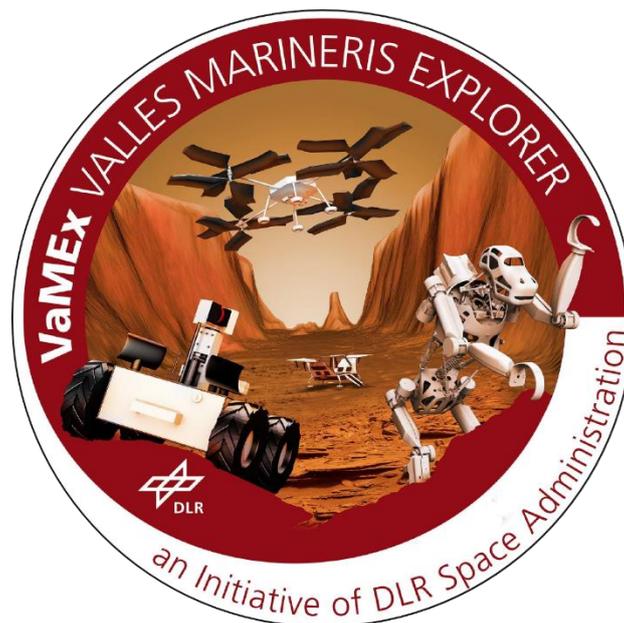


# Abschlussbericht

## VaMEx CoSMiC

FKZ 50NA1520 / FKZ 50NA1521



von:

Institut für Flugführung der TU Braunschweig,

Institut für Kommunikation und Navigation des DLR Oberpfaffenhofen,

Lehrstuhl für Flugsystemdynamik und  
Lehrstuhl für Kommunikation und Navigation der TU München

2018

Autoren:

Ulf Bestmann (ORCID: 0000-0003-4532-5274), Mario Gäbel, Thomas Krüger, Stefan Nowak;  
Institut für Flugführung der TU Braunschweig

Emanuel Staudinger, Thomas Wiedemann, Siwei Zhang; Institut für Kommunikation und  
Navigation des DLR Oberpfaffenhofen

Michael Krenmayr; Lehrstuhl für Flugsystemdynamik der TU München

Patrick Henkel, Young-Hee Lee; Lehrstuhl für Kommunikation und Navigation der TU  
München



# Inhaltsverzeichnis

Literatur .....	4
1 Einleitung.....	7
2 Voraussetzungen, unter denen das Vorhaben durchgeführt wurde.....	8
3 Planung und Ablauf des Vorhabens.....	8
4 Wissenschaftlicher und technischer Stand.....	10
5 Zusammenarbeit mit anderen Stellen .....	12
6 Erzielte Ergebnisse.....	13
7 Wichtigste Positionen des zahlenmäßigen Nachweises.....	95
8 Notwendigkeit und Angemessenheit der geleisteten Arbeit.....	97
9 Voraussichtlicher Nutzen, Verwertbarkeit.....	97
10 Bekanntgewordener Fortschritt anderer Stellen .....	106
11 Zusammenfassung .....	107

## Literatur

- [1] A. HORNING, K. M. WURM, M. BENNEWITZ, C. STACHNISS, W. BURGARD: OCTOMAP: AN EFFICIENT PROBABILISTIC 3D MAPPING FRAMEWORK BASED ON OCTREES, AUTONOMOUS ROBOTS, 2013.
- [2] S. VANNESTE, B. BELLEKENS, M. WEYN: 3DVFH+: REAL-TIME THREE-DIMENSIONAL OBSTACLE AVOIDANCE USING AN OCTOMAP, CoSYS-LAB, FACULTY OF APPLIED ENGINEERING, ANWERPEN, 2014.
- [3] I. ULRICH, J. BORENSTEIN: VFH+: RELIABLE OBSTACLE AVOIDANCE FOR FAST MOBILE ROBOTS, UNIVERSITY OF MICHIGAN, 1998.
- [4] B. YAMAUCHI. A FRONTIER-BASED APPROACH FOR AUTONOMOUS EXPLORATION. IN COMPUTATIONAL INTELLIGENCE IN ROBOTICS AND AUTOMATION, 1997. CIRA'97., PROCEEDINGS., 1997 IEEE INTERNATIONAL SYMPOSIUM ON, PAGES 146–151, JUL 1997.
- [5] OCTOMAP 3D SCAN DATASET; ALBERT-LUDWIGS-UNIVERSITÄT FREIBURG, AUTONOME INTELLIGENTE SYSTEME; [HTTP://AIS.INFORMATIK.UNI-FREIBURG.DE/PROJECTS/DATASETS/OCTOMAP/](http://ais.informatik.uni-freiburg.de/projects/datasets/octomap/); 2018.
- [6] MUR-ARTAL, R., MONTIEL, J. M. M., & TARDOS, J. D. (2015). ORB-SLAM: A VERSATILE AND ACCURATE MONOCULAR SLAM SYSTEM. *IEEE TRANSACTIONS ON ROBOTICS*, 31(5), 1147-1163.
- [7] BURRI, M., NIKOLIC, J., GOHL, P., SCHNEIDER, T., REHDER, J., OMARI, S., ... & SIEGWART, R. (2016). THE EUROC MICRO AERIAL VEHICLE DATASETS. *THE INTERNATIONAL JOURNAL OF ROBOTICS RESEARCH*, 35(10), 1157-1163.
- [8] CADENA, C., CARLONE, L., CARRILLO, H., LATIF, Y., SCARAMUZZA, D., NEIRA, J., ... & LEONARD, J. J. (2016). PAST, PRESENT, AND FUTURE OF SIMULTANEOUS LOCALIZATION AND MAPPING: TOWARD THE ROBUST-PERCEPTION AGE. *IEEE TRANSACTIONS ON ROBOTICS*, 32(6), 1309-1332.
- [9] DELMERICO, J., & SCARAMUZZA, D. (2018, MAY). A BENCHMARK COMPARISON OF MONOCULAR VISUAL-INERTIAL ODOMETRY ALGORITHMS FOR FLYING ROBOTS. IN *2018 IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION (ICRA)* (PP. 2502-2509). IEEE.
- [10] NÜTZI, G., WEISS, S., SCARAMUZZA, D., & SIEGWART, R. (2011). FUSION OF IMU AND VISION FOR ABSOLUTE SCALE ESTIMATION IN MONOCULAR SLAM. *JOURNAL OF INTELLIGENT & ROBOTIC SYSTEMS*, 61(1-4), 287-299.
- [11] MUR-ARTAL, R., & TARDÓS, J. D. (2017). VISUAL-INERTIAL MONOCULAR SLAM WITH MAP REUSE. *IEEE ROBOTICS AND AUTOMATION LETTERS*, 2(2), 796-803.
- [12] SCHREIBER, M., KÖNIGSHOF, H., HELLMUND, A. M., & STILLER, C. (2016, JUNE). VEHICLE LOCALIZATION WITH TIGHTLY COUPLED GNSS AND VISUAL ODOMETRY. IN *2016 IEEE INTELLIGENT VEHICLES SYMPOSIUM (IV)* (PP. 858-863). IEEE.
- [13] HOELLER, D., LEDERGERBER, A., HAMER, M., & D'ANDREA, R. (2017). AUGMENTING ULTRA-WIDEBAND LOCALIZATION WITH COMPUTER VISION FOR ACCURATE FLIGHT. *IFAC-PAPERS ON LINE*, 50(1), 12734-12740.
- [14] BENINI, A., MANCINI, A., & LONGHI, S. (2013). AN IMU/UWB/VISION-BASED EXTENDED KALMAN FILTER FOR MINI-UAV LOCALIZATION IN INDOOR ENVIRONMENT USING 802.15. 4A WIRELESS SENSOR NETWORK. *JOURNAL OF INTELLIGENT & ROBOTIC SYSTEMS*, 70(1-4), 461-476.
- [15] MOORE, D., OLSON, E., AND HUANG, A., "LIGHTWEIGHT COMMUNICATIONS AND MARSHALLING FOR LOW-LATENCY INTERPROCESS COMMUNICATION," 2009.

