 06.04.2014

# Glossar

<b>APT-Repository</b>	Mittels APT verwaltet Ubuntu die im System installierten Pakete. APT-Pakete enthalten Programme, Treiber oder Bibliotheken. Ein APT-Repository ist eine Quelle für solche Pakete. Willow Garage unterhält ein eigenes APT-Repository mit den zu <a href="#">ROS</a> gehörenden APT-Paketen. <a href="#">60</a>
<b>ARIA</b>	ARIA ist eine für die Steuerung des Pioneer 3-DX zuständige Funktionsbibliothek. Sie wird vom Hersteller des Pioneer 3-DX entwickelt und ausgeliefert. [ <a href="#">Adept MobileRobots (2014a)</a> ] <a href="#">96</a>
<b>Daemon</b>	Eigenständig arbeitender Hintergrundprozess in Linux. <a href="#">60</a>
<b>FFF</b>	Fused Filament Fabrication ist eine Methode des 3D Drucks. Dabei wird geschmolzenes Filament (zumeist Kunststoff) über einen Extruder schichtweise auf ein Druckbett aufgetragen. Durch den schichtweisen Aufbau entsteht so nach und nach das fertige Objekt. <a href="#">47</a>
<b>FTDI</b>	Future Technology Device International Ltd. - Hersteller von Chips zur Übersetzung zwischen USB und zahlreichen seriellen Übertragungstechnologien und Bussystemen. Für den Einsatz in dieser Arbeit sind zwei Module von FTDI in Verwendung: USB ↔ <a href="#">RS-485</a> für die Steuerung der Servomotoren der <a href="#">KE</a> und USB ↔ <a href="#">RS-232</a> für die Steuerung des Pioneer 3-DX Fahrwerks. <a href="#">52</a>
<b>Gamepad</b>	Eingabegerät aus dem Videospielektor, ausgelegt auf beidhändige Eingabe von Bewegungs- und Aktionsbefehlen. <a href="#">26</a> , <a href="#">27</a> , <a href="#">53</a> , <a href="#">66</a>

---

<b>Git-Repository</b>	Git ist eine freie Software zur verteilten Versionsverwaltung von Dateien, die ursprünglich für die Quellcode-Verwaltung des Linux-Kernels entwickelt wurde. [Wikipedia (2014)] Ein Repository ist eine Git-Datenbank, die Versionsunterschiede speichert und darüber unter anderem die Wiederherstellung älterer Versionsstände ermöglicht. 59, 62
<b>Headset</b>	Kombination aus Kopfhörer und Mikrofon. 27
<b>Headtracking</b>	Bewegung, Lage und Orientierung des Kopfes des Bedieners werden sensorisch erfasst und in digitalisierter Form einer Software verfügbar gemacht. 23, 26
<b>holonomisches Fahrwerk</b>	Fahrwerk mit drei Freiheitsgraden. Kann sich in der Ebene in alle Richtungen bewegen und sich simultan um die vertikale Körperachse drehen. 17
<b>IEEE</b>	Das Institute of Electrical and Electronics Engineers (IEEE) ist ein weltweiter Berufsverband von Ingenieuren hauptsächlich aus den Bereichen Elektrotechnik und Informationstechnik mit juristischem Sitz in New York City und Betriebszentrale in Piscataway, New Jersey. [Wikipedia (2014b)] 11, 50, 52
<b>Immersion</b>	Immersion beschreibt die Überführung in einen Bewusstseinszustand (Eindruck), bei dem sich die Wahrnehmung der eigenen Person in der realen Welt vermindert und die Identifikation mit dem „Ich“ (dem Avatar) in der virtuellen Welt vergrößert. [Wikipedia (2014a)] Wird diese Beschreibung angewendet auf einen TPR, entspricht die reale Welt der Umgebung am Ort des Bedieners und die virtuelle Welt der Umgebung des TPR. 11, 12, 22
<b>IP</b>	Internet Protokoll; Netzwerkprotokoll der Schicht 3 des OSI-Modells. 24, 59, 95, 96
<b>Kernel</b>	Betriebssystemkern. Basisprozess eines jeden Betriebssystems. Beinhaltet unter anderem Algorithmen zur Arbeitsteilung von Prozessen, Speichermanagement, Hardwaretreiber. 44

