



Weiterentwicklung des Rovers SEAR (Small Exploration Assistant Rover) für die Teilnahme am DLR SpaceBot Cup 2015

Abschlussbericht

Dokument: SEAR-AB-2015
Version: 1.0
Datum: 26. Mai 2016
Förderkennzeichen: 50 RA 1420



Unterschriften

	Name	Datum	Unterschrift
Erstellt:	Lennart Kryza, B.Sc. Studentischer Mitarbeiter		
Geprüft:	Sascha Kapitola, M.Sc. Wissenschaftlicher Mitarbeiter		
Freigegeben:	Prof. Dr.-Ing. K. Briß (Verantwortlicher Hochschullehrer)		

Kontaktadresse

Technische Universität Berlin
Fakultät Transport- und Maschinensysteme
Institut für Luft- und Raumfahrt
Fachgebiet Raumfahrttechnik

Marchstraße 12, 10587 Berlin
Telefon: +49 (0)30 314-21305
Telefax: +49 (0)30 314-21306

<http://www.raumfahrttechnik.tu-berlin.de>

Partnerinstitut

Technische Universität Berlin
Fakultät IV für Elektrotechnik und Informatik
Distributed Artificial Intelligence Laboratory

Ernst-Reuter-Platz 7, 10587 Berlin
Telefon: +49 (0)30 314 74000 / 02
Telefax: +49 (0)30 - 314 74003

<http://www.dai-labor.de/>

Dokument: SEAR-AB-2015
Version: 1.0
Datum: 26. Mai 2016

SEAR - Small Exploration Assistant Rover
DLR SpaceBot Cup 2015



Abschlussbericht

Versionshistorie

Version	Datum	Änderungen	Bearbeiter
1.0	26. Mai 2016	Erstellung des Abschlussbericht	Lennart Kryza

Inhaltsverzeichnis

1	Kurze Darstellung	9
1.1	Aufgabenstellung	9
1.1.1	Auffinden und Identifizieren von unterschiedlichen Objekten	9
1.1.2	Transport und Objektmontage	9
1.2	Voraussetzungen	10
1.3	Planung und Ablauf des Vorhabens	11
1.3.1	Vorhabensbezogene Ressourcenplanung	11
1.4	Technischer Stand	14
1.4.1	Kinematik und Lokomotion	14
1.4.2	Manipulatorarm	14
1.5	Zusammenarbeit mit anderen Stellen	15
2	Eingehende Darstellung	16
2.1	Verwendung der Zuwendung und erzielte Ergebnisse	16
2.1.1	Ergebnisse im Bereich der Mechatronik	16
2.1.1.1	Kameramast	17
2.1.1.2	Körper/Wartung	17
2.1.1.3	Manipulator	18
2.1.1.4	Transportbehälter	21
2.1.2	Ergebnisse im Bereich der Software	23
2.1.2.1	Aufgabenplanung	23
2.1.2.2	Bodensegment (Überblick)	25
2.1.2.2.1	Motor Control GUI	26
2.1.2.2.2	Monitoring Tab	27
2.1.2.2.3	Control Tab	27
2.1.2.2.4	Emergency Break	27
2.1.2.3	Objekt Kalibrierungs GUI	28
2.1.2.3.1	Objekt Erkennungs GUIs	28
2.1.2.3.2	Objekt Erkennungs Kalibrierung	28
2.1.2.3.3	Object recognition operation control	28
2.1.2.4	Objekterkennung	29
2.1.2.4.1	Algorithmen	29
2.1.2.4.2	Applikationen zum Kalibrieren und Testen	31
2.1.2.4.3	Ausblick	31
2.1.2.5	SLAM - Simultaneous Localization and Mapping	32
2.1.2.5.1	Ansatz	32
2.1.2.5.2	Visuelle Odometrie	32
2.1.2.6	Manipulatorsteuerung	33
2.1.2.7	Kommunikation	36
2.1.2.7.1	Probleme im SpaceBot Cup 2013	36
2.1.2.7.2	Aufbau	36
2.1.2.7.3	Transportprotokoll	37
2.2	Nutzen und Verwendbarkeit der Ergebnisse	39



2.3 Fortschritte auf dem Gebiet des Vorhabens bei anderen Stellen	39
2.4 Veröffentlichungen	39

Abbildungsverzeichnis

1.1	Objekte des Spacebot Cups	10
1.2	SEAR CAD-Modell	14
1.3	Manipulatorarm	14
2.1	Kamera-Mast	17
2.2	Körper	18
2.3	Wartungssystem von SEAR	19
2.4	Arm 2.0 der Firma Invensciene LC	19
2.5	Knick-Arm Konzept und neues Konzept für den SEAR Manipulator	20
2.6	Greiferkonstruktion	21
2.7	Konstruktion des neuen Manipulators	21
2.8	Handgelenkantriebsstrang	22
2.9	Batterie-Pack-Transportsystem	22
2.10	Auszug aus SEARs Behaviour Tree	24
2.11	Motor GUI im Überblick	26
2.12	Engine-GUI overview	27
2.13	Darstellung der Objekteigenschaften	28
2.14	Video feeds mit der Objekterkennung	29
2.15	Object calibration GUI	30
2.16	Zielobjekte im Original und nach Filterung des jeweiligen Farbbereichs	31
2.17	SIFT-feature matches (RANSAC inliers)	32
2.18	Generierte Karte	33
2.19	ROS Paket, Manipulator Control	34
2.20	Aufbau der Kommunikationsstrecke	36
2.21	Prinzip der Datenübertragung	37



Tabellenverzeichnis

1.1	Personenstunden pro Arbeitspaket	12
1.2	Zeitplan	13
2.1	Gelenke des Manipulators	33
2.2	Projektbezogene Veröffentlichungen	40

1 Kurze Darstellung

1.1 Aufgabenstellung

Im November 2015 organisierte das deutsche Zentrum für Luft- und Raumfahrt (DLR) einen in Deutschland einzigartigen Robotik-Wettbewerb, das DLR SpacebotCamp. Die Aufgabenstellung ähnelte der im SpaceBot Cup 2013. Diese leitete sich aus einem typischen Explorationsszenario auf einer Planetenoberfläche ab. Die robotischen Systeme sollten in der Lage sein, in einem schwierigen Gelände Objekte aufzufinden, zu identifizieren und zu transportieren, um sie schließlich zu einem Gesamtsystem zusammenzufügen. Die besondere Herausforderung lag in der Voraussetzung, dass die robotischen Systeme alle Aufgaben mit dem höchsten Grad der Autonomie bewältigen sollten.

1.1.1 Auffinden und Identifizieren von unterschiedlichen Objekten

Auf der simulierten Planetenoberfläche wurden drei Objekte unterschiedlicher Form und Farbe von den Veranstaltern platziert. Dabei handelte es sich zum einen um ein gelbes, quaderförmiges Objekt, das einen Batteriepack darstellen sollte. Dessen Gewicht betrug ca. 800 g und konnte sich an einer beliebigen Stelle, meist hinter einem Hindernis versteckt, auf dem Gelände befinden. Das Zweite Objekt wurde als Wasserglas bezeichnet. Dieses zylinderförmige, blaue Objekt, wurde zum größten Teil mit Wasser gefüllt und konnte sich ebenfalls irgendwo auf dem Gelände befinden. Dieses wurde jedoch meist nicht versteckt, sondern auf einer freien Fläche abgestellt, so dass es auch von fliegende Systeme hätte ausfindig gemacht werden können. Das letzte Objekt sollte die Basis-Station darstellen, die jeweils eine Anschlussvorrichtung für die anderen beiden Objekte besaß.

Vor Beginn des Wettbewerbs erhielten die teilnehmenden Teams eine Höhenkarte, sowie eine Gelände-Draufsicht des Terrains, welche Aufnahmen eines sich im Orbit befindlichen Satelliten darstellen sollten. Auf der Draufsicht war lediglich die Basis-Station, so wie der Startpunkt, bzw. der Landeplatz des Systems zu erkennen. Die Höhenkarte des Geländes war graucodiert, welche eine grobe Aussage über die Höhen und Steigungsverhältnisse des Gebietes zuließ. Die Aufgabe des robotischen Systems bestand nun darin, die drei benannten Objekte aufzufinden und zu identifizieren. Um dies zu bewerkstelligen, musste es zum einen seinen Weg durch nahezu unbekanntes und unwegsames Gebiet bahnen und dabei das Gelände absuchen. Hier sollte die zuvor erhaltene Karte durch Scannen der Umgebung verfeinert und mit Bodenbeschaffenheits-, sowie Steigungswerten versehen werden. Wird ein Objekt auf dieser Explorationsfahrt entdeckt und identifiziert, ist es die Aufgabe des Systems, die genaue Position und den Objekttypen in die besagte Karte einzufügen.

1.1.2 Transport und Objektmontage

Wurden die Objekte einmal richtig identifiziert und ihre genaue Lage festgestellt, sollte das System befähigt sein, das Wasserglas und den Batteriepack aufzugreifen und zu der Basisstation transportieren zu können. Dazu gehörte zum einen, einen befahrbaren Weg zu den Objekten planen und befahren zu können und zum anderen, sie auf eine geeignete Weise aufzugreifen und schließlich für den Transport zu verstauen. Im Falle des Wasserglases wurde ein besonderes Augenmerk auf

den Transport geworfen. Würde das Glas beim Transport zu viel Wasser verlieren, hätte dies eine Waage auf der Basisstation gemerkt und die verlorene Menge an Wasser würde dann in Strafzeiten verrechnet werden.

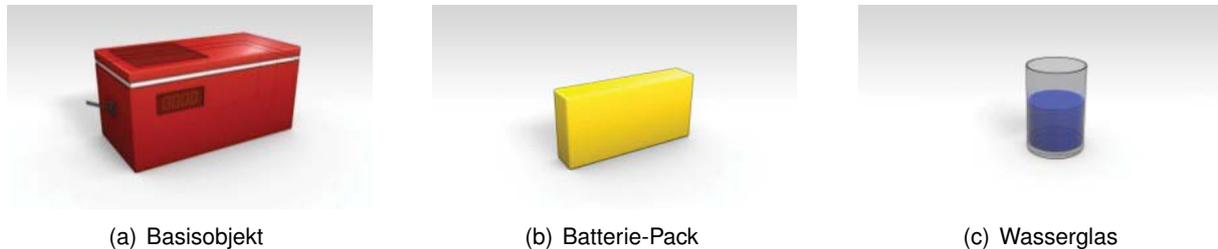


Abbildung 1.1: Objekte des Spacebot Cups

Zusätzlich zum Abstellen des Wasserglases sollte der Batteriepack in einen im Basisobjekt integrierten Steckplatz montiert werden. Wenn dies erledigt wurde, war es die Aufgabe des Systems, seine Beendigung der Aufgaben durch das Umlegen eines Schalter am Basisobjekt (s. Bild 1.1(a)) zu signalisieren. Dadurch würde das Basisobjekt aufleuchten und diese Aufgabe hätte somit als erfolgreich erledigt gegolten. Zum Schluss war es die Aufgabe des Systems sich einen Weg zurück zu seiner Startposition zu finden.

Für all diese Aufgaben hatte jedes System der Teams 60 Minuten Zeit, wobei in der 20sten und 40sten Minute ein vierminütiger Ausfall der Kommunikation simuliert wurde. In dieser Zeit war es nicht möglich, Daten an das System zu senden, sondern nur zu empfangen. Insgesamt hatte das Team die Möglichkeiten, drei sogenannte Checkpoints à fünf Minuten während des Laufs zu nehmen um die Kommunikation mit dem System aufzunehmen und eventuelle Änderungen vorzunehmen. Die Inanspruchnahme weiterer Checkpoints hätte Strafzeiten mit sich geführt.

1.2 Voraussetzungen

Das Institut für Luft- und Raumfahrt (ILR) gehört zur Fakultät V für Verkehrs- und Maschinensysteme der Technischen Universität Berlin. Das Institut bildet Ingenieure für Luft- und Raumfahrttechnik aus, die neben einer hohen fachlichen Qualifikation auch die Fähigkeiten besitzen, komplexe Zusammenhänge zu erfassen und zu bearbeiten, ein Projekt ganzheitlich zu betrachten und in einem Team zu arbeiten. Das Fachgebiet Raumfahrttechnik ergänzt das luftfahrttechnische Know-how in Forschung und Lehre mit raumfahrtspezifischen Grundlagen und Anwendungen. Das Ziel des Fachgebietes ist es, Systemingenieure für die Raumfahrt auszubilden und auf die heutigen Marktanforderungen vorzubereiten. Der Entwurf, die praktische Realisierung und der Betrieb von Kleinsatellitenmissionen mit Studenten stehen im Mittelpunkt der Lehre und Forschung. Am ILR wurden bereits 8 Mikro- und Nanosatelliten erfolgreich gebaut und in den Orbit gebracht und das ILR ist an der BIRD-Satellitenmission des Deutschen Zentrums für Luft- und Raumfahrt beteiligt. Das Know-how der TUBSAT-Flotte von Prof. Renner wird nach seinem Ruhestand in 2006 im Fachgebiet Raumfahrttechnik in Form einer Forschungsgruppe gepflegt und weiterentwickelt. Derzeit sind die Satellitenprojekte TUBIN, TechnoSat, S-NET und BEESAT-4 in Bearbeitung.

Für die Forschung und Lehre verfügt das Institut über moderne Forschungs- und Versuchsanlagen. Im Raumfahrtbereich stehen zur Verfügung: die TUBSAT-Satellitenflotte, ein Teststand zur Lageregelung von Satelliten, ein Satellitenentwurfslabor, zwei Satellitenbodenstationen, Satellitenintegrations- und Testlabore und ein Raumflugkontrollzentrum. Die Forschung wird analytisch, numerisch und ex-

perimentell betrieben. Für die Umsetzung des Roverprojektes wurde ein eigenes Planetary Rover Laboratory eingerichtet, welches auch über ein abgegrenztes Missionskontrollzentrum zum Test des Roverbetriebs unter Wettbewerbsbedingungen verfügt.