- [16] BENITEZ PÉREZ, H., ORTEGA ARJONA, J. L., AND LATIF SHABGAHI, G. R., "DEFINITION AND EMPIRICAL EVALUATION OF VOTERS FOR REDUNDANT SMART SENSOR SYSTEMS," *COMPUTACIÓN Y SISTEMAS*; VOL. 11, NO. 1, 2007, PP. 39–60.
- [17] MESKIN, N., AND KHORASANI, K., *FAULT DETECTION AND ISOLATION: MULTI-VEHICLE UNMANNED SYSTEMS*, SPRINGER SCIENCE & BUSINESS MEDIA, 2011.
- [18] EASA, "CERTIFICATION SPECIFICATIONS AND ACCEPTABLE MEANS OF COMPLIANCE FOR NORMAL, UTILITY, AEROBATIC, AND COMMUTER CATEGORY AEROPLANES," [HTTPS://WWW.EASA.EUROPA.EU/SYSTEM/FILES/DFU/CS-23%20AMENDMENT%204.PDF](https://www.easa.europa.eu/system/files/dfu/cs-23%20amendment%204.pdf).
- [19] SAE94910, "AEROSPACE - VEHICLE MANAGEMENT SYSTEMS - FLIGHT CONTROL DESIGN, INSTALLATION AND TEST OF, MILITARY UNMANNED AIRCRAFT, SPECIFICATION GUIDE FOR: AEROSPACE RECOMMENDED PRACTICE," SAE, DECEMBER 2012.
- [20] EASA, "SOFTWARE ASPECTS OF CERTIFICATION: CERTIFICATION MEMORANDUM," EASA, [HTTPS://WWW.EASA.EUROPA.EU/SITES/DEFAULT/FILES/DFU/CERTIFICATION-DOCS-CERTIFICATION-MEMORANDUM-EASA-CM-SWCEH-002-ISSUE-01-REV-01-SOFTWARE-ASPECTS-OF-CERTIFICATION.PDF](https://www.easa.europa.eu/sites/default/files/dfu/certification-docs-certification-memorandum-easa-cm-swceh-002-issue-01-rev-01-software-aspects-of-certification.pdf).
- [21] RTCA/DO-178C, "SOFTWARE CONSIDERATIONS IN AIRBORNE SYSTEMS AND EQUIPMENT CERTIFICATION," RTCA, DECEMBER 2011.
- [22] HEISE, C. D., *SURVIVABLE FLIGHT CONTROL WITH GUARANTEED STABILITY AND PERFORMANCE CHARACTERISTICS*.
- [23] SAE AS94900, "AEROSPACE - FLIGHT CONTROL SYSTEMS - DESIGN, INSTALLATION AND TEST OF PILOTED MILITARY AIRCRAFT, GENERAL SPECIFICATION FOR: AEROSPACE STANDARD," SAE, JULY 2007.
- [24] SAE ARP94910, "AEROSPACE - VEHICLE MANAGEMENT SYSTEMS - FLIGHT CONTROL DESIGN, INSTALLATION AND TEST OF, MILITARY UNMANNED AIRCRAFT, SPECIFICATION GUIDE FOR: AEROSPACE RECOMMENDED PRACTICE," SAE, DECEMBER 2012.
- [25] MIL-HDBK-1797, "FLYING QUALITIES OF PILOTED AIRCRAFT," DEPARTMENT OF DEFENCE.
- [26] MERHEB, A.-R., NOURA, H., AND BATEMAN, F., "DESIGN OF PASSIVE FAULT-TOLERANT CONTROLLERS OF A QUADROTOR BASED ON SLIDING MODE THEORY," *INTERNATIONAL JOURNAL OF APPLIED MATHEMATICS AND COMPUTER SCIENCE*; VOL. 25, NO. 3, 2015, PP. 561–576.
- [27] FREDDI, A., LONGHI, S., MONTERIÙ, A., AND PRIST, M., "ACTUATOR FAULT DETECTION AND ISOLATION SYSTEM FOR AN HEXACOPTER," *MECHATRONIC AND EMBEDDED SYSTEMS AND APPLICATIONS (MESA), 2014 IEEE/ASME 10TH INTERNATIONAL CONFERENCE ON*, 2014, PP. 1–6.
- [28] ACHTELNIK, M., DOTH, K.-M., GURDAN, D., AND STUMPF, J., "DESIGN OF A MULTI ROTOR MAV WITH REGARD TO EFFICIENCY, DYNAMICS AND REDUNDANCY," *AIAA GUIDANCE, NAVIGATION, AND CONTROL CONFERENCE*, 2012, P. 4779.
- [29] HEISE, C. D., FALCONI, G. P., AND HOLZAPFEL, F., "HEXACOPTER OUTDOOR FLIGHT TEST RESULTS OF AN EXTENDED STATE OBSERVER BASED CONTROLLER," *AEROSPACE ELECTRONICS AND REMOTE SENSING TECHNOLOGY (ICARES), 2014 IEEE INTERNATIONAL CONFERENCE ON*, 2014, PP. 26–33.
- [30] VEY, D., AND LUNZE, J., "EXPERIMENTAL EVALUATION OF AN ACTIVE FAULT-TOLERANT CONTROL SCHEME FOR MULTIROTOR UAVs," *CONTROL AND FAULT-TOLERANT SYSTEMS (SYS TOL), 2016 3RD CONFERENCE ON*, 2016, PP. 125–132.

- [31] YANG, Y., IWAKURA, D., NAMIKI, A., NONAMI, K., AND WANG, W., "AUTONOMOUS FLIGHT OF HEXACOPTER UNDER PROPULSION SYSTEM FAILURE," *JOURNAL OF ROBOTICS AND MECHATRONICS*; VOL. 28, No. 6, 2016, PP. 899–910.
- [32] MUELLER, M. W., AND D'ANDREA, R., "STABILITY AND CONTROL OF A QUADROPTER DESPITE THE COMPLETE LOSS OF ONE, TWO, OR THREE PROPELLERS," *ROBOTICS AND AUTOMATION (ICRA), 2014 IEEE INTERNATIONAL CONFERENCE ON*, 2014, PP. 45–52.
- [33] FALCONÍ, G. P., ANGELOV, J., AND HOLZAPFEL, F., "ADAPTIVE FAULT-TOLERANT POSITION CONTROL OF A HEXACOPTER SUBJECT TO AN UNKNOWN MOTOR FAILURE," *INT. J. APPL. MATH. COMPUT. SCI*; VOL. 28, No. 2, 2018, PP. 309–321.
- [34] FALCONI, G. P., AND HOLZAPFEL, F., "POSITION TRACKING OF A MULTICOPTER USING A GEOMETRIC BACKSTEPPING CONTROL LAW," *PROCEEDINGS OF THE 2ND CEAS SPECIALIST CONFERENCE ON GUIDANCE, NAVIGATION & CONTROL (EUROGNC), 2013, PP*, 2013, PP. 1527–1545.
- [35] KHALIL, H. K., AND GRIZZLE, J. W., *NONLINEAR SYSTEMS*, PRENTICE HALL UPPER SADDLE RIVER, NJ, 2002.
- [36] FALCONI, G. P., ANGELOV, J., AND HOLZAPFEL, F., "HEXACOPTER OUTDOOR FLIGHT TEST RESULTS USING ADAPTIVE CONTROL ALLOCATION SUBJECT TO AN UNKNOWN COMPLETE LOSS OF ONE PROPELLER," *CONFERENCE ON CONTROL AND FAULT-TOLERANT SYSTEMS (SysTol)*, 2016, PP. 373–380.
- [37] CHOWDHARY, G., AND JOHNSON, E., "CONCURRENT LEARNING FOR CONVERGENCE IN ADAPTIVE CONTROL WITHOUT PERSISTENCY OF EXCITATION," *DECISION AND CONTROL (CDC), 2010 49TH IEEE CONFERENCE ON*, 2010, PP. 3674–3679.
- [38] ESA, "MARS CLIMATE DATABASE," [HTTP://WWW-MARS.LMD.JUSSIEU.FR/MARS/INFO\\_WEB/INDEX.HTML](http://www-mars.lmd.jussieu.fr/mars/info_web/index.html), [RETRIEVED 19 JANUARY 2019].
- [39] DANIEL BODMER, *DESIGN OF A MATLAB/SIMULINK BASED SIMULATION MODEL FOR A FLYING VEHICLE ON MARS*, DECEMBER 2017.
- [40] JOHNSON, W., *HELICOPTER THEORY*, COURIER CORPORATION, 2012.
- [41] PHILLIPS, W. F., *MECHANICS OF FLIGHT*, JOHN WILEY & SONS, 2004.
- [42] HERCEG, M., KVASNICA, M., JONES, C. N., AND MORARI, M., "MULTI-PARAMETRIC TOOLBOX 3.0," *CONTROL CONFERENCE (ECC), 2013 EUROPEAN*, 2013, PP. 502–510.
- [43] SASCHA RIMET, *DYNAMICS AND CONTROLLER EMULATION OF A MARS VEHICLE*, MUNICH, OCTOBER 2018.