---

<b>Launchfile</b>	In XML verfasste Startkonfiguration für einen <a href="#">Stack</a> oder ein <a href="#">Paket</a> . Wird mit dem Befehl <code>roslaunch</code> gestartet. <a href="#">62</a> , <a href="#">63</a> , <a href="#">67</a> , <a href="#">71</a> , <a href="#">73</a> , <a href="#">102</a>
<b>Loopback</b>	Lokale Netzwerkschnittstelle mit <a href="#">IP</a> -Adressen nach dem Muster 127.x.x.x. Nutzt keine Netzwerkhardware. <a href="#">59</a>
<b>Motion Sickness</b>	Übelkeit, hervorgerufen durch Diskrepanzen in der Wahrnehmung von Bewegung zwischen dem Gleichgewichtssinn und den externen Sinnesorganen. <a href="#">46</a>
<b>Nachricht</b>	Zur Kommunikation zwischen <a href="#">Nodes</a> verwendetes Datenpaket. <a href="#">59</a> , <a href="#">66–68</a> , <a href="#">95–97</a>
<b>Node</b>	Teil eines <a href="#">Pakets</a> . Prozess, der <a href="#">Nachrichten</a> verarbeitet und erzeugt oder <a href="#">Services</a> zur Verfügung stellt oder konsumiert. <a href="#">59–64</a> , <a href="#">66–69</a> , <a href="#">71</a> , <a href="#">73</a> , <a href="#">95–97</a> , <a href="#">102</a>
<b>Odometrie</b>	Odometrie oder auch Hodometrie bezeichnet eine Methode der Schätzung von Position und Orientierung eines mobilen Systems anhand der Daten seines Vortriebsystems. [ <a href="#">Wikipedia (2014c)</a> ] <a href="#">76</a>
<b>Paket</b>	Teil eines <a href="#">Stacks</a> . Wird für Modularisierung des <a href="#">Stacks</a> nach Verwendungszweck eingesetzt. <a href="#">55</a> , <a href="#">59</a> , <a href="#">64–67</a> , <a href="#">69</a> , <a href="#">75</a> , <a href="#">95</a> , <a href="#">97</a>
<b>Parameter</b>	<a href="#">Nodes</a> können mittels Parametern konfiguriert werden. Der Kernprozess <code>roscore</code> speichert diese Parameter und stellt sie allen <a href="#">Nodes</a> zur Laufzeit zur Verfügung. Parameter können numerische und alphanumerische Daten enthalten. <a href="#">67</a> , <a href="#">68</a>
<b>Pick and Place</b>	Autonomes Erkennen, Aufheben und Ablegen von Objekten <a href="#">57</a>
<b>Quaternion</b>	Komplexe Zahl mit dreidimensionalem Imaginärteil. Wird unter anderem verwendet, um Orientierung im dreidimensionalen Raum mathematisch darzustellen. <a href="#">63–65</a>

- ROS-Subscriber** Objekt, das eine Verbindung zwischen einer Funktion und einem [Topic](#) herstellt. Kommen [Nachrichten](#) auf dem abonnierten [Topic](#) herein, so wird vom Subscriber die entsprechende Funktion aufgerufen und es wird ihr die [Nachricht](#) übergeben. Die Funktion kann die [Nachricht](#) verarbeiten. [65](#), [68](#)
- RosAria** Schnittstelle zwischen [ROS](#) und der Bibliothek [ARIA](#). [[Hedges \(2014b\)](#)] [66](#), [102](#)
- RS-232** [Electronic Industries Alliance \(EIA\)](#)-Standard einer seriellen Punkt-zu-Punkt-Schnittstelle mit Spannungspegel  $-15\text{ V}$  bis  $-3\text{ V}$  für logisch 1 und  $3\text{ V}$  bis  $15\text{ V}$  für logisch 0. [93](#)
- RS-453** [EIA](#)-Standard für Steckverbinder im Kleinspannungsbereich. Typischerweise eingesetzt im Audio- und Videobereich. Auch: Klinkenstecker [54](#)
- RS-485** [EIA](#)-Standard einer seriellen Bus-Schnittstelle, basierend auf zwei Adern 'A' und 'B'. Die Differenzspannung  $U_{diff} = A - B$  bestimmt den übertragenen Bitwert. Die Logikpegel sind festgelegt als  $U_{diff} < -3\text{V}$  für logisch 1 und  $U_{diff} > +3\text{V}$  für logisch 0. [[Wiesemann und Theis GmbH. \(2014\)](#)] [93](#)
- Service** Auf Request-Response-Paradigma basierende Kommunikationsschnittstelle zwischen [Nodes](#). Ermöglicht im Gegensatz zu [Topics](#) synchrone bidirektionale Datenübertragung zwischen [Nodes](#). [59](#), [69](#), [95](#)
- Singularität** Konstellation von Variablen, die dazu führt, dass das Ergebnis einer Funktion undefiniert ist. Beispiel:  $f(x) = \frac{1}{x}$  besitzt eine Singularität bei  $x = 0$ . Das Funktionsergebnis ist undefiniert. Die mechanische Singularität des Gimbal Lock ergibt sich aus der Definition der Rotationsmatrix für Roll-Nick-Gier-Winkel. [[Wikipedia \(2014d\)](#)] [34](#)
- Socket** Schnittstelle auf Basis des [IP](#)-Protokolls für Kommunikation zwischen Prozessen. Netzwerkfähig. Wird repräsentiert durch eine [IP](#)-Adresse und eine Portnummer. [59](#), [61](#)