1.3 Planung und Ablauf des Vorhabens

Die geplanten Arbeiten gliederten sich in 21 Arbeitspakete, von denen einige nachfolgend ausführlich beschrieben werden. Alle Arbeiten fanden innerhalb des Förderzeitraums statt und wurden planmäßig mit der Austragung des Wettbewerbs im November abgeschlossen. Da das genaue Datum des Wettbewerbs zum Zeitpunkt der Antragstellung noch nicht bekanntgegeben wurde, wurde das Vorhaben bis Ende November geplant.

1.3.1 Vorhabensbezogene Ressourcenplanung

Bei der Betrachtung dieser Ressourcenplanung ist zu beachten, dass sich diese nur auf das so bezeichnete Kernteam bestehende aus Studentischen Hilfskräften bezieht. Nicht mit einbezogen sind zusätzliche Leistungen von studentischen Teilnehmern der Lehrveranstaltung, so wie Abschlussarbeiten und die Hilfe von Wissenschaftlichen Mitarbeitern.

Abschlussbericht

Arbeitspaketname	Zeitraum	Personenstunden*
AP 1000 - Management	03/2015 - 11/2015	40
AP 1100 - Projektplanung- und Organisation	03/2015 - 11/2015	0
AP 1200 - Öffentlichkeitsarbeit	03/2015 - 11/2015	40
AP 2000 - Mission	03/2015 - 07/2015	360
AP 2100 - Missionsplanung	03/2015 – 04/2015	60
AP 2200 - Missionsbetrieb	03/2015 – 07/2015	120
AP 3000 - Hardware	03/2015 - 09/2015	420
AP 3100 – Struktur und Mechanismen	03/2015 – 03/2015	120
AP 3200 – Bordelektronik	03/2015 – 08/2015	60
AP 3300 – Fahrwerk	03/2015 – 04/2015	40
AP 3400 – Manipulator	03/2015 – 08/2015	160
AP 3500 – Bodenhilfseinrichtung	03/2015 – 03/2015	40
AP 3600 – UAV Hardware	03/2015 – 09/2015	0
AP 4000 – Software	03/2015 – 11/2015	830
AP 4100 – Aufgabenplanung	03/2015 – 11/2015	260
AP 4200 – Bodensegment	03/2015 – 09/2015	160
AP 4300 – Objekterkennung	03/2015 – 08/2015	150
AP 4400 – Navigation (SLAM)	03/2015 – 11/2015	140
AP 4500 – Manipulatorsteuerung	03/2015 – 10/2015	80
AP 4600 – Lokomotion	03/2015 – 03/2015	40
AP 4700 – UAV Software	03/2015 – 09/2015	0
AP 4800 – Rover/UAV Kooperation	04/2015 – 09/2015	0
AP 5000 – Tests	03/2015 – 11/2015	690
AP 5100 – Simulation	03/2015 – 05/2015	100
AP 5200 – Testgelände	03/2015 – 06/2015	30
AP 5300 – Testkampagne	05/2015 – 11/2015	560
Summe		2340

Aufgaben/Zeit 2015	Mär	Apr	Mai	Jun	Jul	Aug	Sep	Okt	Nov
AP 1000 - Management									
AP 2000 - Mission									
AP 2100 - Missionsplanung t									
AP 2200 - Missionsbetrieb									
AP 3000 - Hardware									
AP 3100 - Struktur und Mechanismen									
AP 3200 - Bordelektronik									
AP 3300 - Fahrwerk									
AP 3400 - Manipulator									
AP 3500 - Bodenhilfseinrichtungen									
AP 3600 - UAV Hardware									
AP 4000 - Software									
AP 4100 - Aufgabenplanung									
AP 4200 - Bodensegment									
AP 4300 - Objekterkennung									
AP 4400- Navigation (SLAM)									
AP 4500- Manipulatorsteuerung									
AP 4600 - Locomotion									
AP 4700 - UAV Software									
AP 4800 - Rover/UAV Kooperation									
AP 4000 - Tests									
AP 5100 - Simulation									
AP 5200 - Testgelände									
AP 5300 - Testkampagne									

Tabelle 1.2: Zeitplan

1.4 Technischer Stand

1.4.1 Kinematik und Lokomotion

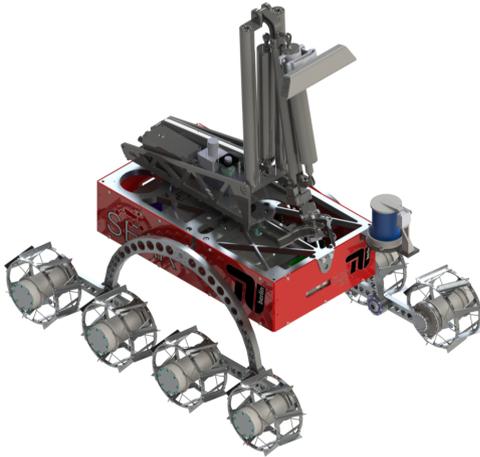


Abbildung 1.2: SEAR CAD-Modell

Im Zuge der Vorbereitungen auf dem SpacebotCup 2013 wurden umfangreiche Analysen von ausgewählten Fahrwerken mit Hilfe der Matlab/Simulink Toolbox SimMechanics durchgeführt [Nip11]. Dabei lag der Schwerpunkt auf der Auswertung der Reaktionskräfte, die sich bei den einzelnen Fahrwerken ergeben, da diese zum einen Aufschluss auf die Gewichtskraftverteilung und die Traktion, sowie die Drehmomenteigenschaften der Fahrwerke geben. Des Weiteren wurde eine Untersuchung der Verschiebung der Räder entlang der x-Achse durchgeführt. Aus diesen Ergebnissen kristallisierte sich ein Fahrwerk, welches in den meisten Test-Kategorien am besten abschloss. Der Rover ist somit kinematisch nach dem Rocker-Parallelogramm-Bogie-Konzept aufgebaut, welches ihn ermöglicht, Stufen von bis zu 150 mm und Steigungen von bis zu 25° überwinden (s. Abb. 1.2). Des weiteren wurde der Rover ist für weiche Untergründe wie Sand oder Lehmboden, wie sie zum größten Teil auf

dem Mars und dem Mond zu finden sind, ausgelegt.

1.4.2 Manipulatorarm

Insgesamt besitzt der Manipulator fünf Freiheitsgrade und weist mit dem in Abb. 1.3 angeschlossenem Greifer eine maximale Reichweite von 725,5 mm auf. Der Roboterarm besitzt eine Masse von 1,40 kg und ist für eine maximale Nutzlastmasse von 1 kg ausgelegt. Um den Energieverbrauch so gering wie möglich zu halten, sind die Hauptgelenke des Manipulators selbsthemmend ausgelegt, was die Benötigung eines Haltemoments eliminiert. Basierend auf diesem Marsmodell des Manipulators wurde ein Labormodell erstellt [Son12]. Bei der Substitution wurde besonders darauf geachtet, dass der prinzipielle Aufbau des Marsmodells erhalten bleibt, so dass ein direkter Rückschluss auf die Funktionsweise des Marsmodells möglich ist. Das Labormodell des Manipulators besitzt dabei eine maximale Reichweite von 722,5 mm bei einer Masse von 1,42 kg.

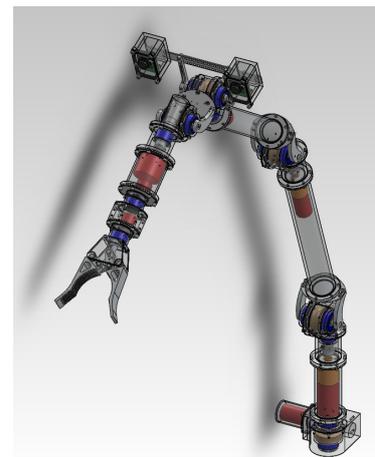


Abbildung 1.3: Manipulatorarm



1.5 Zusammenarbeit mit anderen Stellen

Im Zuge des Projektes wurde eng mit den Mitarbeitern des DAI-Labors zusammengearbeitet, welches auch an der Technischen Universität Berlin angesiedelt ist. DAI steht für Distributed Artificial Intelligence und die Forscher sind Experten für die Entwicklung von Algorithmen auf dem Gebiet der künstlichen Intelligenz.

Sie haben das SEAR Team während der Entwicklungsphase beratend zur Seite gestanden und sollten den Rover mit einem von ihnen entwickelten UAV (Unmanned Air Vehicle) im Wettbewerb unterstützen.



2 Eingehende Darstellung

2.1 Verwendung der Zuwendung und erzielte Ergebnisse

Die Zuwendung wurde ausschließlich für das Erreichen der wissenschaftlich-technischen Ziele des Vorhabens eingesetzt. Die Zuwendung war notwendig, um die Kernziele des Vorhabens zu erreichen. Über die Zuwendung hinaus, wurde ein beträchtlicher Anteil in Eigenleistung durch den Zuwendungsempfänger eingebracht (Personalstunden, Anschaffungen, Reisemittel). Das Fachgebiet Raumfahrttechnik bietet für Forschungsvorhaben diverse Laborräume mit entsprechender Ausstattung. Hierzu gehören u.a. Werkzeug, Messgeräte, Laborutensilien, sowie viele herkömmliche elektronische Bauteile. Des Weiteren werden Mitarbeitern Computerarbeitsplätze, sowie Laborarbeitsplätze bereitgestellt. Das Institut für Luft- und Raumfahrt stellt den Fachgebieten Dienstleistungen in den Bereichen Metall- und, Kunststofffertigung zur Verfügung.

Das Team im Projekt setzte sich aus verschiedenen Personenkreisen zusammen. Wissenschaftliche Mitarbeiter waren in freiwilliger Zuarbeit für die Koordinierung und fachliche Unterstützung verantwortlich. Im Rahmen einer Lehrveranstaltung wurden die Entwurfsarbeiten von ca. 20 Studierenden aus verschiedenen Fachrichtungen maßgeblich vorangetrieben. Nach Abschluss der Lehrveranstaltung haben Studierende freiwillig oder im Rahmen einer Abschlussarbeit das Projekt weiter begleitet. Die Arbeiten um die Mechatronik und Software des autonomen Systems erforderten aufgrund ihrer Komplexität klare Definitionen ihrer Schnittstellen. Die Grundlegende Infrastruktur für die Zusammenarbeit in einem großen Team musste von einem festen Kernteam über die Projektlaufzeit bereitgestellt und gewartet werden. Aus genannten Gründen wurden mehrere studentische Hilfskräfte eingestellt. Es handelte sich um zwei Hilfskräfte, die vorrangig im Bereich der Konstruktion tätig waren und drei Hilfskräfte, die maßgeblich für die Softwareentwicklung verantwortlich waren. Die Aufgabe des Systemingenieurs fiel ebenfalls einer der studentischen Hilfskräfte aus dem Bereich Software zu.

Ein Großteil der Mittel wurde für die Weiterentwicklung der Hardware des robotischen Systems eingesetzt. Besonders ins Gewicht sind hierbei die Kosten der Getriebemotoren für das den neuen Manipulator mit sechs Freiheitsgraden gefallen, sowie die Fertigung eines neuen Korpus.

Weitere Kostenpunkte sind unter anderem durch die Notwendigkeit von Bodenhilfseinrichtungen, der Einrichtung einer Bodenstation, sowie des Aufbaus eines Testgeländes entstanden. Eine genaue Auflistung der Ausgaben ist dem Verwertungsplan zu entnehmen.

2.1.1 Ergebnisse im Bereich der Mechatronik

Anknüpfungspunkt für die Weiterentwicklung den Robotiksystems SEAR war das 2013 im Zuge des SpacebotCups entwickelte System. Die dort gesammelten Erfahrungen und Ideen wurden an das bestehende System adaptiert. Hierzu gehörten zu einem großen Teil die neuen Hauptkomponenten von SEAR.

- Der Körper und die dazugehörigen mechatronischen Komponenten
- Der Manipulator

– Der Kameramast

2.1.1.1 Kameramast

2013 besaß SEAR zwei Kinect Kameras, die jeweils nach vorne und nach hinten gerichtet waren. Diese dienten der Kartierung, der Objekterkennung so wie der Hindernisvermeidung. Die geringe Anzahl der Kameras war zwar ausreichend, verschaffte allerdings nur einen geringen Überblick über die Umgebung. Um einen Gewinn an Überschaubarkeit zu erhalten, wurde entschlossen, ein System zur optischen Erfassung der Umgebung zu entwerfen.

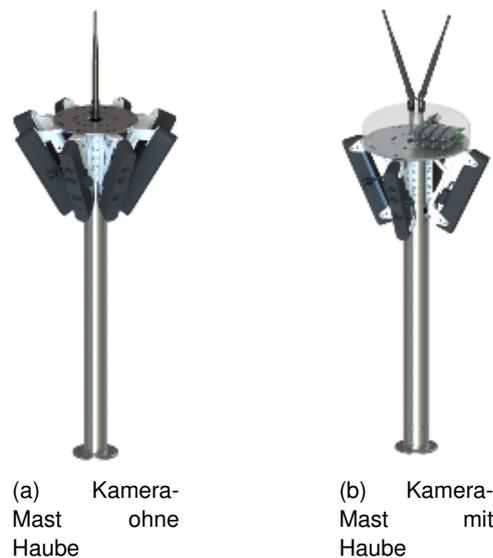


Abbildung 2.1: Kamera-Mast

Die Abbildungen 2.1(a) und 2.1(b) zeigen das entwickelte Design des Systems Kamera-Mast. Bei dem Entwurf wurde insbesondere darauf geachtet, einen Sichtbereich von 360° abzudecken. Verwendet wurden 8 neue Asus Xtion Tiefenbildkameras, die in einem vertikalem Abstand von ca. einem Meter vom Körper angebracht wurden. Diese dienen sowohl der Objekterkennung, als auch der Kartierung und Hinderniserkennung.