# 1 Einleitung

Die Suche nach Leben ist eines der spannendsten Themen der Raumfahrt. Sie erfolgte - mit Ausnahme der amerikanischen Mondmissionen - bisher nur mit Raumsonden und teilautonom agierenden Erkundungsfahrzeugen, wie den drei Mars-Rovern „Spirit“, „Opportunity“ und „Curiosity“ der NASA.



**Abbildung 1.1: Valles Marineris**



**Abbildung 1.2: Mars-Rover Curiosity**

Raumsonden erlauben Aufnahmen weitläufiger Areale aus einer Umlaufbahn um den Planeten. Große Strukturen wie Canyons und Berge können identifiziert werden. Bodengebundene Fahrzeuge ermöglichen den Zugang zu barrierefreien Gebieten, weshalb sich vor allem die hauptsächlich mit Staub und Sand bedeckte Oberfläche für die Exploration mit unbemannten Vehikeln eignet. Schwerer zugängliche Gebiete wie z.B. Canyons und Höhlen lassen sich mit den bisher entwickelten Systemen nur schwer oder gar nicht erkunden. Gerade solche Gebiete sind aber von besonderem wissenschaftlichem Interesse, z.B. bei der Suche nach extraterrestrischem Leben.

Ein solches Beispiel ist das „Valles Marineris“, ein 4000km langes, 700km breites und 7000m tiefes Canyonsystem, das sich entlang des Marsäquators erstreckt (s. Abbildung 1.1). Aufgrund der Tiefe des Canyons ist zu erwarten, dass der atmosphärische Druck oberhalb des Tripelpunkts von Wasser liegt. In den Sommermonaten besteht deshalb die Möglichkeit, dass aufgrund der erhöhten Temperaturen Wasser auch in flüssiger Form vorhanden ist. Dies stellt die Grundlage für die Entstehung von Leben dar. Neben diesem Aspekt der Astrobiologie bietet eine Exploration dieses Canyons einen massiven Wissensgewinn auf dem Gebiet der Planetengeologie.

Eine Exploration des Canyonsystems erfordert Aufgaben denen heutige Systeme nicht gewachsen sind. Hindernisse in Form von Geröll und Gesteinsbrocken schränken den Einsatz bisheriger großer und sperriger Vehikel, wie dem Mars-Rover „Curiosity“ (s. Abbildung 1.2) stark ein. Aufgrund der erhöhten Atmosphärendichte am Grunde des Canyons ist allerdings der Einsatz fliegender und schwebefähiger Vehikel möglich. Für die Erkundung

des Canyonsystems bietet sich deshalb ein heterogener, automatischer Schwarm bestehend aus speziellen bodengebundenen und flugfähigen Vehikeln an.

In einem Schwarm Szenario stellt vor allem die selbstständige Navigation eine Schlüsseltechnologie dar, um eine sinnvolle Aufklärung zu ermöglichen. Neben dem Schwarm als Ganzen stellt die Regelung der fliegenden Schwarmteilnehmer in einer nicht-terrestrischen Atmosphäre eine Herausforderung dar.

In dem hier dokumentierten Projekt sollen Verfahren für eine automatische Schwarmaufklärung untersucht, entwickelt und getestet werden. Eine automatisch erfolgende Exploration des Valles Marineris stellt ein Fernziel dar, dessen Realisierung nach Projektende weiter verfolgt werden soll.

## 2 Voraussetzungen, unter denen das Vorhaben durchgeführt wurde

Bei VaMEx-CoSMiC handelt es sich um ein Verbundvorhaben mit den beteiligten Partnern TU Braunschweig IFF (Konsortialführer), TU München KN, TU München FSD, sowie dem DLR-KN. Die akademischen Partner wurden zu 100% vom DLR Raumfahrtmanagement gefördert. Das Institut für Kommunikation und Navigation des DLR Oberpfaffenhofen war im Unterauftrag der TU Braunschweig eingebunden, um spezielle benötigte Technologien weiterzuentwickeln. Grundlage bildeten die im Vorgängervorhaben "VaMEx - Valles Marineris Explorer" (FKZ 50NA1212-13) erzielten Arbeitsergebnisse.

Die beteiligten Einrichtungen waren:

- Institut für Flugführung der Technischen Universität Braunschweig,
- Lehrstuhl für Kommunikation und Navigation der Technischen Universität München,
- Lehrstuhl für Flugsystemdynamik der Technischen Universität München,
- Institut für Kommunikation und Navigation des DLR Oberpfaffenhofen.

## 3 Planung und Ablauf des Vorhabens

### 3.1 Zeitplanung

Das Projekt war für eine Dauer von 3 Jahren konzipiert. Die folgende Abbildung enthält eine Übersicht über den zeitlichen Ablauf des Projektes.

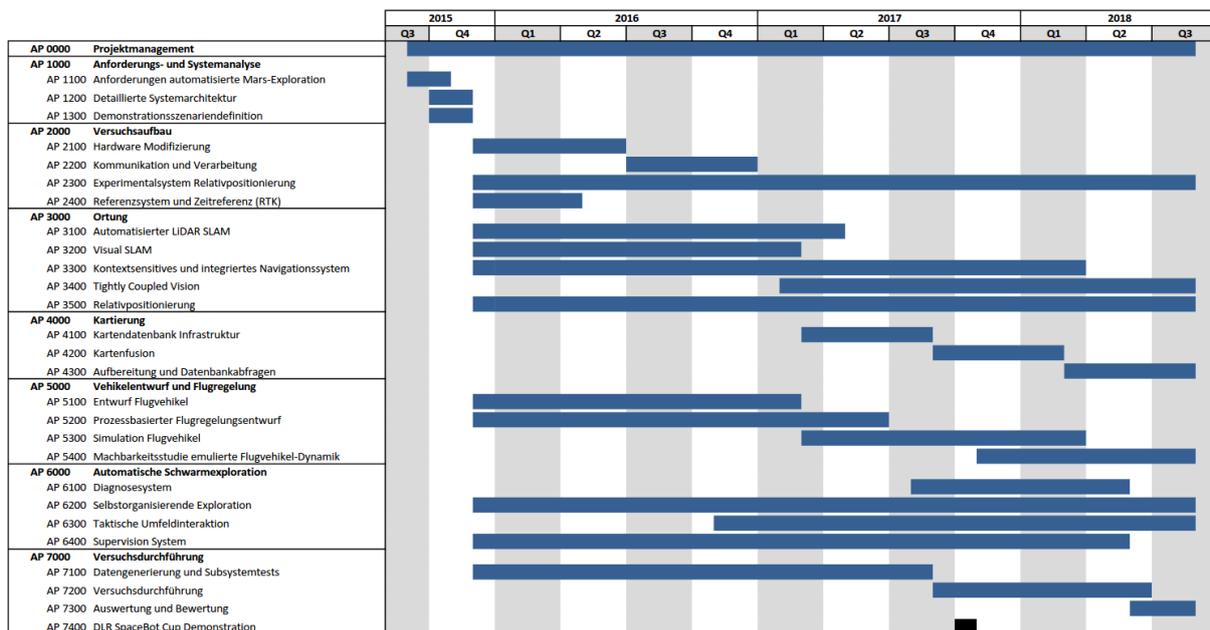


Abbildung 3.1: Gantt Chart Valles Marineris Explorer

Das Projekt ist zum 01.09.2015 gestartet und zum 31.08.2018 erfolgreich abgeschlossen worden.