---

<b>Stack</b>	Sammlung von <a href="#">Paketen</a> . Fasst mehrere Pakete zu einer Gruppe zusammen, deren Gemeinsamkeit ein bestimmter Zweck oder eine bestimmte Hardware ist. <a href="#">59</a> , <a href="#">62</a> , <a href="#">63</a> , <a href="#">75</a> , <a href="#">95</a>
<b>Stereovision</b>	Tiefenwahrnehmung mittels zweier optischer Sinnesorgane oder Sensoren, deren Sichtfelder sich überlagern. <a href="#">23</a>
<b>Stick</b>	Auf die Bedienung durch den Daumen ausgelegter Steuerknopf des Gamepads mit zwei Freiheitsgraden. Liefert je Freiheitsgrad einen von der Auslenkung abhängigen Dezimalwert zwischen -1 und +1. Die Mittelstellung entspricht dem Wert 0. <a href="#">55</a> , <a href="#">66</a> , <a href="#">67</a>
<b>Tiefsetzsteller</b>	Spannungswandler für Gleichspannung. Wandelt höhere in niedrigere Spannungen um. Auch: Abwärtswandler, step-down converter. <a href="#">41</a> , <a href="#">50</a>
<b>Topic</b>	Unidirektionaler, asynchroner Kommunikationskanal zwischen <a href="#">Nodes</a> . Basiert auf Producer-Consumer-Paradigma. Auf einem Topic können mehrere Consumer lauschen. Es können mehrere Producer <a href="#">Nachrichten</a> versenden, die von allen Consumern erhalten werden. <a href="#">59</a> , <a href="#">65</a> , <a href="#">66</a> , <a href="#">68</a> , <a href="#">69</a> , <a href="#">73</a> , <a href="#">96</a>
<b>Workspace</b>	Arbeitsbereich in dem die Software des <a href="#">TPR</a> entwickelt wird. Er liegt im Pfad <code>/home/tpr/tpr</code> . <a href="#">59</a> , <a href="#">62</a> , <a href="#">101</a> , <a href="#">102</a>