2.1.1.2 Körper/Wartung

Der Körper beinhaltet die *Talente* des Rovers. Hier sind die benötigten Komponenten wie unter anderem der Computer und die Hazcams (Xtion) angebracht (s. Abb. 2.2(b)), die das autonome System benötigt um sich zu orientieren. Je eine Xtion ist nach Vorne und nach Hinten gerichtet, um die jeweiligen Fahrtrichtungen im Blick zu haben.

Herzstück des Körper stellt der Computer da und liegt im Zentrum des Körpers. An ihm sind die Kameras und die Motioncontroller angeschlossen. Er übernimmt die Auswertung der Daten, fällt Entscheidungen und steuert die Motoren, so wie den Manipulator

Die am Fahrwerk angebrachten Motoren für das Fahrwerk, so wie dem Manipulator benötigen jeweils einen Motioncontroller. Insgesamt sind 12 Stück sind an äußeren Seiten des Körpers angebracht, da von dort Kabel nach außen führen müssen und somit ihr Weg im Körper reduziert wird.

Zusätzlich benötigt das System noch elektrische Energie, welche durch neun zusammengeschaltete Batteriezellen geliefert wird und die restlichen Komponenten mit Strom versorgt. Die neu am Institut für Luft und Raumfahrttechnik entwickelte, sogenannte Bordelektronik steuert die ordnungsgerechte Stromversorgung.

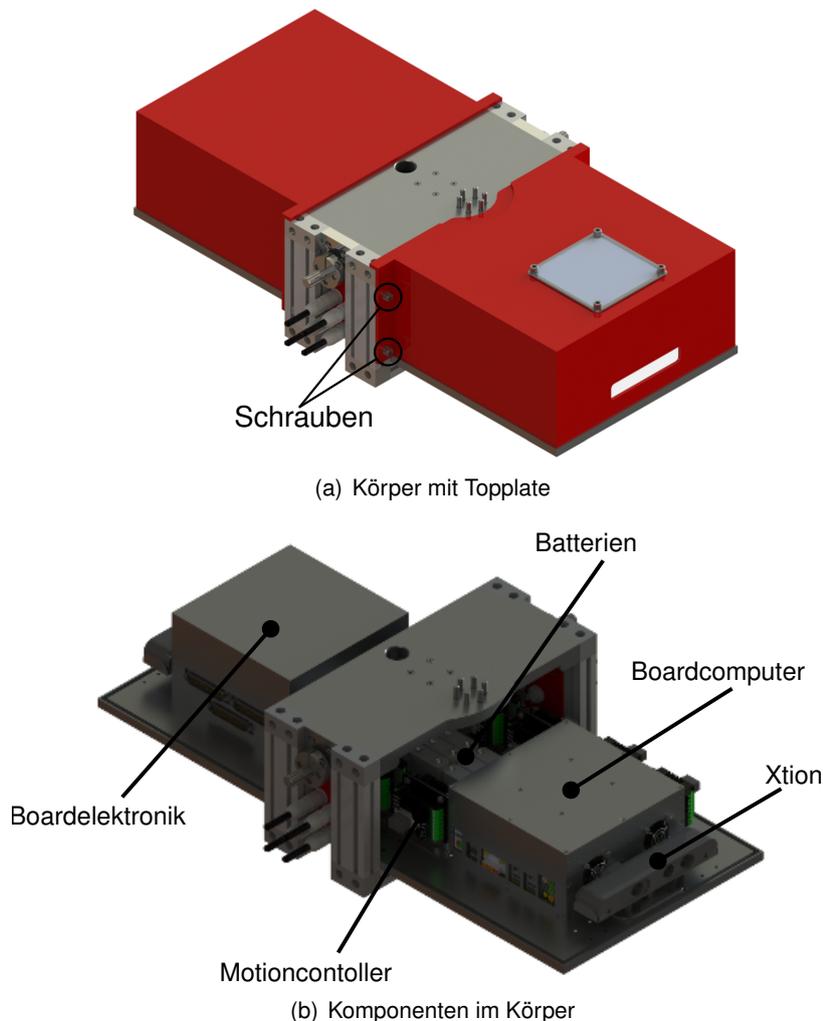


Abbildung 2.2: Körper

Im zusammengebauten Zustand hängt der Körper wie schon bei dem vorherigem Körper an dem Fahrwerk. Die in Abb. 2.2(a) rot abgebildeten sog. Abdeckungen sind passgerecht an den Körper angepasst. Sie werden in auf der Oberfläche der Bodenplatte des Körpers eingefräste Nuten eingesteckt und schließlich mit jeweils vier Schrauben an den Vierkantprofilen fixiert. Dies erleichtert zum einen die Wartung des System, da es für eine sehr gute Erreichbarkeit der integrierten Komponenten sorgt (s. Abb. 2.3). Zum anderen konnte das Gesamtgewicht des Körpers im Vergleich zu dem Vorherigen Entwurf drastisch gesenkt werden.

2.1.1.3 Manipulator

Der derzeitige Manipulator von SEAR mit dem Namen ARM 2.0 (s. Abb.2.4) ist ein Kaufteil der Firma Invenscience LC und arbeitet größtenteils mit Linearmotoren, was den zu erreichenden Arbeitsraum

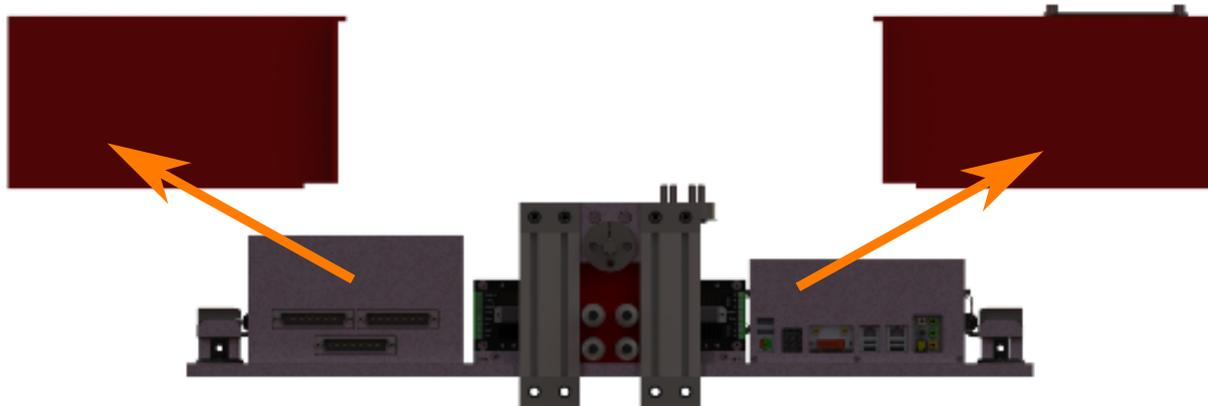


Abbildung 2.3: Wartungssystem von SEAR

einschränkt (Gelenke, wie das Schulter- und Ellenbogengelenk besitzen dadurch eine Bewegungsfreiheit geringer als 90°). Zusätzlich besitzen die einzelnen Gelenke ein erhebliches Spiel, woraus eine ungünstige Positionier- und Wiederholgenauigkeit des Manipulators entsteht und die Ansteuerung und präzise Bewegungen des Manipulators erschwert. Weitere nachteilige Eigenschaften des ARM 2.0 sind ein hohes Gewicht und große geometrische Dimensionen, wodurch vor allem durch die ausladende Basis viel Bauraum auf der Oberseite des Rovers eingenommen wird.



Abbildung 2.4: Arm 2.0 der Firma Invenscience LC

Es wurde sich daher für die Konstruktion eines eigenen Manipulators in der folgenden Projektphase entschieden. Innerhalb einer Lehrveranstaltung wurden aufbauend auf einer Diplomarbeit zur Konstruktion eines Manipulators für eine Marsmission, neue Konzepte von Studenten erarbeitet und ausgewertet. Dabei sind folgende Anforderungen an einen neuen Manipulator herausgestellt worden:

- Heben und bewegen einer Nutzlastmasse von wenigstens 1 kg

- Integration des Manipulators auf der Oberseite des Rovers
- Arm mit 6 Freiheitsgraden
- Die Längen der Armsegmente müssen das Greifen vom Boden ermöglichen, ohne dass der Arm mit dem Rover kollidiert
- Möglichst geringes Gewicht durch Leichtbau
- Minimiertes Spiel in den Gelenken
- Möglichst geringe Produktionskosten
- Möglichst geringer Bauraum vor allem bzgl. der Basis
- Rückgabe der Gelenkstellungen (Winkel) soll ermöglicht werden

Mithilfe dieser Anforderungen wurde ein Manipulatorkonzept mit Knick-Arm Konstruktion und geringem Gelenkspiel ausgewählt, das mittels eines Antriebsstrangs bestehend aus einem DC Servomotor, einem Planetengetriebe und einem Schneckengetriebe in jedem Gelenk erreicht werden sollte. Das Konzept eines Knick-Arm Manipulators und das in der Lehrveranstaltung entstandene konzeptionelle Design ist in Abb. 2.5 zu erkennen. Der vergrößerte Arbeitsraum erhöht die allgemeine Funktionsfähigkeit des gesamten Robotiksystems erheblich und zusätzlich wird die Toleranz in der Fahrwerksansteuerung erhöht. Die verbesserte Mobilität des neuen Arms ermöglicht weiterhin eine erhöhte Flexibilität beim Heranfahren an ein zu greifendes Objekt, welche bisher sehr genau und präzise sein musste.

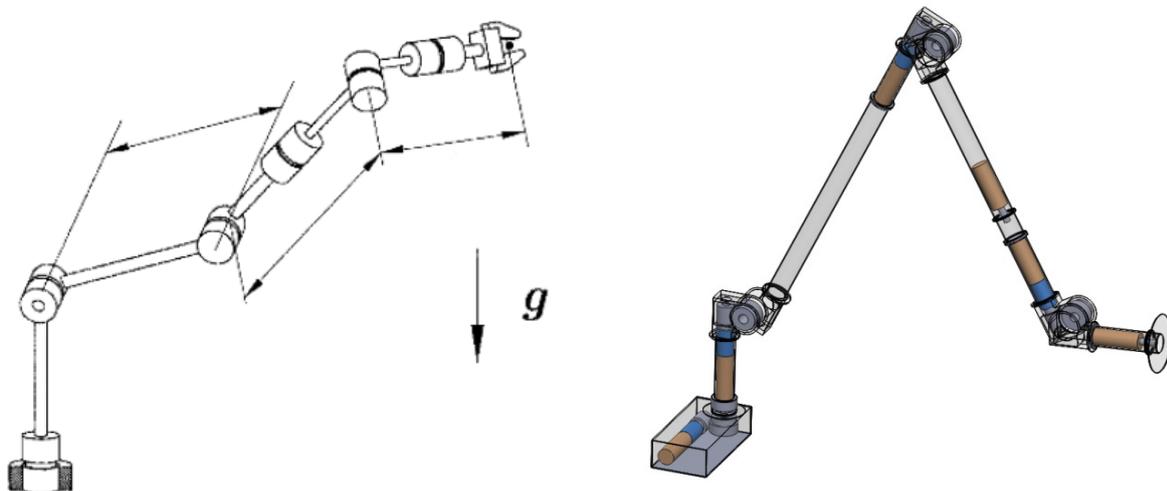


Abbildung 2.5: Knick-Arm Konzept und neues Konzept für den SEAR Manipulator

Ausgehend von diesem Konzept wurde eine Konstruktion des Arms aus hauptsächlich Aluminium und bürstenlosen DC Servomotoren gekoppelt mit Planetengetrieben von Faulhaber und selbsthemmenden Schneckengetrieben ausgelegt. Diese Motoren ermöglichen eine präzise und zügige Ansteuerung der Gelenke, während die Schneckengetriebe neben der gelieferten Übersetzung zusätzlich, aufgrund ihrer Selbsthemmung, die Möglichkeit bieten, die Motoren nicht permanent mit Spannung versorgen zu müssen. Daher kann eine Stellung des Manipulators angefahren und auch ohne anliegende Spannung gehalten werden, was zu einer potentiellen Einsparung elektrischer Energie führt. Außerdem wurde ein neuer Greifer entwickelt, der darüber hinaus auch mit dem ARM 2.0 schnittstellenkompatibel und in Abb. 2.6 abgebildet ist. Das Design des neuen Manipulators ist in

Abb. 2.7 dargestellt. Ein weiterer Vorteil des neuen Designs ist, dass der Arm während des Fahrens, oder wenn er momentan nicht in Benutzung ist, zusammen geklappt und auf SEAR abgelegt werden kann. Dies verlagert den Schwerpunkt des Rovers mittig und stabilisiert so das Fahrverhalten. Auch der zusammengeklappte Arm ist in Abb. 2.7 zu erkennen.

Einige Komponenten des neuen Arms werden derzeit konstruktionstechnisch noch finalisiert, wobei sich einige zu fertigende Teile bereits in der Fertigung befinden bzw. bereits gefertigt sind. Teile des Antriebsstrangs für das Handgelenk mit integriertem Getriebemotor sind in Abb. 2.8 dargestellt. Es kann davon ausgegangen werden, dass der Manipulator bald implementiert und anschließend Tests durchgeführt werden können. Die Ansteuerung des neuen Manipulators wird in Form einer Abschlussarbeit durch ein Mitglied des Projektes entwickelt.