### 3.2 Arbeitspaketstruktur

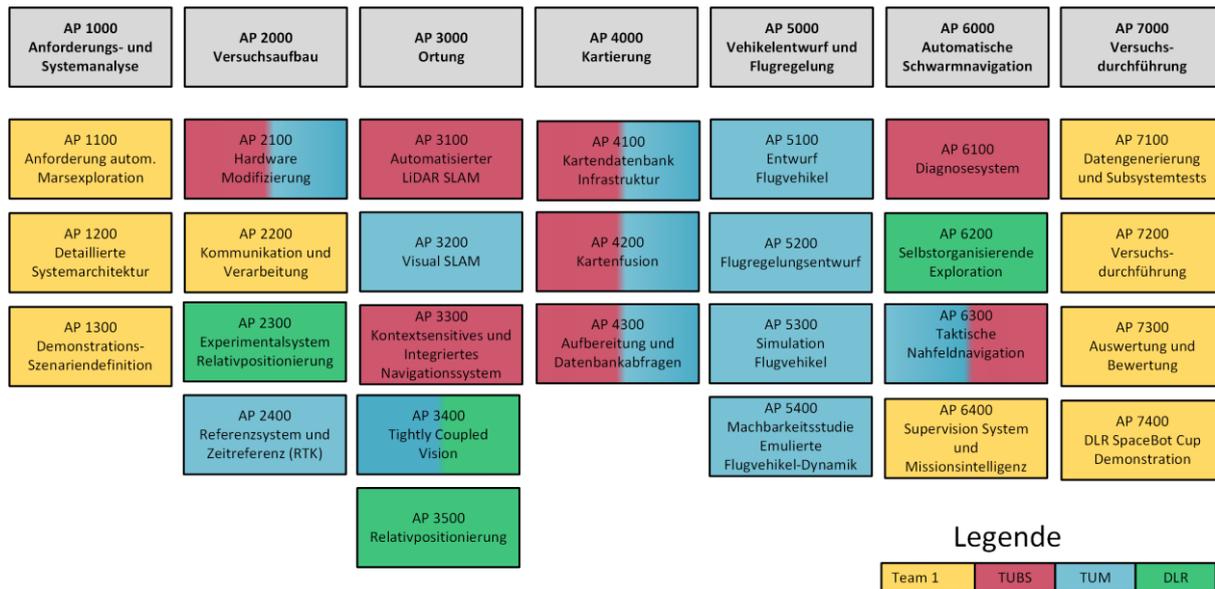


Abbildung 3.2: Arbeitspaketstruktur

Das Projekt gliederte sich in mehrere Hauptarbeitspakete. Im Hauptarbeitspaket 1000 wurden die Anforderungen an die zu untersuchenden Systeme konkretisiert. Ebenfalls musste festgelegt werden, wie bestimmte Funktionalitäten zu testen sind, da keine Versuche auf dem Mars durchgeführt werden konnten und einige Systeme im Projektverlauf nicht mit vertretbarem Aufwand miniaturisiert oder beschafft werden konnten. Unter diesen Randbedingungen mussten Ersatzszenarien definiert und abgestimmt werden.

In dem Arbeitspaket 2000 sollten die Versuchsplattformen und deren Sensorenpakete modifiziert und aufgebaut werden. Diese Plattformen wurden mit weiteren Sensoren ausgestattet und sollten über die entsprechende Infrastruktur für das Auswerten und die Weiterleitung der gewonnenen Daten verfügen.

In den Hauptarbeitspaketen 3000, 4000, 5000 und 6000 sollten die technologischen Grundlagen in Form von Algorithmen und Verfahren erarbeitet werden. Hierbei sollte, soweit möglich, auf vorhandene Geräte und Infrastruktur aufgebaut werden. So konnten an den Instituten vorhandene IMU, UAV oder Navigationsempfänger (zu Referenzzwecken) je nach Eignung und Verfügbarkeit mit verwendet werden.

In Hauptarbeitspaket 6000 sollten die Verfahren und Algorithmen für die Schwarmintelligenz und für die selbstorganisierende Exploration entwickelt werden.

In Arbeitspaket 7000 wurden anhand der in AP 1000 definierten Testszenarien einzelne Systemkomponenten und das Gesamtsystem getestet und mit den Anforderungen verglichen.

## 4 Wissenschaftlicher und technischer Stand

### Flugregelung

Klassische lineare Regelung, wie sie bis zum heutigen Tag fast ausschließlich in der Luft- und Raumfahrt üblich sind, bedürfen eines möglichst exakten linearen Modells für deren Auslegung. Die Dynamik von Fluggeräten jeglicher Art ist im Allgemeinen hochgradig nichtlinear. Um ein lineares Modell zu erhalten muss die nichtlineare Dynamik um einen stationären Punkt, einen sogenannten Trimpunkt linearisiert werden. Dieses lineare Modell gibt die Dynamik nur näherungsweise und auch nur in der Umgebung des Trimpunktes wieder. Wenn der auf diese Weise ausgelegte Regler in der realen Anwendung eingesetzt wird, muss verifiziert werden, ob das lineare Modell die Realität mit ausreichender Genauigkeit widerspiegelt. Der Aufwand für die möglichst exakte Modellierung verschlingt dabei in hohem Maße Zeit und Kosten und in einigen Fällen können Modellparameter nur geschätzt werden. Der Regler ist wegen der Linearisierung auch nur in einer lokalen Umgebung um den Trimpunkt gültig. Um hochagile Manöver auszuführen, die den gesamten Betriebsbereich ausreizen, ist aber eine globale Gültigkeit wünschenswert. Überdies erfüllt der so ausgelegte Regler auch lokal die Leistungsanforderungen nur unter Nominalbedingungen zufriedenstellend. Ändert sich die Dynamik plötzlich während des Betriebes, beispielsweise aufgrund von Fehlerfällen, wird das lineare Modell der Strecke hinfällig und der Regler muss deaktiviert werden.

Bisher kaum verbreitet sind nichtlineare Flugregler. Die F-35 von Lockheed Martin ist das erste nichtlinear geregelte kommerzielle Flugzeug. Nichtlineare Regelungen besitzen über nahezu den gesamten Zustandsraum Gültigkeit und ermöglichen damit auch präzise und hochagile Manöver.

Speziell die adaptive Regelung verspricht eine Lösung für die Probleme konventioneller Regler, da sie sich automatisch an eine unbekannte Dynamik anpasst. Damit wird der Modellierungsaufwand zur Auslegung drastisch reduziert und Fehlerfälle im Betrieb können ausgeglichen werden, solange das Flugzeug noch physikalisch fliegbar ist. Innerhalb dieser Randbedingung muss die Art des Fehlers dem nichtlinearen Regler im Vergleich zu seinem linearen Gegenstück auch nicht bekannt sein. Eine Langzeitmission auf dem Planeten Mars motiviert die Benutzung nichtlinearer Regelungsalgorithmen, da auch unvorhergesehene Schadensfälle innerhalb der physikalischen Grenzen kompensiert werden können.

Auch im Vorgängerprojekt VaMEx 1 wurde ein adaptiver Regler innerhalb einer nichtlinearen Reglerstruktur benutzt, um Parameterfehler sowie die Degradation von Aktuatoren zu kompensieren. Sie alle haben die Erwartungen eindrucksvoll erfüllt und motivieren, weitere Forschungsarbeiten in Richtung industrieller Anwendungen durchzuführen.

### Optimalsteuerung / Trajektorienoptimierung

Die Berechnung von optimalen Trajektorien kommt immer dann zum Einsatz, wenn für ein gegebenes dynamisches System und Mission eine bestimmte Kostenfunktion minimiert werden soll. Diese kann beliebig gewählt werden, in den meisten Fällen ist jedoch eine Minimierung des Energieverbrauchs oder aber auch die geringste Flugzeit gewünscht. Der Zustands- und Steuergrößenverlauf, der zu einer Minimierung der Kostenfunktion führt, beschreibt die Lösung des sogenannten Optimalsteuerungsproblems.