# Abkürzungsverzeichnis

<b>EIA</b>	Electronic Industries Alliance <a href="#">54</a> , <a href="#">96</a>
<b>FPS</b>	Frames per Second <a href="#">43</a>
<b>GUI</b>	Graphical User Interface <a href="#">26</a>
<b>HMD</b>	Head-Mounted Display <a href="#">23–28</a> , <a href="#">38</a> , <a href="#">44</a> , <a href="#">54</a> , <a href="#">55</a> , <a href="#">57</a> , <a href="#">58</a> , <a href="#">63</a> , <a href="#">76</a> , <a href="#">79</a> , <a href="#">85</a>
<b>HUD</b>	Head-Up-Display <a href="#">79</a>
<b>KE</b>	Kopfeinheit <a href="#">7</a> , <a href="#">11</a> , <a href="#">23–28</a> , <a href="#">30</a> , <a href="#">31</a> , <a href="#">33–42</a> , <a href="#">44–50</a> , <a href="#">54</a> , <a href="#">57–59</a> , <a href="#">62–64</a> , <a href="#">70</a> , <a href="#">71</a> , <a href="#">76</a> , <a href="#">80</a> , <a href="#">84</a> , <a href="#">85</a> , <a href="#">93</a> , <a href="#">101</a> , <a href="#">102</a>
<b>ME</b>	Mobileinheit <a href="#">3</a> , <a href="#">4</a> , <a href="#">6–8</a> , <a href="#">12–14</a> , <a href="#">20</a> , <a href="#">22–28</a> , <a href="#">31</a> , <a href="#">36–39</a> , <a href="#">45</a> , <a href="#">50–55</a> , <a href="#">57</a> , <a href="#">58</a> , <a href="#">60</a> , <a href="#">62</a> , <a href="#">63</a> , <a href="#">66</a> , <a href="#">68–71</a> , <a href="#">73</a> , <a href="#">74</a> , <a href="#">76–79</a> , <a href="#">81–86</a> , <a href="#">98</a> , <a href="#">102</a>
<b>MESS</b>	Steuerungssystem der ME <a href="#">3</a> , <a href="#">24</a> , <a href="#">27</a> , <a href="#">28</a> , <a href="#">38–40</a> , <a href="#">51–53</a> , <a href="#">58–62</a> , <a href="#">68</a> , <a href="#">69</a> , <a href="#">78</a> , <a href="#">85</a>
<b>MIT</b>	Massachusetts Institute of Technology <a href="#">12</a> , <a href="#">13</a>
<b>OPS</b>	Bedienerstation <a href="#">3</a> , <a href="#">4</a> , <a href="#">6</a> , <a href="#">7</a> , <a href="#">14</a> , <a href="#">20</a> , <a href="#">22–25</a> , <a href="#">27</a> , <a href="#">28</a> , <a href="#">31</a> , <a href="#">36–39</a> , <a href="#">45</a> , <a href="#">50</a> , <a href="#">51</a> , <a href="#">53–55</a> , <a href="#">57–63</a> , <a href="#">66</a> , <a href="#">68</a> , <a href="#">70</a> , <a href="#">73</a> , <a href="#">74</a> , <a href="#">78</a> , <a href="#">81</a> , <a href="#">84–86</a> , <a href="#">102</a>
<b>PoC</b>	Proof of Concept <a href="#">41</a> , <a href="#">44</a> , <a href="#">84</a>
<b>ROS</b>	Robot Operating System <a href="#">7</a> , <a href="#">12</a> , <a href="#">13</a> , <a href="#">28</a> , <a href="#">38</a> , <a href="#">39</a> , <a href="#">55</a> , <a href="#">58–62</a> , <a href="#">64</a> , <a href="#">70</a> , <a href="#">74</a> , <a href="#">75</a> , <a href="#">84–86</a> , <a href="#">93</a> , <a href="#">96</a>
<b>SIP</b>	Session Initiation Protocol <a href="#">39</a> , <a href="#">74</a>
<b>SLAM</b>	Simultaneous Localization And Mapping <a href="#">17</a> , <a href="#">24</a> , <a href="#">36</a> , <a href="#">57</a> , <a href="#">75</a> , <a href="#">76</a>

---

<b>TP</b>	Telepräsenz <a href="#">12</a> , <a href="#">15</a> , <a href="#">84</a>
<b>TPR</b>	Telepräsenzroboter <a href="#">5</a> , <a href="#">9–13</a> , <a href="#">15</a> , <a href="#">17–19</a> , <a href="#">21</a> , <a href="#">23</a> , <a href="#">27</a> , <a href="#">41</a> , <a href="#">57–59</a> , <a href="#">75</a> , <a href="#">84</a> , <a href="#">86</a> , <a href="#">94</a> , <a href="#">97</a> , <a href="#">101</a>
<b>TPS</b>	Telepräsenzsystem <a href="#">9</a> , <a href="#">10</a> , <a href="#">12</a> , <a href="#">14</a> , <a href="#">58</a>
<b>VR</b>	Virtual Reality <a href="#">54</a>
<b>XMLRPC</b>	Extensible Markup Language Remote Procedure Call <a href="#">59</a>