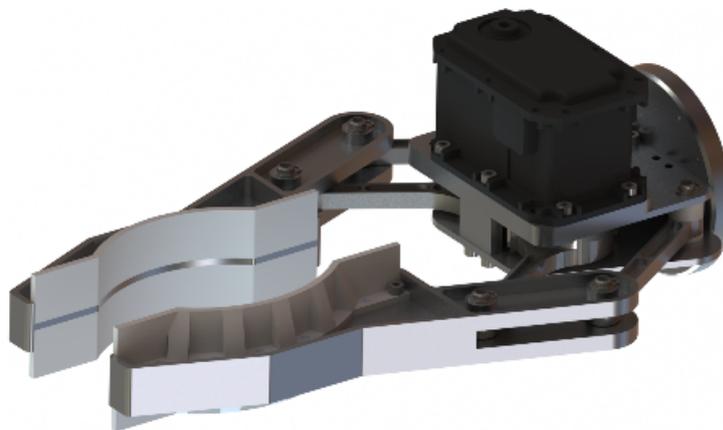


Abbildung 2.6: Greiferkonstruktion



Abbildung 2.7: Konstruktion des neuen Manipulators

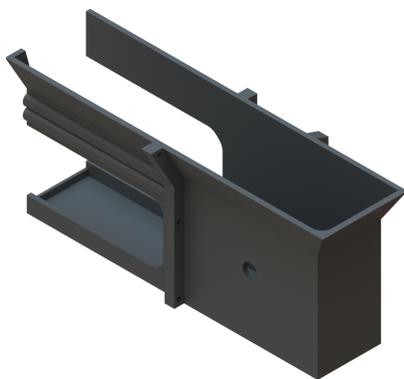
2.1.1.4 Transportbehälter

Aufgabe des System war es die gegriffenen Objekte zur Basisstation zu bringen. Dafür musste der Rover die Objekte verstauen können und beim Erreichen der Basisstation wieder Greifen und logisch mit der Basisstation zusammensetzen können. Hierfür benötigte der Rover spezielle Transportbehälter, die die Objekte aufnehmen konnten und aus der richtigen Position ein erneutes Greifen zulassen. Dies bedeutet, dass ihre Lage in der Reichweite des Manipulators liegen musste. Weiterhin mussten die die Objekte sicher verstauen können.

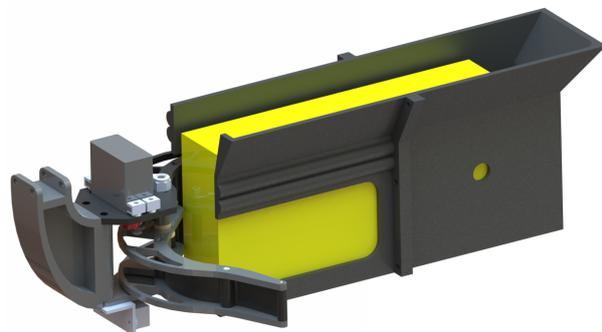


Abbildung 2.8: Handgelenkantriebsstrang

Wie schon erwähnt, musste beim Transport des Batteriepacks darauf geachtet werden, dass das Objekt in einer hoch-kanten Position gegriffen wird, da es in dieser Lage letztendlich in die Basisstation eingefügt werden sollte. Somit lag eine der Anforderungen für dieses Transportsystem darin, dass das Batteriepack immer in dieser Position gegriffen werden kann, egal wie es in das Transportsystem eingesetzt wurde. Der obere, Trichterförmige Teil des Systems (s. Abb. 2.9(a)) dient dazu, das hineinfallende Objekt in seine Richtige Lage zu drehen. Der Ausschnitt auf der linken Seite dient dem Zugang des Greifers um die darin liegende Batterie korrekt greifen zu können. Auch dieses System wurde am vorderen Teil des Rovers angebracht (s. Abb. 2.9(b)).



(a) Batteriehalterung



(b) Greifvorgang

Abbildung 2.9: Batterie-Pack-Transportsystem

2.1.2 Ergebnisse im Bereich der Software

Die Softwareentwicklung wurde zu einem Schwerpunkt für die Arbeiten in Hinblick auf den SpaceBot Cup 2015 gemacht. SEAR's konstruktionstechnische Grundlage wurde bereits 2013 geschaffen und während diese in 2015 mit Neuentwicklungen weiter ausgearbeitet werden sollte, galt es im Bereich der Software nicht nur weiterführende Arbeiten zu leisten, sondern komplett neue Implementierungen zu etablieren. Es werden im Folgenden die Ergebnissen einer Arbeitspakete vorgestellt. Es handelt sich bei dem vorgestellten vor allem um Neuentwicklungen, die maßgeblich für das Bestehen der Qualifikation und die erfolgreiche Teilnahme an der Leistungsshow im November verantwortlich waren.

2.1.2.1 Aufgabenplanung

Die Forderung der Autonomie ist eine zentrale Thema auch im Jahr 2015 für den DLR SpaceBot Cup gewesen. SEAR verfolgt einen hierarchischen Autonomieansatz. Die bedeutet, dass autonome Prozeduren in den jeweiligen Subsystemen bereits von den Entwicklern vorgesehen werden. Dies hatte einerseits den Vorteil, dass die komplexe Aufgabe der Entwicklung von spezifischer Autonomie vom jeweiligen Fachpersonal durchgeführt wurde, andererseits aber auch die Testbarkeit des Systems enorm gesteigert werden konnte.

Eine funktionieren Autonomie innerhalb eines Subsystems schafft jedoch noch nicht ein autonome Gesamtsystem, welches selbstständig das Missionsszenario absolviert. Hierfür ist es nötig, dem Rover die Fähigkeit zur Entscheidungsfindung zu geben, damit die geeignet Subroutinen zur richtigen Zeit ausgelöst werden. Auch muss verhindert werden, dass der Rover nicht weiß was zu tun ist und keine weiteren Aktionen ausführt. Bereits im Vorfeld des Projektes im Jahr 2015, wurden in Lehrveranstaltungen Untersuchungen durchgeführt, welche Ansätze am besten geeignet sind, um eine missionsspezifische Autonomie zu entwerfen. Es wurde sich für den Ansatz der **Behaviour Trees** entschieden. Diese Entwicklung stammt aus der Computerspielindustrie, in welches das Verhalten von computergesteuerten Charakteren kontrolliert werden muss.

Behaviour Trees bestehen aus einer Reihe von Logikbausteinen, welches verschiedene Typen haben können und selbst auch aus denselben Bausteinen bestehen können. Es ergibt sich eine *Parent-Children*-Beziehung in der Implementierung. Die Typen unterscheiden sich grundsätzlich in der Form ihrer Rückmeldung an die jeweiligen *Parents* und den Kriterien für die Rückmeldung. Für die Rückmeldung werden die drei Zustände *SUCCESS* (Erfolg), *FAILURE* (Fehlschlag) und *RUNNING* (Laufend) verwendet, welche jeweils dem Rückgabewert einer *run*-Funktion (der Hauptfunktion eines Elementes) entsprechen. Die Funktionalität soll anhand einiger grundlegenden Elemente und Beispiele verdeutlicht werden. Es werden hierbei die Elemente *Sequence*, *Selector*, *Condition* und *Task* erklärt.

1. Condition

Die *Condition* ist das einfachste Logikelement und dient vor allem der Überprüfung von Bedingungen anhand externer Daten. Die Bedingung vermeldet einen *SUCCESS*, wenn sie erfüllt ist und ansonsten einen *FAILURE*. Während dieser Baustein so wirken könnte, als bedürfe er keiner eigenen Charakterisierung, übernimmt er eine wichtige Aufgabe bei der Vermeidung unnötiger Prozessierung und bei der Einhaltung des Entwurfsmusters.

2. Sequence

Die *Sequence* gibt einen Erfolg zurück, wenn alle ihre Tochterbausteine einen einen Erfolg berichteten. Sobald eines der *Children* einen Fehlschlag meldet, gibt sie selbst ebenfalls einen Fehlschlag zurück.

Sie wird vor allem dafür eingesetzt, die Durchführung einer bestimmten Aufgabe zu überwachen, die aus mehreren notwendigen Teilaufgaben besteht.

3. Selector

Der *Selector* führt nacheinander die *run*-Methode seiner *Children* aus, bis eine von diesen einen Erfolg meldet. In diesem Falle berichtet er selbst ebenfalls einen Erfolg. Der Baustein ignoriert somit Fehlschläge und schlägt nur fehl, wenn keiner der Tochterbausteine erfolgreich ist. Dieser Baustein wird vor allem dafür verwendet, um Verhalten in ihrer Priorität zu ordnen.

4. Task

Tasks repräsentieren den Vorgang einer abgegliederten Aufgabe, die in der Regel von sehr geringer Komplexität ist und sich nicht mehr weiter unterteilbar. Sie übermitteln je nach Erfolg der Aufgabe ein *SUCCESS* oder *FAILURE*. Ein Beispiel für eine *Task* wäre das Schließen des Greifers.

Im folgenden ist ein Auszug aus dem Behaviour Tree von SEAR zu sehen, wie er während des SpaceBot Camps 2015 zum Einsatz kam. In Abbildung 2.10 ist ersichtlich, dass die Hauptroutine,

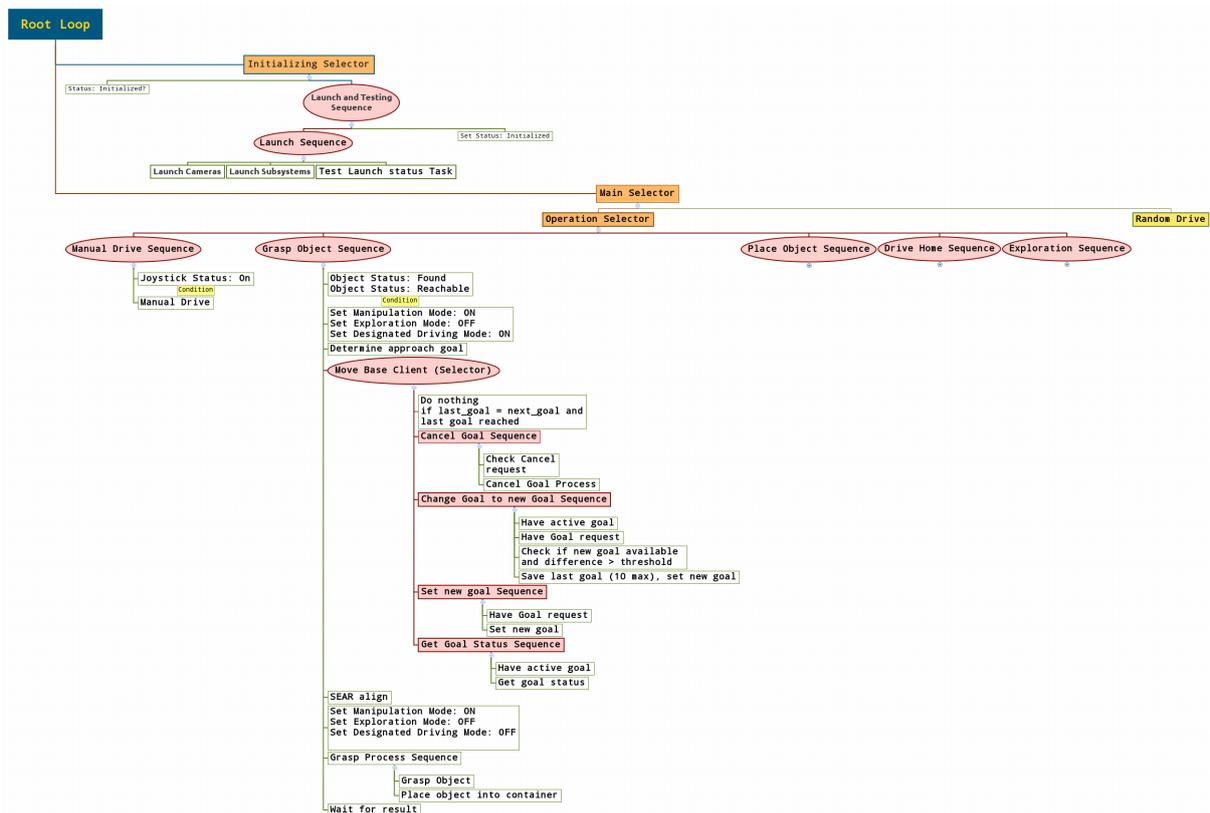


Abbildung 2.10: Auszug aus SEARs Behaviour Tree

in welcher sich der Rover hauptsächlich befindet, der Operation Selector ist. In diesem werden verschiedene mögliche Operationsszenarien geprüft und ggf. initiiert. Für die meisten Aufgaben sind in der Mission die Voraussetzungen nicht gegeben, weshalb der Rover einer Explorationsstrategie folgt, bis er z.B. eines der Objekte findet.

2.1.2.2 Bodensegment (Überblick)

Das SEAR-System wurde während der Entwicklung und dem SpaceBot Cup im Jahr 2013 vollständig über ein Kommandozeileninterface überwacht und kontrolliert. Dieses ist ausschließlich textbasiert, welches die folgenden Nachteile zur Folge hatte:

- Informationen waren oft unstrukturiert oder gar unklar
- Darstellung der Ausgaben nur bedingt anpassbar
- Eingaben konnten nur seriell und nicht parallel erfolgen
- Informationsaufgabe kann sich verschieben oder verschwinden durch folgenden Terminaloutput
- Eingabe von Informationen ggf. zeitintensiv
- Keine intuitive Eingabe, da Möglichkeiten nicht dargestellt sind

Das Nutzen von graphischen Benutzeroberflächen oder englisch Graphical User Interface (GUI), kann diese Nachteile beseitigen. Obwohl die Implementierung von GUIs für die einzelnen Subsysteme einen bedeutenden zusätzlichen Aufwand bedeutet, werden diese im Gegenzug höchstwahrscheinlich die folgenden positiven Effekte haben:

- Verbesserung der Fehlerfindung und Korrektur während der Entwicklung
- Reduktion der Betriebszeit während Tests
- Reduzierung der Zeit zur Ursachenforschung im Falle einer Anomalie
- Verbesserung des Verständnisses des individuellen Funktionsprinzips der Subsysteme.

Es wurden deshalb eine Vielzahl von Benutzeroberflächen entwickelt, um beim SpaceBot Cup 2015 eingesetzt zu werden. Diese sind im folgenden gelistet, die wichtigsten GUIs sind gesondert **markiert** und werden separat genauer beschrieben.

Das im Robot Operating System enthaltene Paket rviz wurde zusätzlich regelmäßig genutzt.