In der Optimalsteuerung unterscheidet man zwei Lösungsansätze: die direkten und die indirekten Methoden. Bei indirekten Methoden wird eine Optimalitätsbedingung erstellt, die im Anschluss entweder analytisch oder numerisch gelöst wird. Da für komplexe reale Anwendungen diese Lösungsmethode schnell an ihre Grenzen stößt, verwendet man meistens direkte Methoden. Hierbei wird das Optimalsteuerungsproblem zeitlich diskretisiert (Kollokation / Multiple Shooting) und in ein Parameteroptimierungsproblem überführt. Dieses kann im Anschluss mit handelsüblichen Algorithmen gelöst werden (z.B. ipopt).

## Visuelle und LiDAR gestützte Navigation

Die visuelle Navigation und die Navigation mit LiDAR-Sensoren ist zu einem großen Forschungsthema in den vergangenen Jahren aufgestiegen, da die Satellitennavigation in einigen Anwendungsbereichen nicht oder nur unzureichend verfügbar ist. Kameras können eine präzise Navigation ähnlich dem menschlichen Auge/ Gehirn ermöglichen. Die Daten sind allerdings um ein Vielfaches größer als in der Satellitennavigation und erfordern eine Extraktion der wichtigsten Merkmale. In anspruchsvollen Umgebungsbedingungen mit sandigem Untergrund kann die visuelle Odometrie genauere Informationen liefern als die herkömmliche rad-gebundene Odometrie, deren Genauigkeit durch Schlupf beeinträchtigt ist. Mit Hilfe von LiDAR-Sensoren ist es zudem möglich zwei- oder dreidimensionale Punktwolken der näheren Umgebung aufzuzeichnen und Rückschlüsse auf die Pose des Sensors bzw. der Trägerplattform zu ziehen. Neben der Trajektorien-Bestimmung kommt der gemeinsamen Positionsbestimmung und Kartierung (Visual Simultaneous Localization And Mapping, VSLAM oder LiDAR SLAM) eine große Bedeutung zu. Die rein bildbasierten VSLAM Verfahren vertrauen im Vergleich zu Laser-Scannern zu einem erheblich höheren Maß der korrekten Merkmals-Zuordnung.

Die Bildsensoren können lediglich eine zwei-dimensionale Projektion des dreidimensionalen Raums erfassen. Dennoch lässt sich von der Eigenbewegung der Kamera eine 3D-Information ableiten. Dieser Prozess ist unter dem Stichwort „structure from motion“ bekannt. Für die Rekonstruktion der 3D Information ist eine Tiefeninformation unerlässlich. In den meisten Fällen (einschließlich des Menschen) wird diese über eine Stereo-Anordnung erfasst.

Eine Stereo-Anordnung lässt sich allerdings in einigen Systemen aufgrund der stringenten Anforderungen an die Größe und die verfügbare Rechenleistung nicht immer realisieren. Als Ersatz wurden verschiedene Ansätze zur visuellen Navigation mit *monokularen* Kameras entwickelt. Es haben sich allerdings zwei größere Herausforderungen herauskristallisiert: Zum einen kann die globale absolute Skalierung nicht ausschließlich mit monokularen Kameras bestimmt werden. Zum anderen kann die Schätzung der Positionsänderung einen Fehler aufweisen und der resultierende relative Skalierungsfehler sich fortpflanzen. Die Genauigkeit und Zuverlässigkeit der Positions- und Lageschätzung lässt sich durch eine Sensor-Fusion der Kamera- und IMU-Daten erheblich verbessern. In einer eng-gekoppelten Sensorfusion kann auch die Drift der IMU bestimmt werden.

## RTK

RTK ist ein Verfahren zur zentimeter-genauen, relativen Positionsbestimmung mit zwei GNSS-Empfängern. Die hohe Genauigkeit wird durch die Auswertung der Trägerphasenmessungen ermöglicht, die mit Millimeter- bis Zentimeter-Genauigkeit getrackt werden können. Daher bietet sich RTK als Referenzsystem an. Trägerphasenmessungen sind allerdings von Phasensprüngen und Codemessungen von erheblichen Mehrwegefehlern betroffen. Ein zuverlässiges RTK-System beinhaltet daher eine eng-gekoppelte GPS/INS-Sensorfusion, um Phasensprünge korrigieren und Mehrwegefehler schätzen zu können. Ein RTK-System kann eine „float“ und „fixed“ Lösung liefern: Die „fixed“ Lösung weist die größtmögliche Genauigkeit auf und ist in günstigen Umgebungsbedingungen verfügbar; andernfalls wird in der Regel eine „float“ Lösung mit etwas verringerter Genauigkeit ausgegeben.

## Exploration

Bei der Exploration von unbekanntem Umgebungen mit Robotern stellen die Explorationszeit und die erkundete Fläche die Hauptparameter der Exploration dar. Wenn diese Exploration autonom erfolgen soll, muss der Roboter selbst erkennen was unzureichend oder gar nicht erkundet ist und wann ein Gebiet ausreichend exploriert ist (Termination des Algorithmus). Dabei muss der Roboter ebenfalls die Pfadplanung in einer Karte selbst übernehmen, in dieser Karte Hindernisse erkennen und ihnen ausweichen (Hindernis-Vermeidung). Im Fall einer Multi-Agenten Exploration kommt die Koordination der einzelnen Agenten noch hinzu. All diese Herausforderungen müssen gelöst werden, damit ein Schwarm von Robotern eine automatische Exploration erfolgreich durchführen kann.

## **Kooperative Schwarmnavigation mit Funksignalen**

Bei terrestrischen Positionierungssystemen und Lokalisierungsansätzen werden im Allgemeinen infrastrukturbasierte Ansätze verfolgt. Dabei greift man üblicherweise auf GNSS wie GPS oder GALILEO oder Kommunikationsnetze wie UMTS, 3GPP-LTE zurück. Für eine genaue Lokalisierung wird eine direkte Sichtverbindung zwischen Sender und Empfänger vorausgesetzt. Diese direkte Sichtverbindung ist bei klassischen Kommunikationsnetzen nicht erfüllt. Kooperative Ansätze, welche zusätzlich eine Distanzschätzung zwischen einzelnen Funkteilnehmern durchführen, wurden deshalb bereits untersucht und werden als zielführend betrachtet. Für eine Schwarmnavigation auf dem Mars kommt weiters hinzu, dass sämtliche Infrastruktur fehlt. Erste Ansätze zur infrastrukturlosen Positionierung speziell für die Schwarmnavigation und die Schwarmformationsschätzung wurden im Projekt VaMEx untersucht. Für eine optimale Schwarmnavigation müssen Teilaspekte wie der effiziente, dezentrale Kanalzugriff, die zeit-basierte Distanzschätzung und die dezentrale Schwarmsteuerung eng gekoppelt werden. Aktueller Gegenstand der Forschung sind sogenannte Cyber-Physical Systems (CPS), in welcher Lokalisierung eine wichtige Rolle spielt. Allerdings nutzen diese CPS wiederum terrestrische Infrastruktur, weshalb aktuelle Ansätze aus diesem Bereich nur bedingt für eine Schwarmnavigation auf dem Mars einsetzbar sind. Daher müssen neue Ansätze untersucht, und bestehende Ansätze aus z.B. dem Vorgängerprojekt VaMEx verbessert werden, um die Schwarmnavigation mit der Exploration optimal verknüpfen zu können.

## **5 Zusammenarbeit mit anderen Stellen**

Das Gesamtvorhaben VaMEx-CoSMiC war ein Verbundvorhaben mit folgenden beteiligten Einrichtungen:

- Institut für Flugführung der Technischen Universität Braunschweig,
- Lehrstuhl für Kommunikation und Navigation der Technischen Universität München,
- Lehrstuhl für Flugsystemdynamik der Technischen Universität München,
- Institut für Kommunikation und Navigation des DLR Oberpfaffenhofen im Unterauftrag der TU Braunschweig.