# Symbolverzeichnis

$G_{Kamera}$	N	Gewichtskraft der Kamera.
$J_{Kamera}$	$\text{kg m}^2$	Trägheitsmoment der Kamera in erster Näherung als Massenpunkt.
$M_{ges}$	N m	Gesamtes anliegendes Moment bei Winkelbeschleunigung $\alpha$ .
$M_{max}$	N m	Stillstandsmoment der Servomotoren.
$M_{stat}$	N m	Statisches Drehmoment am Servomotor.
$\alpha$	$\text{s}^{-2}$	max. Winkelbeschleunigung des Pitch-Servos.
$d$	m	Entfernung zwischen Schwerpunkt der Kamera und Pitch-Achse.
$m_{Kamera}$	kg	Masse der Kamera.

# A. Datenträger

Der dieser Arbeit beiliegende Datenträger hat folgenden Inhalt. Alle Pfade beziehen sich auf das Wurzelverzeichnis des Datenträgers.

## Wurzelverzeichnis /

Im Basisverzeichnis des Datenträgers befindet sich diese Arbeit als PDF-Datei. Darüber hinaus befinden sich dort die drei folgenden Ordner:

<b>Konstruktion</b>	Teile und Baugruppen der <a href="#">KE</a> . Die Dateien liegen im Solidworks-Format SLDPRT und SLDASM vor. Nicht enthalten sind Normteile aus der Solidworks-Toolbox. Beim Öffnen der Baugruppen ist dies ggfs. zu beachten.
<b>Software</b>	<a href="#">Workspace</a> des <a href="#">TPR</a> als TGZ-Archiv und als Ordner
<b>Quellen</b>	In der Arbeit verwendete Zitatquellen als PDF-Dateien

## /Konstruktion/Kopfeinheit

Einige der Dateien in diesem Pfad sind anzupassen, wenn die Servomotoren der Kopfeinheit durch stärkere Motoren ersetzt werden sollen. Die folgende Auflistung zeigt, welche Dateien angepasst werden müssen:

<b>ServoTraegerA.SLDPRT</b>	an größeres Servogehäuse anzupassen
<b>ServoTraegerB.SLDPRT</b>	an größeres Servogehäuse anzupassen
<b>Link_Yaw1.SLDPRT</b>	an größeres Servogehäuse und Horn anzupassen
<b>Link_Roll1.SLDPRT</b>	an größeres Servogehäuse und Horn anzupassen
<b>Nickgelenk_links.SLDPRT</b>	anzupassen, wenn sich die Achse des neuen Servos in der rechten Schulter bezüglich ihrer Lage nicht mehr mit der Achse des alten übereinstimmt

## /Software/ROS Workspace

Der [Workspace](#) im Pfad /Software/ROS Workspace/tpr/src und in der TGZ-Datei /Software/ROS Workspace/rosworkspace.tgz:/tpr/src weist folgende Struktur auf:

<b>base/launch</b>	<a href="#">Launchfiles</a> für Betriebsmodi <a href="#">ME</a> und <a href="#">OPS</a>
<b>dynamixel_motor</b>	Treiber und Controller für Dynamixel Servomotoren
<b>headunit/scripts</b>	<a href="#">Nodes</a> für <a href="#">KE</a> und Übersetzung des Headtrackings
<b>oculus</b>	Treiber und Viewer für Oculus Rift
<b>p3dx/scripts</b>	Teleoperation- <a href="#">Node</a> für die Verbindung zwischen SIXAXIS™-Controller und Pioneer 3-DX sowie <a href="#">Node</a> für Totmannknopf
<b>rosaria</b>	<a href="#">RosAria</a> Treiber für Pioneer 3-DX
<b>rqt_ops</b>	RQT Plugin für Betriebsüberwachung der <a href="#">ME</a>

# Versicherung über die Selbstständigkeit

Hiermit versichere ich, dass ich die vorliegende Arbeit im Sinne der Prüfungsordnung nach §16(5) APSO-TI-BM ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Hilfsmittel benutzt habe. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen habe ich unter Angabe der Quellen kenntlich gemacht.

Hamburg, 15.04.2014

\_\_\_\_\_  
Ort, Datum

\_\_\_\_\_  
Unterschrift