Entwickelte GUIs:

- **Motor Control**
- Pybash
- **Object calibration**
- Object recognition
- Housekeeping
- Task planning
- Simulation launcher
- Master
- Status Bar

2.1.2.2.1 Motor Control GUI Die grundlegende Idee der Motor GUI ist es, Anpassungen an jedem einzelnen Motors vornehmen zu können, vorzugsweise mit Hilfe eines Clicks. Es sind deshalb nur die wichtigsten Parameter im *Monitoring* Tab standardmäßig sichtbar. Zu diesen gehören z.B. die aktuelle Drehrate.

Zusätzliche Informationen und anpassbare Parameter werden in einem separaten Tab dargestellt, dem *Configuration* Tab. Nichtsdestotrotz ist das Senden und Empfangen von CANopen Nachrichten immer noch möglich, um weitere Informationen abrufen oder Befehle schicken zu können.

Wie in der Abbildung 2.11 ersichtlich, ist die Ausgabe zweigeteilt. Die Motorfläche ist auf der linken Seite und enthält die *Monitoring* und *Control* Tab. Die rechte Seite ist für die Kontrolle der Motoren über CANopen Nachrichten vorgesehen.

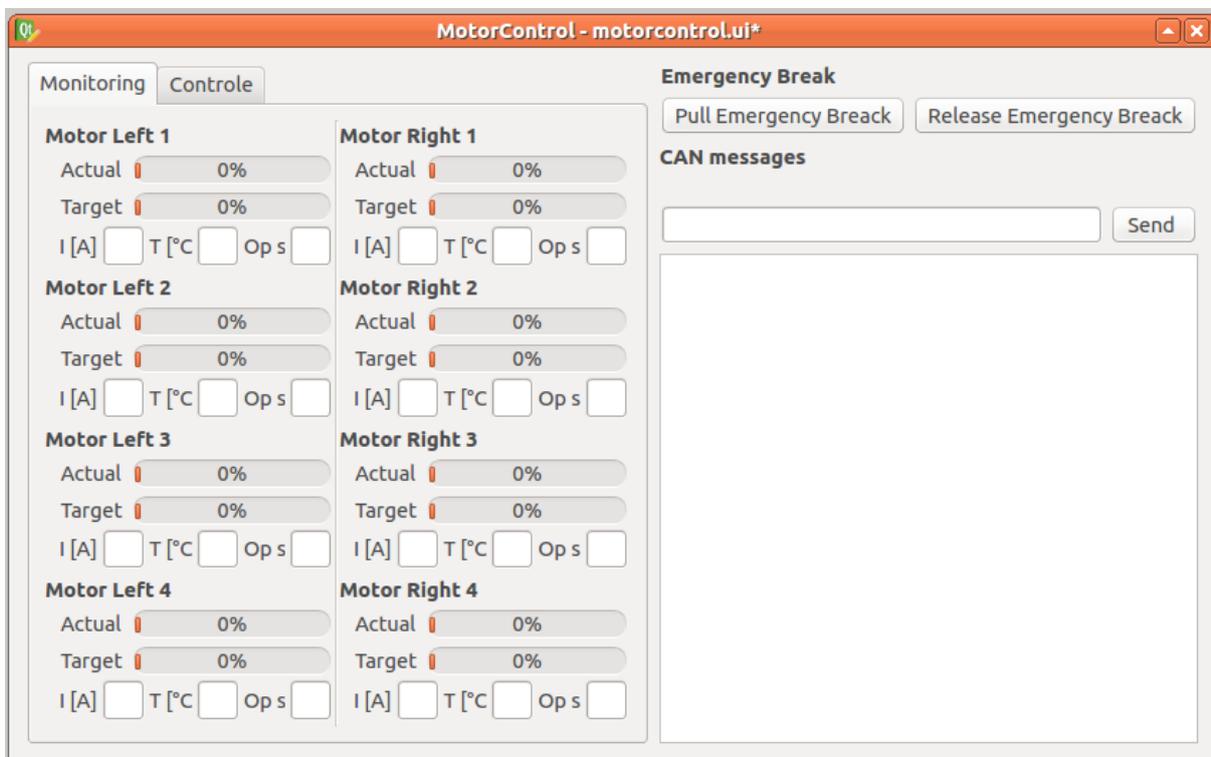


Abbildung 2.11: Motor GUI im Überblick

2.1.2.2.2 Monitoring Tab Jeder Motor ist symbolisiert durch eine Parameterbox, was zu einer Darstellung in acht Boxen führt. Die wichtigsten Parameter sind dargestellt mit:

- Start, stop, reset Buttons
- Balken um aktuelle und gezielte Drehzahl anzuzeigen
- Anzeige von elektrischem Strom I [A], Temperatur T [K] und Operationsstatus

2.1.2.2.3 Control Tab Der Kontrollreiter soll volle Kontrolle über die Motoren des Rover erlauben. Deshalb ist der Tab geteilt in zwei Abschnitte. Im ersten Abschnitt ist die Rover Kontrolle lokalisiert, Geschwindigkeit des Rover und Rotation der Räder können hier angepasst werden. Weiter unten findet sich die Single Motor Control Sektion, in der einzelne Motoren gezielt angesteuert werden können, zu sehen in Abbildung 2.12.

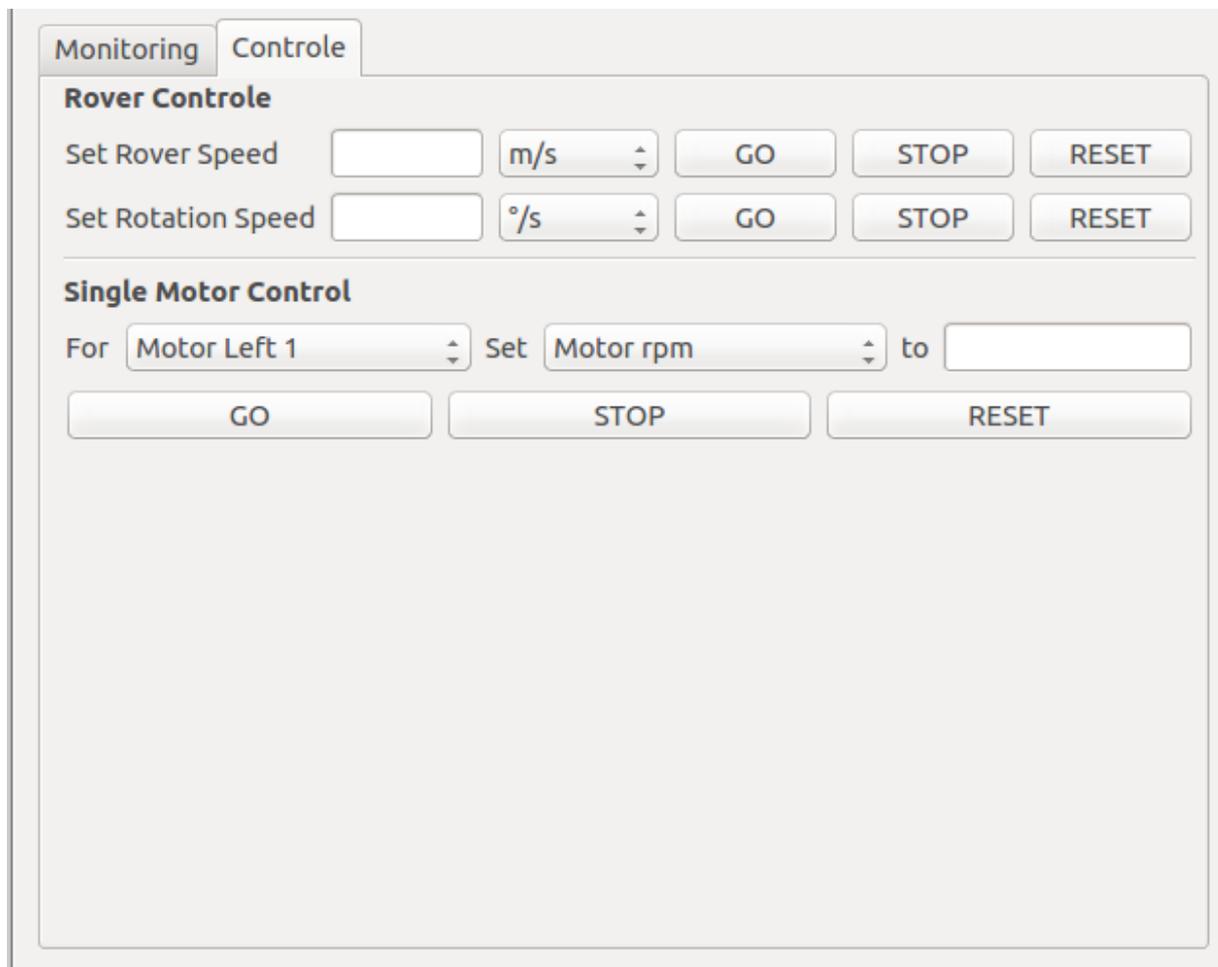


Abbildung 2.12: Engine-GUI overview

2.1.2.2.4 Emergency Break Der Emergency Break Button ist über dem Abschnitt für CANopen Nachrichten platziert und erlaubt dem Bodenpersonal jeder Zeit das Stoppen des Rover im Problemfall.

2.1.2.3 Objekt Kalibrierungs GUI

2.1.2.3.1 Objekt Erkennungs GUIs Die Objekt-Erkennung ist ein Subsystem, das SEAR ermöglichen soll, autonom Objekte in seiner Umgebung zu finden. Der Algorithmus dafür war bereits implementiert und getestet. Allerdings war noch eine Schnittstelle zu dessen Überwachung und Kontrolle notwendig. Außerdem war die präoperative Kalibrierung auf Lichtverhältnisse unhandlich und ungenau. Zwei Grafische Nutzer Schnittstellen wurden implementiert um diese Mängel zu beheben.

2.1.2.3.2 Objekt Erkennungs Kalibrierung Der Algorithmus zur Erkennung vordefinierter Objekte stützt sich hauptsächlich auf die Farbe des Objekts. Aufgrund dieser Abhängigkeit haben sich verändernde Beleuchtungs-Verhältnisse einen gravierenden Einfluss auf die Zuverlässigkeit des Algorithmus. Zur Anpassung an die aktuellen Lichtverhältnisse können Kalibrierungsdaten in JSON Dateien abgelegt werden. Die Kalibrierungs GUI ermöglicht ein simples, intuitives Erfassen dieser Daten. Die GUI integriert ein aktives *Kinect/Xtion* Kamera Bild mit einem eingeblendeten Fadenzentrum in der Mitte. Sobald ein Objekt fokussiert und eine Schaltfläche betätigt wird, speichert die GUI die extrahierte Farbinformation (Farbton, Sättigung, Helligkeit) und zeigt sie an. Es stehen außerdem Kontrollelemente zur manuellen Veränderung der Farbdaten zur Verfügung. Die GUI ist des Weiteren in der Lage JSON files abzuspeichern und ihren Inhalt wiederzugeben. Die Objekt-Eigenschaften werden angezeigt wie in Abbildung 2.15 dargestellt.

Objects	
Batterys	
Hue	12 - 26
Saturation	150 - 255
Brightness	140 - 255
Glas	
Hue	130 - 152
Saturation	145 - 255
Brightness	140 - 255
Base Station	
Hue	78 - 85
Saturation	150 - 255
Brightness	137 - 255

Abbildung 2.13: Darstellung der Objekteigenschaften

2.1.2.3.3 Object recognition operation control Das General User Interface kann ein live stream Video mit überlagerter Objekterkennung anzeigen (s. Abb.2.14)

Um eine Überwachung und Steuerung der Objekterkennung von der Bodenstation zu erlauben, ist eine Fernsteuerung und der Empfang von Telemetriedaten über ein ROS Paket möglich. Als weiteres Vorgehen ist eine Auswertung der GUIs geplant, um eventuelle Verbesserungspunkte zu identifizieren.

Das General User Interface für die Fernsteuerung der Objekterkennung beinhaltet folgende Funktionen:

- Eine Wahlleiste, um zwischen verschiedenen Standard Einstellungen wechseln zu können. (Oben links)
- Eine Wahlleiste, um zwischen verschiedenen Bildquellen zu wählen. (Oben rechts)
- Eine Darstellung der einzelnen Objektkalibrationen. (mittig links)

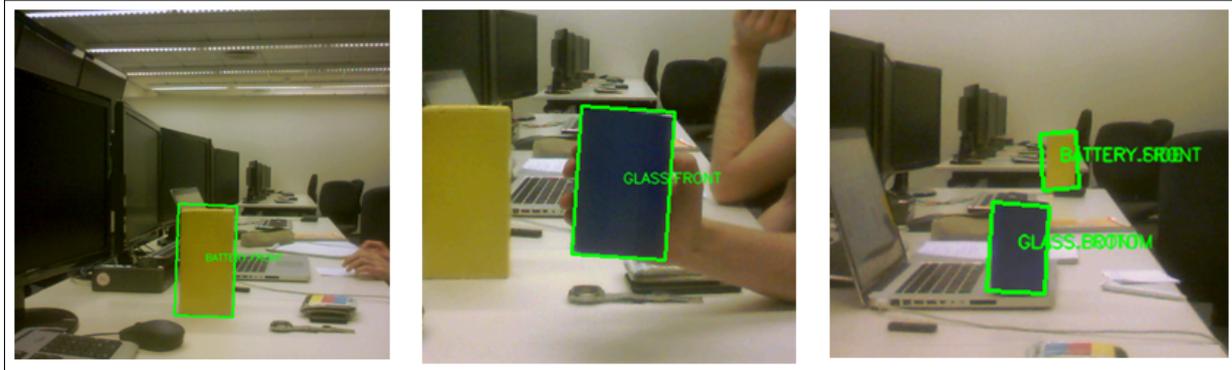


Abbildung 2.14: Video feeds mit der Objekterkennung

- Ein beweglicher und größenverstellbarer Kreis, zur direkten Farbenerkennung aus dem Live Stream.
- Ein Live Stream der Kamera mit überlagelter Objekterkennung, s. Abb.2.14 (rechte Seite)
- Eingabelemente für individuelle Betriebseinstellungen, wie z. B.:
 - Farbton,
 - Sättigung,
 - Helligkeit.