Verbundführer war das Institut für Flugführung der Technischen Universität Braunschweig.

## 6 Erzielte Ergebnisse

Die im Folgenden beschriebenen erzielten Ergebnisse enthalten sowohl die Arbeiten des Teilprojektes TU Braunschweig (FKZ 50 NA 1520) als auch die Arbeiten der Teilprojekte DLR KN (FKZ 50 NA 1520) und TU München (FKZ 50 NA 1521). Dies ist der Tatsache geschuldet, dass die Arbeiten in dem Verbundvorhaben VaMEx-CoSMiC in vielen Bereichen stark verzahnt durchgeführt wurden.

### 6.1 Anforderungs- und Systemanalyse

#### 6.1.1 Robustheit und Fehlertoleranz der fliegenden Schwarmteilnehmer

Der Einsatz des Schwarms über einen langen Zeitraum hinweg erfordert die Robustheit der einzelnen Schwarmteilnehmer gegenüber Degradations- bzw. Schadensfällen. Letztere lassen sich für fliegende Schwarmteilnehmer in Hardwarefehler und physikalische Fehler unterteilen.

Hardwarefehler vereinigen unter anderem Fehler der Sensoren als auch Fehler, die z.B. durch Bitflips auf den mitgeführten Rechnern entstehen können. Für die fliegenden Schwarmteilnehmer ist ein funktionierendes Set an Sensoren notwendig, um eine vollständige Navigationslösung erhalten zu können. Während der Ausfall der Positionslösung unter bestimmten Umständen kurzfristig verkraftet werden kann, führt der Wegfall der Lage- und Rateninformation fast zwangsläufig zu einem unerwünschten Flugzustand. Die Anforderungen an eine vollständige Navigationslösung können wiederum durch Robustheitsanforderungen an die Hardware oder Redundanz erfüllt werden.

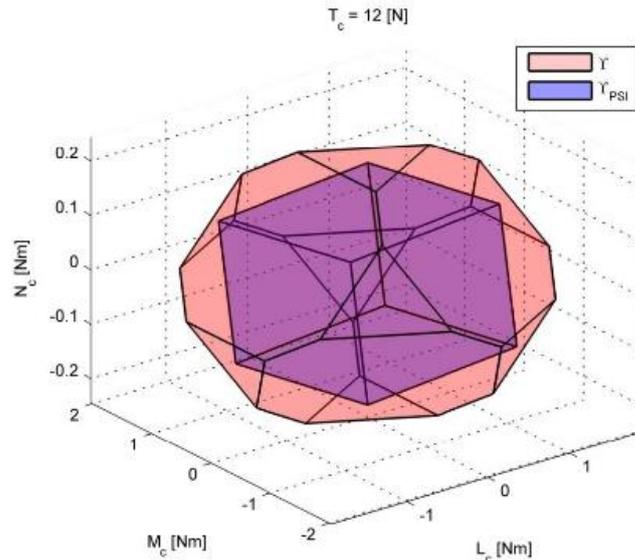
Physikalische Fehler beziehen sich auf Schadensfälle, die eine Änderung der Flugphysik zur Folge haben. Dies kann z.B. durch den Ausfall eines Antriebssystems erfolgen. Abbildung 6.1.1 zeigt beispielhaft den Momentenraum eines fliegenden Schwarmteilnehmers bei  $12N$  Schub. Das äußere Polytop definiert die vom Vehikel physikalisch erreichbaren Momente. Für einen Schwebeflug muss der Ursprung innerhalb des Polytops liegen damit kein resultierendes Moment entsteht, das das Vehikel in Bewegung versetzt. Die gewählte Darstellung über Polytope ermöglicht somit eine einfache Analyse ob gewünschte Flugzustände überhaupt erreicht werden können.

Es ist allerdings nicht sichergestellt, dass der Regler die gewünschte Kombination an Schub und Momenten auch erreichen kann. Beispielhaft zeigt das innere Polytop in Abbildung 6.1.1 welche Momentenkombinationen vom Regelungsalgorithmus erreicht werden können. Dementsprechend leiten sich die Anforderungen an die Robustheit des Systems gegenüber physikalischer Degradation nicht nur aus den physikalischen Randbedingungen des Systems ab, sondern sind auch vom eingesetzten Regelungsalgorithmus abhängig.

Die elegante Darstellung über Polytope im Momentenraum erlaubt eine Klassifizierung möglicher Schadensfälle: Die erdenklichen Fehlerfälle müssen nicht einzeln betrachtet werden, sondern können basierend auf ihrer Auswirkung auf den Momentenraum zusammengefasst werden. Dies erlaubt eine effiziente Untersuchung, auch wenn sich das betrachtete Vehikel oder der Regelungsalgorithmus ändern.

Neben der Untersuchung der reinen Robustheit gegenüber Schadensfällen existieren Algorithmen, die es ermöglicht Fehler zu detektieren und zu klassifizieren. Z.B. kann ein Votingssystem durch redundante Hardware feststellen, ob ein Rechner fehlerhafte Signale liefert (u.A. ). Des Weiteren existieren z.B. für den Ausfall von Aktuatoren sogenannte „Fault Detection and Isolation“ Methoden, die es erlauben fehlerhafte Effektoren zu detektieren und den Fehler abzuschätzen. Die Erkennung und Klassifizierung von Schadensfällen erlaubt es dem einzelnen Vehikel und der Missionsintelligenz abzuschätzen inwiefern eine Fortsetzung der Mission durch den beschädigten Schwarmteilnehmer möglich ist. Ist das Vehikel immer noch einsatzbereit erlaubt diese Analyse eine Abschätzung der Einschränkungen der Fähigkeiten des Schwarmteilnehmers, die durch den Schaden hervorgerufen wurden. Dies ermöglicht den weiteren Explorationseinsatz eines Schwarmteilnehmers mit einem

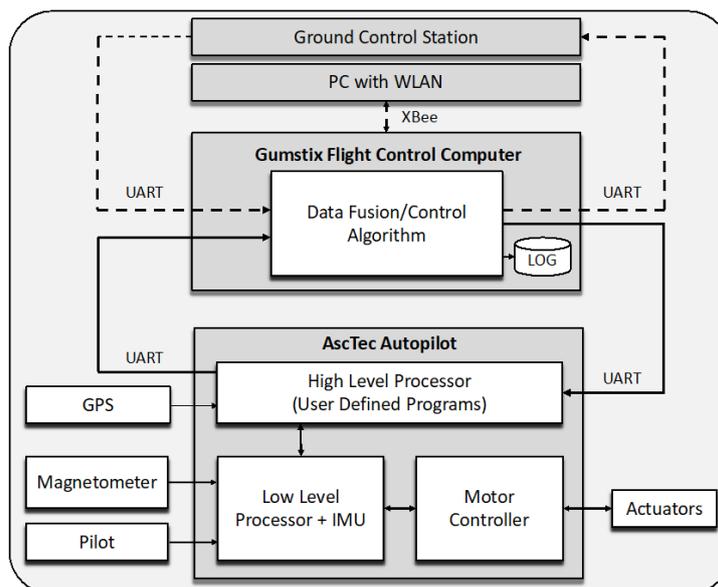
fehlerhaften speziellen Sensor auch dann wenn er seine Spezialaufgabe nicht mehr ausführen kann.



**Abbildung 6.1.1: Beispielhafter Momentenraum eines fliegenden Schwarmteilnehmers bei 12N Schub**

### 6.1.2 Hexakopter Systemarchitektur

Die Architektur vom Hexakopter Autopilot kann Abbildung 6.1.2 entnommen werden. Diese Architektur wurde im Laufe des Vorgängerprojekts VaMEx erfolgreich benutzt. In CoSMiC sollen die Schnittstellen zu der Bodenstation sowie zu der übergeordneten Schwarmintelligenz und Missionsplanung angepasst werden. Ziel ist, dass die AscTec Firefly's in derselben Weise wie die anderen Schwarmteilnehmer bedient werden können. Dafür wird das MAVLink Protokoll implementiert, um die Daten zwischen den unterschiedlichen Systemen zu Übertragen. Außerdem werden ausgewählte Modi vom Pixhawk Autopilot zur Verfügung gestellt. Dieses Vorgehen erlaubt das Weiternutzen des vertrauten Systems und reduziert somit die Entwicklungszeit. Außerdem wird eine gewisse Modularität erreicht, indem sichergestellt wird, dass die Schwarmteilnehmer über das gleiche Protokoll und die gleichen Befehle angesprochen werden können.