2.1.2.4 Objekterkennung

Im Wettbewerbsszenario des SpacebotCup sollte der Rover selbstständig Objekte, mit bestimmten Charakteristiken, in seiner unmittelbaren Umgebung erkennen.

Die Wahl der Sensoren und der verwendeten Algorithmen waren den Teams freigestellt. Beim ersten SpacebotCup, 2013, wurden insgesamt zwei Microsoft Kinect Kameras verwendet um die Objekterkennung zu betreiben. Die Idee dahinter war es, mit einer Kamera sowohl RGB, also auch Tiefeninformationen der Umgebung zu erhalten. Zum SpaceBot Camp, 2015, wurden die beiden Kinect Kameras durch insgesamt 6 Asus Xtion Kameras ersetzt. Damit wurde es möglich ein viel größeren Ausschnitt aus der Umgebung des Rover zu beobachten und auszuwerten. Darüber hinaus lösen die Xtion Kameras höher auf und weisen insgesamt eine bessere Bildqualität auf.

2.1.2.4.1 Algorithmen Zu Beginn der Vorbereitungsphase auf das SpaceBotCamp, 2015, existierte bereits ein rudimentärer Algorithmus, der auf der klassischen *Blob-Detection* basierte. Dieser war durch die rein funktionale Programmierung nur schwer zu warten und zu erweitern. In wenigen Wochen wurde der gesamte Code für die *Blob-Detection* vollständig in objektorientierter Manier neu geschrieben und mit einer Vielzahl von zusätzlichen Features verbessert.

Grundsätzlich basiert die Erkennung von Objekten durch die *Blob-Detection* auf deren Farbe (Abbildung 2.16). Dazu wird ein bestimmter Farbbereich im HSV-Farbraum festgelegt, in dem das Objekt erwartet wird. Wobei die wahrgenommene Farbe eines Objekts durch die Lichtbedingungen mitunter stark variieren kann. Das ist auch eine der größten Schwächen der *Blob-Detection*: Schwache

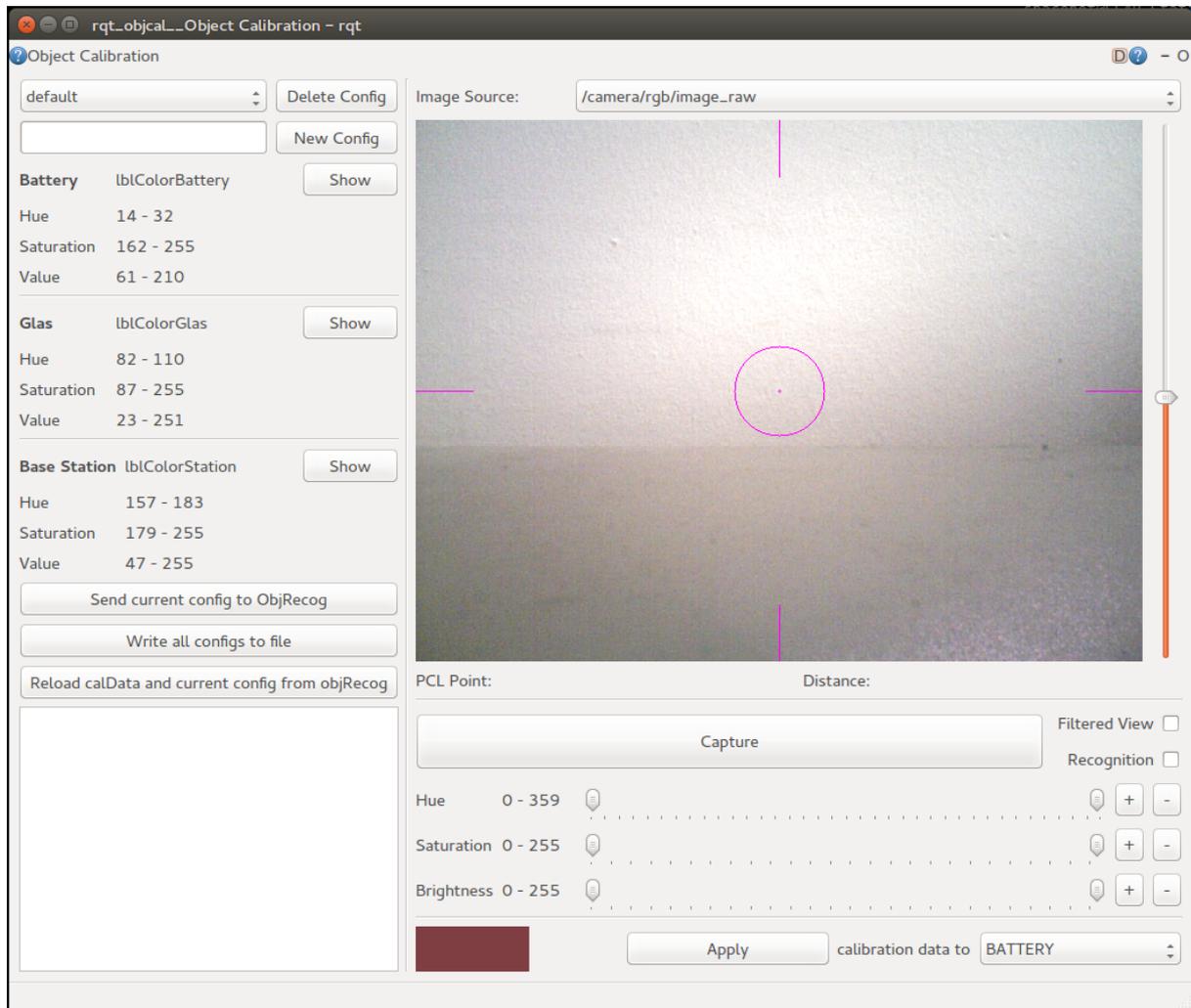


Abbildung 2.15: Object calibration GUI

Beleuchtung oder Beleuchtung durch nicht weiße Quellen, können die scheinbare Farbe von Objekten mitunter so stark verändern, das nichts erkannt wird. Ein weiterer Nachteil ist die Tatsache, dass sich aus einem erkanntem *Blob* keine Aussage über die Lage des Objekts machen lässt. Um das zu ermöglichen werden die Tiefenbilder herangezogen, die die Xtion Kameras gleichzeitig mit den RGB-Bildern liefern. Die Tiefenbilder werden zunächst in 3D-Punktwolken konvertiert und aus diesen wird dann näherungsweise versucht, die Lage von den mit der Blob-Detection erkannten Objekte zu bestimmen. Zwischen der Qualifikation und dem SpacebotCamp, 2015, als das Team erkannt hat, dass die *Blob-Detection* zu viele False-Positives erzeugt hat, wurde ein weiterer Ansatz zur Erkennung von Objekten hinzugezogen. Ein sogenannter *Cascade Classifier* der Objekte durch *Haar-like* features klassifiziert. *Haar-like* features sind sehr simple Anordnungen von Bereichen aus hell und dunkel, die einem Bild zugeordnet werden können. Ein Classifier muss zunächst mit einer Vielzahl von Bildern auf denen das Zielobjekt ist trainiert werden. Wichtig dabei ist, dass der Hintergrund bzw. die Umgebung des Zielobjektes auf den Trainingsbildern möglichst variiert. Nur so kann ein Zielobjekt später unabhängig vom Hintergrund erkannt werden. Über die Anzahl der Kaskaden, mit denen der *Classifier* auf ein Zielobjekt trainiert wird lässt sich der Grad zwischen zu vielen False-Positives und zu vielen False-Negatives einstellen.

In Verbindung mit der *Blob-Detection*, die dazu neigt, viele False-Positives zu erkennen, ist es ratsam

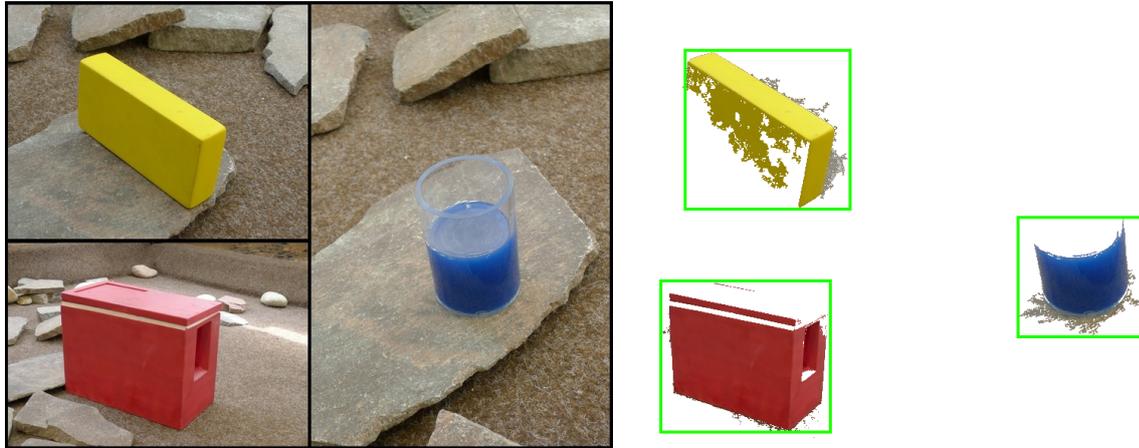


Abbildung 2.16: Zielobjekte im Original und nach Filterung des jeweiligen Farbbereichs

auch den *Cascade Classifier* so zu trainieren, dass eher zu viel als zu wenig erkannt wird. Denn die Tatsache, dass die Algorithmen, Objekte durch völlig unterschiedliche Kriterien erkennen, macht es sehr wahrscheinlich, dass sie auch völlig unterschiedliche False-Positives erzeugen. Das macht man sich zu nutze, indem man die Schnittmenge aus den Kandidaten der beiden Algorithmen bildet. Es wird also nur dann ein Objekt als erkannt gemeldet, wenn beide Algorithmen einen Kandidaten an der gleichen Position im Bild gefunden haben. Der Ansatz beide Algorithmen zu kombinieren, hat sich als deutlich überlegen gegenüber der einfachen Blob-Detection herausgestellt. Der Nachteil ist, dass für jedes Zielobjekt eine Vielzahl von Bildern aufgenommen werden muss, auf denen das Objekt in allen möglichen Lagen abgebildet ist. Darüber hinaus kann der Trainingsprozess viele Stunden in Anspruch nehmen.

2.1.2.4.2 Applikationen zum Kalibrieren und Testen

Für die Blob-Detection wurde eigens eine Applikation mit Benutzeroberfläche entwickelt, die es ermöglicht während des Betriebs eine Kamera des Rovers als Bildquelle zu wählen um dann die Kalibration für ein Objekt im Bild vorzunehmen. Die Applikation ermöglicht es auch, direkt am Livebild zu sehen, ob der Farbbereich in dem nach einem Objekt gesucht wird eventuell zu groß oder zu klein eingestellt ist. Darüber hinaus kann der Erkennungsalgorithmus zum Nachkalibrieren auch innerhalb dieser Applikation angewendet werden. Eine weitere Anwendung wurde für den Cascade Classifier geschrieben. Auch hier war das Ziel der Anwendung eine möglichst einfache und schnelle Einstellung der relevanten Algorithmusparameter vornehmen zu können.

2.1.2.4.3 Ausblick

Derzeit wird der Nutzen und die Umsetzbarkeit eines weiteren Ansatzes untersucht. Dabei handelt es sich um die Erkennung von Objekten mithilfe von neuronalen Netzen. Damit soll die Zuverlässigkeit der Erkennung weiter gesteigert werden. Es bleibt abzuwarten, ob diese Methode in Kombination mit den bereits eingesetzten zum Einsatz kommen soll oder ob sie, sofern sie tauglich ist, die anderen Methoden vollständig ersetzen kann. Ein vollständiger Ersatz würde den Ressourcenbedarf im Betrieb wohlmöglich deutlich reduzieren.

2.1.2.5 SLAM - Simultaneous Localization and Mapping

2.1.2.5.1 Ansatz

Zur Lösung des Problems der gleichzeitigen Kartenerstellung und Lokalisierung wurde für SEAR ein SLAM Ansatz verwendet. Dieser basiert im Gegensatz zu den häufigsten Umsetzungen nicht auf den Daten eines Laserscanners sondern auf denen von mehreren ASUS Xtion Tiefenkameras. Die Implementierung basiert auf dem ROS-Paket "RGBDSLAMv2". Das SEAR System verzichtet vollständig auf herkömmliche Odometrie (Sensordaten seines Vortriebsystems). Stattdessen werden visuelle, wiedererkennbare Merkmale verwendet um die Eigenbewegung zu schätzen.

2.1.2.5.2 Visuelle Odometrie

Dazu berechnet der Algorithmus zuerst in den RGB-Bildern visuelle Merkmale nach dem SIFT-Verfahren. Ähnliche Merkmale in aufeinanderfolgenden Frames werden gematcht, wobei zur Kontrolle des Verhältnisses zwischen false positives und -negatives ein Schwellenwert für das Verhältnis zwischen dem ähnlichsten und zweit-ähnlichsten Merkmal angesetzt wird. Aus den gematchten dreidimensionalen Merkmals-Vektoren wird eine Transformation berechnet. Dazu wird, aus Gründen der Robustheit gegen false positives, das RANSAC Verfahren verwendet.



Abbildung 2.17: SIFT-feature matches (RANSAC inliers)

Die geschätzten Positionen, die dreidimensionalen Punktwolken und die Merkmals-Vektoren werden in einem Graphen gespeichert. Die SIFT-Merkmale neuer Frames werden nicht nur mit den direkten Vorgängerknoten verglichen, sondern auch mit weiter zurückliegenden, um "loop closures" zu ermöglichen. Als "loop closures" bezeichnet man das erkennen von bereits erkundeten Gebieten und das dadurch mögliche, rückwirkende Korrigieren akkumulierter Fehler im Graphen. Ein Optimierungs-Verfahren durchläuft regelmäßig den Graphen und minimiert dabei eine Fehlerfunktion, um eine optimale, über den gesamten Graphen kohärente, Trajektorie zu erhalten. Die Punktwolken wurden während des Betriebs in ein speicher-effektives Voxelgrid (Octomap) eingetragen.

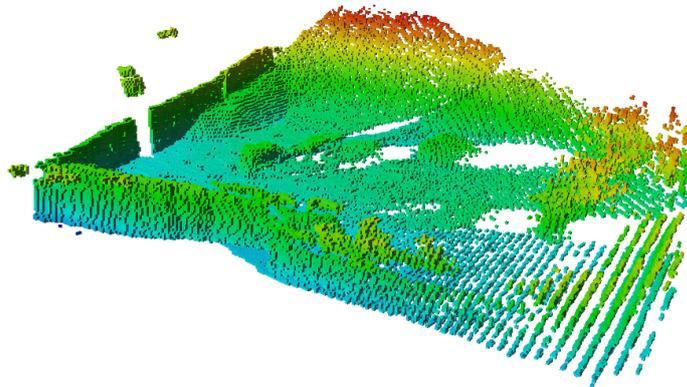


Abbildung 2.18: Generierte Karte

2.1.2.6 Manipulatorsteuerung

Der robotische Manipulator von SEAR, der sowohl in der Qualifikation zum SpaceBot-Cup 2015 als auch während des SpaceBot-Camps 2015 zum Einsatz kam, war ein "Advanced Robotic Manipulator 2.0" der Firma GearWurx (vorher Incenscience LC). Ausgestattet mit fünf Gelenken und einem Greifer - und dementsprechend sechs Freiheitsgraden - wurde der Arm vielfach modifiziert um ihn für die Benutzung auf dem Rover anzupassen.

Gelenk	Bewegungsbereich
Basisdrehgelenk	250°
Schulter	135°
Ellenbogen	135°
Handgelenk	50°
Handgelenk Drehung	270°

Tabelle 2.1: Gelenke des Manipulators

Wie im gesamten Projekt wurde auch zur Steuerung des Manipulators das *Robot Operating System* verwendet. In ROS lässt sich die weit verbreitete Open-Source Software *Movel!* integrieren und nutzen. Durch die Arbeit mit *Movel!* steht einem eine stark communitygestützte Plattform für viele Teilbereiche der Robotik, auf die es im Greifprozess ankommt, zur Verfügung. So kann man z.B. auf Konzepte in Bewegungsplanung (*motion planning*), Kinematik, Steuerung und Navigation zurückgreifen.

Zur Anbindung an das gesamte Softwaresystem des Rovers wurde für die Kontrolle und Steuerung des Manipulators ein eigenes ROS-Paket erstellt. Mehrere ROS-Nodes in diesem Paket ermöglichen durch die Nutzung von *Movel!* die Ansteuerung des robotischen Arms. Die in Tabelle 2.1 beschriebenen Gelenke werden über den im ARM 2.0 integrierten *Micro Maestro 6-Channel USB Servo Controller* der Firma *Pololu* angesteuert. Dafür ist es notwendig die Befehle als Pulsweitenmodulation-

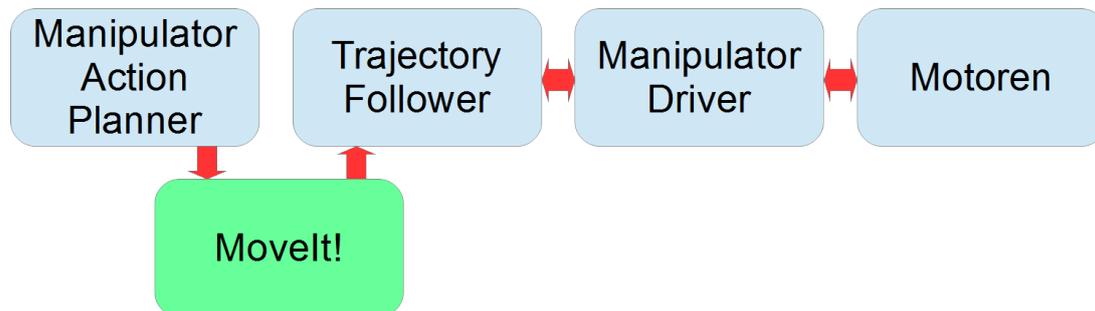


Abbildung 2.19: ROS Paket, Manipulator Control

Signale (*pulse-width modulation*, PWM) an die Motoren zu senden. Die Schnittstelle zum Pololu-Board wird durch den *Manipulator-Driver*-Node bereitgestellt, der ebenso die Umrechnung einer Drehung eines Gelenks um einen bestimmten Winkel in ein PWM-Signal übernimmt. Die Gleichungen um die Servo-Ansteuerung zu ermitteln sind im folgenden ersichtlich. Je nachdem, ob der Motor eine 'umgekehrte' Bewegungsrichtung verfolgt oder nicht werden zwei verschiedene Berechnungen benötigt. Gleichung 2.1 wird bei umgekehrter Bewegungsrichtung verwendet und Gleichung 2.2 bei regulärer Bewegungsrichtung.

$$PWM = \frac{(jointLimit[MAX] - angle) \cdot (pwmLimit[MIN] - pwmLimit[MAX])}{jointLimit[MIN] - jointLimit[MAX] + pwmLimit[MIN]} \quad (2.1)$$

$$PWM = \frac{(jointLimit[MAX] - angle) \cdot (pwmLimit[MAX] - pwmLimit[MIN])}{jointLimit[MIN] - jointLimit[MAX] + pwmLimit[MAX]} \quad (2.2)$$

Damit für eine gewollte Bewegung des Arms die entsprechende inverse Kinematik von MoveIt! berechnet werden kann, benötigt das Programm ein digitales Modell des Rovers und besonders vom Manipulator. Um dieses zu erzeugen wird das *Unified Robot Description Format* (URDF) benutzt, das mit der Struktur der *Extensible Markup Language* (XML) eine Roboterkonfiguration in ihren Links und Joints hinsichtlich ihrer Dimensionen, Positionen und Bewegungsmöglichkeiten korrekt beschreibt. Ein bei MoveIt! mitgeliefertes Tool (MoveIt! Setup Assistant) generiert aus diesem URDF-Modell ein Roboter-Modell, welches als eigenständiges ROS-Paket agiert und so von MoveIt! und der Rover-Software im Allgemeinen genutzt werden kann. Das Paket enthält neben der in der URDF-Datei beschriebenen Konfiguration noch weitere Funktionalitäten wie z.B. Angaben über Links, die nie miteinander kollidieren und so von der Kollisionserkennung ausgeschlossen werden können, und bestimmte festgelegte Posen des Roboters.

Für die direkte Kommunikation und Nutzung der angebotenen Funktionalitäten von MoveIt! ist der *Manipulator-Action-Planner*-Node im Manipulator-Paket implementiert worden. Die im Folgenden aufgelisteten Methoden bieten die für den Greifprozess elementaren Vorgehensweisen an. Dabei wird je nach Methode noch zwischen den beiden Objekten - dem gelben Batterie-Paket und dem blauen Becher - unterschieden.

- Autonomes Greifen mit Hilfe der Objekterkennung
- Autonomes Greifen mit GripperCam und Infrarot-Sensor
- Manuelles Greifen
- Platzieren des Objektes im jeweiligen Objektbehälter

– Greifen des Objektes aus dem jeweiligen Objektbehälter

Das autonome Greifen mit Hilfe der Objekterkennung nutzt das Motion Planning von MoveIt! um unter Kollisionsvermeidung den vorher definierten *End Effector* (in dem Falle der Greifer) - bzw. sein Koordinatensystem - auf das Koordinatensystem des Objektes auszurichten. Beide Koordinatensysteme beziehen sich auf dasselbe Referenzkoordinatensystem. Die Lage des Objektes wird von einem anderen Teilsystem des Rovers ermittelt - der Objekterkennung mit Hilfe der Kameras. Geplant wird dementsprechend eine Trajektorie um das System des Greifers in einem bestimmten Referenzsystem zu der Lage des Systems des Objektes zu bewegen. Der Arm durchlief dabei verschiedene Zustände. Zuerst wurde eine Trajektorie für die sogenannte *PreGraspPose*, die sich 20 cm vor dem Objekt befand, geplant und ausgeführt. Der nächste Schritt war das Öffnen des Greifers um anschließend den End Effector in die Position und Lage des Objekts (*GraspPose*) zu führen. Der Greifer wurde geschlossen und der Arm anschließend auf die *PostGraspPose* 20 cm über der ursprünglichen Position des Objekts bewegt. In der Ausführung erwies sich diese Strategie jedoch nicht als praktikabel, da die Lage des Objektes nicht hinreichend korrekt bestimmt werden konnte um ein Greifen zu ermöglichen.

Die zweite Variante des autonomen Greifens nutzte dabei die durch die Objekterkennung vorhergesagte Position des Objektes als erste Orientierung. Der Arm wird in festem Abstand vor den Rover auf die von der Objekterkennung ermittelte Höhe (über dem Boden) des Objektes bewegt. Anschließend bewegt sich das Handgelenk mit seinem Drehgelenk 90° nach links und danach 180° nach rechts. Dabei scannt der im Greifer integrierte Infrarot-Sensor die Umgebung und überprüft dank der eingebauten Webcam, ob es sich um das zu greifende Objekt handelt, welches in einer bestimmten Entfernung gemessen wurde. Wurde das Objekt erfolgreich ausgemacht so wird der Arm in seitlicher Richtung über das Basisdrehgelenk auf die entsprechende Position gebracht. Nun folgt das Durchlaufen einer Schleife, um sich dem Objekt anzunähern. Dabei wird auch konsequent der sich möglicherweise verändernde seitliche Abstand korrigiert. Misst der Infrarotsensor eine Entfernung unter 6cm, so schließt sich der Greifer und das Objekt wird gegriffen. Viele Feldversuche haben ergeben, dass je nach Orientierung des Objektes der Griff mal mehr und mal weniger sicher und geeignet für das weitere Vorgehen war, sodass eine weitere Maßnahme implementiert wurde. Anschließend wird das Objekt direkt vor dem Rover noch einmal auf dem Boden abgestellt. Durch das gleichmäßige Öffnen des Greifers ist es nun ebenso wie der Rover orientiert und kann sicher gegriffen werden.

Das erfolgreich praktizierte manuelle Greifen beruht auf der Möglichkeit durch eine von MoveIt! geplante Trajektorie den Greifer zu einer gewünschten Position zu bewegen. Nach einer ersten Ausführung dieser Bewegung sollte sich der Greifer knapp über dem Boden befinden, sodass der komplette Rover mit Hilfe der Webcam im Greifer manuell zum Objekt manövriert werden kann. Der Greifer wird bei erfolgreicher Annäherung manuell geschlossen. Optional stünde eine manuelle Auswertung der Daten des Infrarot-Sensors zur Verfügung, um die Annäherung nicht nur mit den Kameras beurteilen zu müssen.

Zum Platzieren der sicher gegriffenen Gegenstände gibt es je Objekt eine fest einprogrammierte Abfolge von Bewegungen der Gelenke. Da die Objektbehälter fest am Roverkorpus angebracht sind, können die Batterie und der Becher auf diese Art und Weise sicher platziert werden.

Da durch die Objektbehälter die Position der darin platzierten Objekte fest vorgegeben ist, kann das Herausnehmen in der gleichen Art und Weise geschehen wie das Platzieren. Eine feste Abfolge von bestimmten Bewegungen der Gelenke ist einprogrammiert worden, so dass die Batterie und der Becher entsprechend entnommen werden können.

Ein abschließendes Platzieren der Objekte erfolgt anschließend wieder manuell. Um eine grobe Position anzusteuern, kann mit MoveIt! eine Trajektorie zu selbiger geplant werden um dann gewisse Feinheiten einerseits entweder mit dem Fahrwerk des Rovers auszugleichen oder andererseits einen *ROS-Service* zu benutzen, um ein bestimmtes Gelenk in seiner Stellung um eine gewisse Gradzahl zu bewegen.

2.1.2.7 Kommunikation

Im Rahmen des Wettbewerbs waren das Boden- und das Raumsegment durch eine kabelgebundene Ethernetverbindung mit einer Übertragungsrate von bis zu 100 MBit/s miteinander verbunden. Innerhalb des Raumsegments erfolgte die Datenübertragung kabellos über ein drahtloses Netzwerk. Zwischen der Bodenstation und dem Raumsegment war ein Netzwerkemulator angesiedelt, mit dem eine Latenz von zwei Sekunden pro Strecke eingebracht wurde.

2.1.2.7.1 Probleme im SpaceBot Cup 2013 Für die reibungslose Kommunikation im zweiten SpaceBot Cup wurden die Probleme von 2013 analysiert und behoben. Im folgenden werden die entscheidenden Einschränkungen erklärt und deren Lösungen im Anschluss vorgestellt.

- Störanfälligkeit der Funkstrecke im 2,4 GHz WLAN Frequenzband: Die zur Verfügung stehende Datenrate im Downlink konnte nicht optimal ausgenutzt werden, da die Qualität des Signals zu gering war.
- Neuübertragungen von Datenpaketen: Im SpaceBot Cup 2013 kam ein Transportprotokoll zum Einsatz, welches verloren gegangene Daten selbstständig neu anfordert. Aufgrund der schlechten Qualität der Funkstrecke, mussten viele Pakete bei hoher Latenz neu übertragen werden und die Datenrate brach ein.
- Aufstauen von Telemetriedaten im Raumsegment wegen zu geringer Datenrate im Downlink: Durch den starken Einbruch der Übertragungsrate konnte kein durchgängiger Telemetriedatenstrom sichergestellt werden. Dadurch stauten sich auf der Seite des Rovers Daten für den Downlink an. Die Folge war das stark verspätete Eintreffen von Zustandsinformationen des Rovers in der Bodenstation.