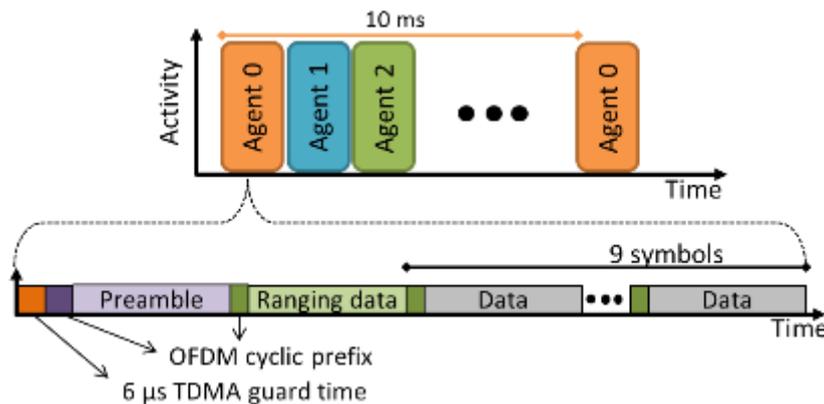


**Abbildung 6.1.2: FSD Hexakopter Autopilot**

## 6.2 Versuchsaufbau

### 6.2.1 Experimentalsystem Relativpositionierung

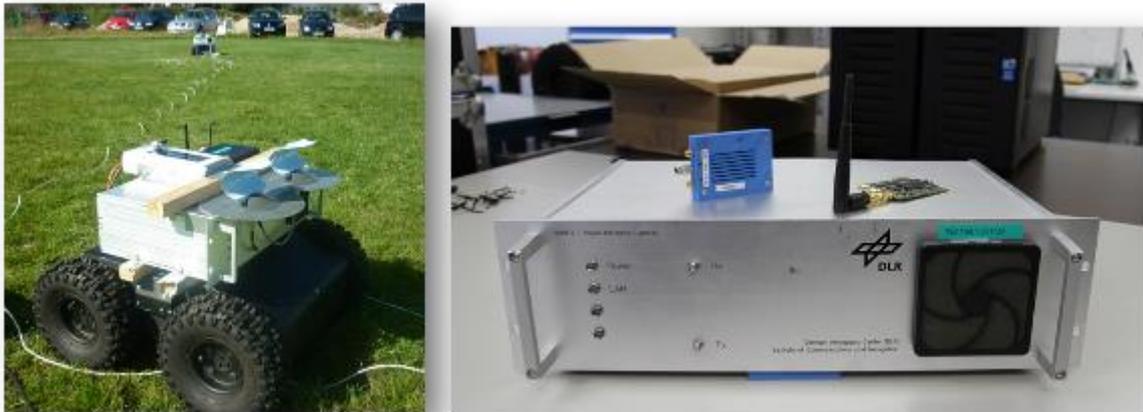
Das Konzept für ein kombiniertes Kommunikations- und Lokalisierungssystem aus dem Vorgängerprojekt VaMEx wurde in einem ersten Schritt überarbeitet. Bei dieser Überarbeitung kamen maßgeblich praktische Einschränkungen einer späteren Realisierung zum Tragen. Dieses neue Konzept erlaubt eine Kommunikation, Lokalisierung, und Zeitsynchronisation in einem Netzwerk von Agenten und basiert auf einem Mehrträgermodulationsverfahren (OFDM). Für den interferenzfreien Kanalzugriff wird ein Zeitmultiplex (TDMA) ohne zentralen Koordinator eingesetzt. D.h., dass Agenten sich relativ zu den Zeitschlitzten der Nachbaragenten synchronisieren und die Sendezeitpunkte ihrer OFDM-Symbole nachregeln. Somit erreicht man einen robusten Mehrnutzerkanalzugriff ohne Single-Point-of-Failure. Die Länge der Zeitschlitzte, die Signalbandbreite, etc., sind konzeptionell flexibel für verschiedene Anwendungen anpassbar, z.B. wenn nur wenige Agenten für eine Raumfahrtmission benötigt werden. Für VaMEx-CoSMiC wurden die Parameter wie folgt gewählt (Auszug): OFDM Signalbandbreite von 25 MHz mit 1024 Unterträgern, und ein Zugriff für 20 Agenten innerhalb von 10 ms. Für das Design des OFDM-Systems wurde von einem Explorationsszenario auf der Mars-Oberfläche ausgegangen. Abbildung 6.2.1 zeigt exemplarisch das TDMA-Schema mit den OFDM-Symbolen. Im aktuellen Konzept mit 10 ms erreicht man eine Gesamtupdaterate von 100 Hz. D.h., dass 100 mal pro Sekunde das gesamte Netzwerk aktualisiert werden kann. Ein Zeitschlitz besteht aus 11 OFDM-Symbolen: das erste OFDM-Symbol ist eine Präambel, welcher zur initialen Kanalschätzung, der Rahmen- und Zeitsynchronisation, und der Agentenidentifikation verwendet wird. Im zweiten OFDM-Symbol sind mit einem Rate 1/3 Faltungscode der Sendezeitstempel, eine Paketnummer, sowie die Daten für die Rückantworten für die Distanzschätzung kodiert. Die restlichen nachfolgenden neun OFDM-Symbole sind frei belegbar, z.B. für die Kommunikation.



**Abbildung 6.2.1: TDMA-Schema und OFDM-Symbole pro Slot für das kombinierte Kommunikations- und Lokalisierungssystem**

Im Vorgängerprojekt VaMEx wurde ein Experimentalaufbau mit kommerziell verfügbaren Komponenten entwickelt, mit welchem nur ein spezielles Verfahren zur Distanzschätzung zwischen mehreren Agenten gezeigt werden konnte. Die Erweiterung auf mehr als sechs Agenten erwies sich als unvorteilhaft. Zu Beginn von VaMEx-CoSMiC kamen miniaturisierte Software-Defined-Radios (SDRs) auf den Markt. Das eingesetzte SDR hat die Funktion eines Datensamplers zum Senden und Empfangen auf beliebigen HF-Frequenzen. Die komplexwertigen Basisbandsamples werden dabei über eine hochratige Busverbindung zu/von einem Computer übertragen. Die gesamte digitale Signalverarbeitung findet in Software auf dem Computer statt. Im Rahmen von VaMEx-CoSMiC wurde das überarbeitete Konzept zur Kommunikation und Lokalisierung mit Anpassungen an das SDR in einer OpenSource-Realisierung namens Gnuradio implementiert. Die realisierten

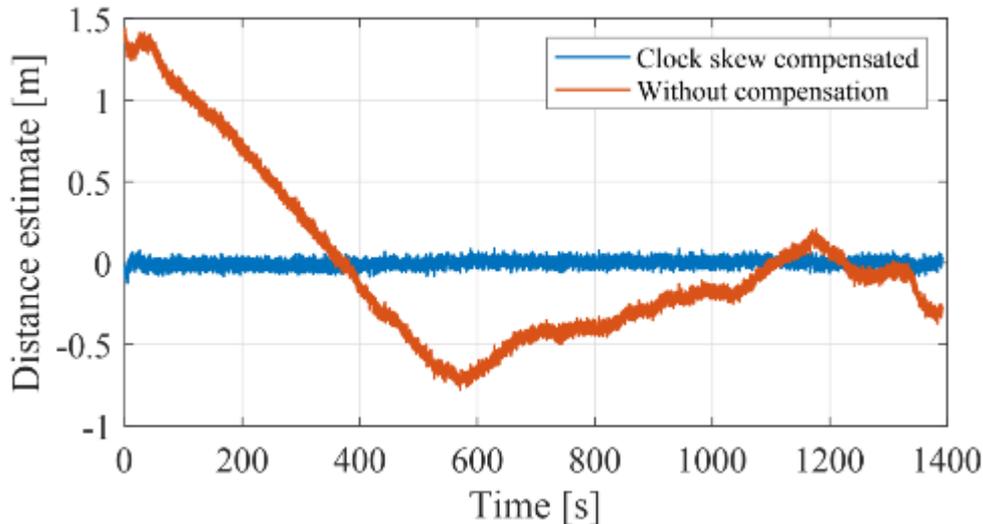
Signalverarbeitungsblöcke beinhalten die OFDM-Rahmensynchronisation, Kanalschätzung, Datendekodierung, und die Distanzschätzung. Als erster Schritt wurde die verteilte Zeitschlitzsynchronisation realisiert und in einem Laborexperiment demonstriert.