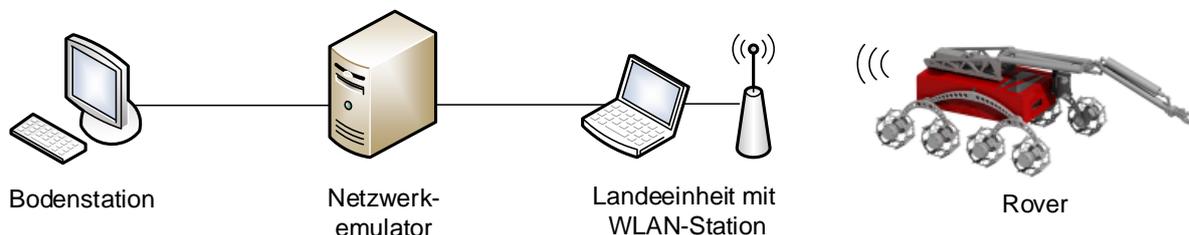


Abbildung 2.20: Aufbau der Kommunikationsstrecke

2.1.2.7.2 Aufbau Im Vergleich zum SpaceBot Cup 2013 wurde die Kommunikationsstrecke im Aufbau und im verwendeten Transportprotokoll für die Datenübertragung verbessert. Auf der Seite des Raumsegments wurde am Startpunkt eine Landeeinheit abgestellt, siehe Abbildung 2.20. Sie ist direkt mit dem kabelgebundenen Ethernet verbunden und bildet den WLAN Zugangspunkt für den Rover. Damit wurde die latenzbehaftete Kommunikationsstrecke direkt am Landepunkt terminiert.

Neuübertragungen von Datenpaketen auf der potentiell gestörten WLAN-Verbindung konnten so mit geringer Verzögerung wiederholt werden. Zusätzlich wurde das Frequenzband der Funkübertragung von 2.4 GHz auf 5 GHz gewechselt, um externe Störungen zu minimieren. Der dabei verwendete WLAN-Standard 802.11n unterstützt eine Datenrate von bis zu 150 MBit/s. Die WLAN-Antennen wurden für eine bessere Abstrahlcharakteristik auf dem neuen Kameramast angebracht. Auf diese Weise konnte innerhalb des Raumsegments eine stabile und robuste Funkverbindung aufgebaut werden.

2.1.2.7.3 Transportprotokoll Im Rahmen dieses Vorhabens wurde ein neues Übertragungsprotokoll für die Telemetrestrecke auf Basis des User Datagram Protocol (UDP) entwickelt. Das Prinzip des Datentransports bleibt bestehen: Nachrichten aus dem Robot Operating System (ROS) werden in einen seriellen Datenstrom zerlegt, zur Bodenstation übertragen, dort zusammen gesetzt und im Telemetriesystem verteilt. Eine schematische Darstellung der Datenübertragung ist in Abbildung 2.21 zu sehen.

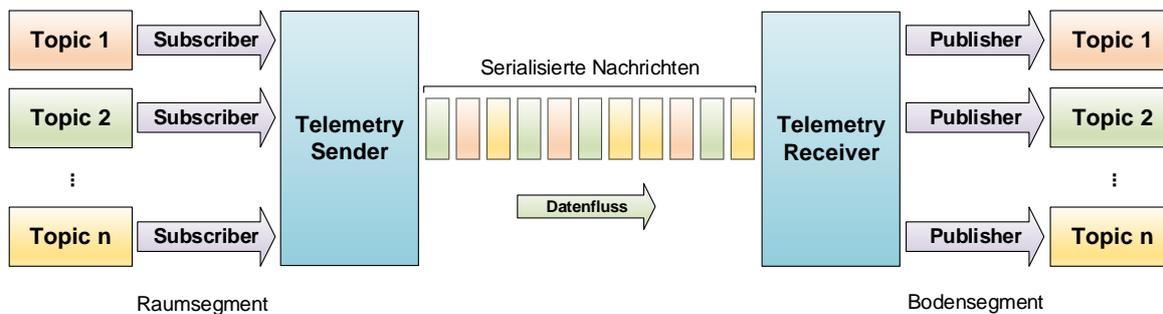


Abbildung 2.21: Prinzip der Datenübertragung

Die beiden Endpunkte dieser Verbindung arbeiten in der Bodenstation und auf der Landeeinheit. Muss hierbei ein Datenpaket neu übertragen werden, verzögert das bei der gegebenen hohen Latenz die Übertragung kompletter ROS Nachrichten.

Aufbauend auf den Erfahrungen des ersten SpaceBot Cups, stellte sich heraus, dass ein Protokoll ausreichend ist, welches folgende Eigenschaften erfüllt:

- Keine Neuübertragungen bei Paketverlusten
- Detektion von Paketverlusten
- Es erfolgen keine Bestätigungen für Datenpakete
- Auf eine Datenflusskontrolle wird verzichtet
- Auf eine Überlaststeuerung wird verzichtet

Im Einsatz mit dem neuen Übertragungsprotokoll zwischen Bodenstation und Landeeinheit wurde deutlich, dass auf der Ethernetverbindung keine Paketverluste auftraten. Der Grund liegt in der geringeren, tatsächlich benötigten Datenrate von etwa 16 MBit/s. So wurde im Downstream das verwendete Netzwerk mit einer maximalen Übertragungsrate von 100 MBit/s zu keiner Zeit überlastet.



Abschlussbericht

Zudem eignete sich das neue Protokoll viel besser für den Einsatz unter hoher Latenz, da in keiner Weise ein Rückkanal auf dem Übertragungsweg genutzt wurde. Die Telemetrie wurde sofort vom Sendeknoten (*Telemetry Sender*, siehe Abbildung 2.21) an die Bodenstation versendet. Dabei wurde nicht auf Bestätigungen gewartet oder auf Überlast geprüft. Das hier verwendete Verhaltensmuster ist am besten mit „fire and forget“ zu bezeichnen.

2.2 Nutzen und Verwendbarkeit der Ergebnisse

Eine direkte wirtschaftliche Verwertung ist nicht abzusehen. In diesem Vorhaben lag aus Verwertungssicht das Ziel, am Wettbewerb teilzunehmen und das bestehende robotische weiterzuentwickeln. Der DLR SpaceBot Cup stellt hierbei eine terrestrische Analogumgebung dar, die sowohl vom operationellen Szenario her als auch durch die Umgebungsbedingungen viele Aspekte der möglichen Zielgebiete von planetaren Explorationsmissionen darstellt. Dies lässt die Ergebnisse dieses Vorhabens zu einem wertvollen Beitrag für das Potential Deutschlands zur Teilnahme an solchen Missionen werden.

Das Vorhaben hat die in 2013 geschaffene Wissensgrundlage auf dem Bereich der Robotik am Fachgebiet Raumfahrttechnik der TU Berlin weiter ausgebaut und zu der Planung weiterer Aktivitäten auf diesem Gebiet geführt. So wird derzeit eine Designstudie in dem angebotenen Modul "Planetare Exploration und Weltraumrobotik" durchgeführt, bei welchem ein Rover entworfen wird, der dann für eine erfolgreiche Teilnahme an der European Rover Challenge (<http://roverchallenge.eu/en/>) geeignet sein soll.

Da sich die Aufgaben und Anforderungen des Wettbewerbs auch teilweise mit Bedürfnissen anderer Einsatzbereiche decken, liegt hier ein großes Transferpotential vor. Zu nennen sind hier zum Beispiel Logistik, Search & Rescue Szenarien und Anwendungen im Straßenverkehr. Besonders die Entwicklungen auf dem Gebiet der Autonomie sollte starke Impulse liefern können.

2.3 Fortschritte auf dem Gebiet des Vorhabens bei anderen Stellen

Mit der Entwicklung des Robotersystems und der Teilnahme am Wettbewerb wurde die TU Berlin in die Lage versetzt die Forschungsarbeiten im Feld stark voranzutreiben. Einen großen Beitrag hat das Projekt für die praktische Ausbildung geleistet. Die 2013 eingeführte Projektwerkstatt, in welcher Studenten bei den Arbeiten am System betreut werden, wurde während der Projektphase fortgeführt und soll auch in Zukunft eine wichtige Rolle in der Lehre des Fachgebiets Raumfahrttechnik spielen. Arbeiten um den Rover werden auch weiterhin in Lehrveranstaltungen als praktische Einheiten abgehalten und durch die Projektwerkstatt betreut.

2.4 Veröffentlichungen

L. Kryza, S. Kapitola, C. Avsar, K. Brieß (2015). DEVELOPING TECHNOLOGIES FOR SPACE ON A TERRESTRIAL SYSTEM: A COST EFFECTIVE APPROACH FOR PLANETARY ROBOTICS RESEARCH. 1st Symposium on Space Educational Activities. Padova (Italy), 2015.

Die Aktivitäten um das Projekt SEAR haben großen Anklang in der Öffentlichkeit gefunden. Unter www.sear-blog.de und einer SEAR Facebook-Seite wurden aktuelle Meldungen veröffentlicht. Im Rahmen des Projekts erfolgten zahlreiche Veröffentlichungen in diversen Medien.

Titel	Medium	Datum
Small Exploration Assistant Rover	Vortrag (Tage der Raumfahrt 2015)	15.11.2015
Developing Technologies for Space on a Terrestrial System: A Cost Effective Approach for Planetary Robotics Research	Vortrag (1st Symposium on Space Educational Activities)	10.12.2015
SEAR - Ein studentischer planetarer Rover	Poster (Tag der Lehre)	18.01.2016

Tabelle 2.2: Projektbezogene Veröffentlichungen



Literaturverzeichnis

- [Arl12] ARLT, Dipl. Ing. F.: *Missionskonzept zur autonomen Marserkundung mit Hilfe verteilter Roversysteme*. Diplomarbeit
- [Fle11] FLEISCHER, B. Sc. P.: *Konstruktion eines passiven Fahrwerks für ein miniaturisiertes extraplanetares Rover-System*. Bachelorarbeit
- [Kö12] KÖHLER, Dipl. Ing. K.: *Entwicklung eines Rad-Nabe-Motor-Systems für einen Micro-Rover*. Diplomarbeit
- [Nip11] NIPPE, Anton: *Analyse ausgewählter Fahrwerke für einen Micro Rover mit Hilfe von SimMechanics*. Studienarbeit
- [Son12] SONSALLA, Dipl. Ing R.: *Konstruktion und Auslegung eines Manipulatorarms für den Einsatz im Bereich der planetaren Exploration*. Diplomarbeit

Berichtsblatt

1. ISBN oder ISSN -	2. Berichtsart Schlussbericht
3. Titel Weiterentwicklung des Rovers SEAR (Small Exploration Assistant Rover) für die Teilnahme am DLR SpaceBot Cup 2015	
4. Autor(en) Brieß, Klaus, Kryza, Lennart, Meuser, Marcus, Sändig, Bertram, Kapitola, Sascha, Wüstenberg, Philipp, Tzschacksch, Felix, Heeg, Francesca, Rückel, Tim	5. Abschlussdatum des Vorhabens 30. November 2015
	6. Veröffentlichungsdatum 27.Mai 2016
	7. Form der Publikation
8. Durchführende Institution(en) Technische Universität Berlin Institut für Luft- und Raumfahrt Fachgebiet Raumfahrttechnik Marchstr. 12-14 10587 Berlin	9. Ber. Nr. Durchführende Institution SEAR-AB-2015
	10. Förderkennzeichen 50 RA 1420
	11. Seitenzahl 41
12. Fördernde Institution Raumfahrtzentrum des Deutschen Zentrum für Luft- und Raumfahrt e.V. Königswinterer Str. 522-524 53227 Bonn	13. Literaturangaben 5
	14. Tabellen 4
	15. Abbildungen 24
16. Zusätzliche Angaben	
17. Vorgelegt bei	
18. Kurzfassung In diesem Vorhaben wurde der planetare Rover SEAR für die Teilnahme am SpaceBot Cup 2015 weiterentwickelt. Im Bericht werden die Ergebnisse der Arbeiten des Vorhabens präsentiert, zu diesen gehören die Entwicklung eines neuen Roboterarms, eines neuen Korpus für den Rover, sowie eines die Erweiterung des visuellen Sensorsystems durch spezielle Kameras. Auch die Weiterentwicklung der Software, insbesondere in Hinblick auf Autonomie und Entscheidungsfähigkeit des Rovers und die Operation eines solchen Raumfahrtsystems unter Missionsbedingungen werden dargestellt.	
19. Schlagwörter Raumfahrt, Raumfahrttechnik, Robotik, Autonomie, Planetare Exploration, Künstliche Intelligenz	
20. Verlag -	21. Preis -

Document Control Sheet

1. ISBN or ISSN	2. type of document report	
3. title Weiterentwicklung des Rovers SEAR (Small Exploration Assistant Rover) für die Teilnahme am DLR SpaceBot Cup 2015		
4. author(s) Brieß, Klaus, Kryza, Lennart, Meuser, Marcus, Sändig, Bertram, Kapitola, Sascha, Wüstenberg, Philipp, Tzschacksch, Felix, Heeg, Francesca, Rückel, Tim		5. end of project November 30th, 2015
		6. publication date May 27th, 2016
		7. form of publication
8. performing organization(s) Technische Universität Berlin Department of Aeronautics and Astronautics Chair of Space Technology Marchstr. 12-14 10587 Berlin		9. originator's report no. SEAR-AB-2015
		10. reference no. 50 RA 1420
		11. no. of pages 41
12. sponsoring agency Raumfahrtzentrum des Deutschen Zentrum für Luft- und Raumfahrt e.V. Königswinterer Str. 522-524 53227 Bonn		13. no. of references 5
		14. no. of tables 4
		15. no. of figures 24
16. supplementary notes		
17. presented at		
18. abstract In this project, the planetary rover SEAR was developed further for participating at the DLR SpaceBot Cup 2015. The report illustrates the results of the work during the project, among these are the development of a new robotic arm, a new body for the rover, as well as the extension of the visual sensor system by using special cameras. The development of software, especially in regard to autonomy, decision making and the operation of this space system under mission conditions are presented as well.		
19. keywords space flight, space technology, robotics, autonomy, planetary exploration, artificial intelligence		
20. publisher		21. price