**Abbildung 6.2.2: 19"-Aufbau aus dem Vorgängerprojekt VaMEx (nur Distanzschätzung) und eingesetztes SDR in VaMEx-CoSMiC (rechtes Bild, blauer Quader).**

Die Distanzschätzung zwischen zwei oder mehreren Agenten erfolgt über ein Two-Way-Ranging. Zur Parallelisierung der Distanzschätzung, wurde das Datenpaket mit der Rückantwort angepasst: es enthält nicht nur einen Zeitstempel von einem Agenten, sondern sämtliche empfangene und korrekt dekodierte Zeitstempel aller Agenten im letzten Zeitschlitz. Somit lassen sich bei einer Anzahl von  $N$  Agenten mit  $N-1$  Kanalzugriffen alle Distanzen im Netzwerk schätzen, anstatt von  $N(N-1)/2$  Kanalzugriffen. Ein Nachteil des klassischen Two-Way-Rangings ist der Einfluss der Uhrendrift (Skew), wenn die Prozessierungsdauer im antwortenden Agenten groß wird (im Bereich von mehreren Millisekunden). Zur Kompensation der Uhrendrift wurde in jedem Agenten für jeden Agentennachbarn ein Kalmanfilter realisiert, welches anhand der empfangenen Datenpakete, der Sende- und Empfangszeitstempel die relative Drift schätzt. Diese Driftschätzung wird in einem zweiten Schritt bei der Berechnung der Distanz aus den Umlaufzeiten berücksichtigt. Für die Parametrisierung der Kalmanfilter wurden mehrere Uhren unterschiedlicher SDRs vermessen und die notwendigen Parameter aus der Allan-Deviation abgeleitet.

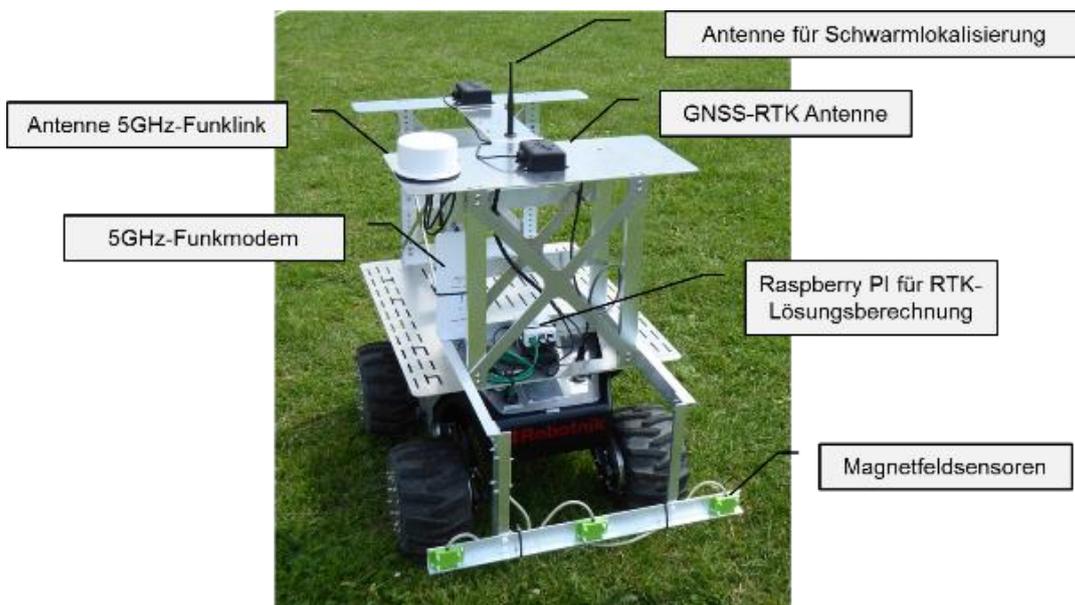
Abbildung 6.2.3 zeigt ein Ergebnis aus einem Laboraufbau. Zwei SDRs wurden über HF-Kabel verbunden und die Distanzschätzung wurde gleichzeitig mit und ohne Uhrendriftschätzung und –kompensation durchgeführt. Zur vereinfachten Darstellung sind beide Kurven mittelwertbefreit, da nur die Änderung der Distanzschätzung über die Zeit von Interesse ist. Die rote Kurve zeigt die Distanzschätzung ohne Uhrendriftkompensation. Man erkennt deutlich, dass über den Verlauf von einigen Minuten die Distanzschätzung über 2 Meter driftet und einem zufälligen Muster folgt. Mit eingeschalteter Kompensation wird der Einfluss der Uhrendrift auf ein Minimum reduziert, siehe die blaue Kurve, und die Distanzschätzung bleibt stabil.



**Abbildung 6.2.3: Ergebnis der Distanzschätzung im Laboraufbau mit und ohne Uhrendriftschätzung/kompensation. Zur vereinfachten Darstellung sind beide Kurven mittelwertbefreit.**

Neben den Signalverarbeitungsblöcken in Gnuradio wurde auch eine Server-Schnittstelle via TCP/IP realisiert. Über diese Schnittstelle kann sich ein Client mit dem SDR eines Agenten verbinden, und erhält automatisch die verfügbaren Distanzschätzungen und Zeitstempel zu den Nachbaragenten. Diese Schnittstelle wurde genutzt, um die Distanzschätzungen über eine weitere Schnittstelle in den Partikelfilter zu übertragen. Der Partikelfilter, siehe Kapitel 6.3.2, schätzt die Relativpositionen der Agenten im Netzwerk basierend auf den Distanzschätzungen.

Zur Hälfte der Laufzeit des Vorhabens ergab sich für DLR-KN die Chance, aus Eigenmitteln sechs idente Rover für Experimente im Außenbereich zu beschaffen. Diese Rover wurden weiter aufgerüstet um eine stabile Kommunikation zu einer gemeinsamen Funkbasis zu ermöglichen und um eine genaue Ortsreferenz via GNSS-RTK in Echtzeit zu erhalten. Der mechanische Aufbau erlaubt flexible Erweiterungen: so wurden unter anderem das SDR-Modul der Schwarmlokalisierung von DLR-KN, sowie die Stereokamera und UWB-Module der TU-München integriert.



**Abbildung 6.2.4: Einer von sechs identen Rovern von DLR-KN für Experimente der Schwarmnavigation und Schwarmexploration.**

Für die Experimente im freien Feld sind eine robuste Kommunikationsverbindung zu den Rovern und Sensoren, sowie eine genaue Ortsreferenz unerlässlich. Daher wurde eine Funkbasisstation für einen autarken Betrieb aufgebaut. Die Kommunikation von der Basis zu den Rovern erfolgt über einen dedizierten 5 GHz Funklink. Den Nahbereich deckt ein WLAN Access-Point ab. Ein Kleinrechner für die Softwareverwaltung, ein LTE-Modem für den Internetzugang, sowie eine Batterie mit Spannungswandler für den autarken Betrieb sind integriert. Abbildung 6.2.5 zeigt den Aufbau der Funkbasisstation. Im Verlauf des Vorhabens wurden die Netzwerkkomponenten der Partner in das bestehende Netz integriert und die GNSS-RTK Korrekturdaten im Netzwerk verteilt.

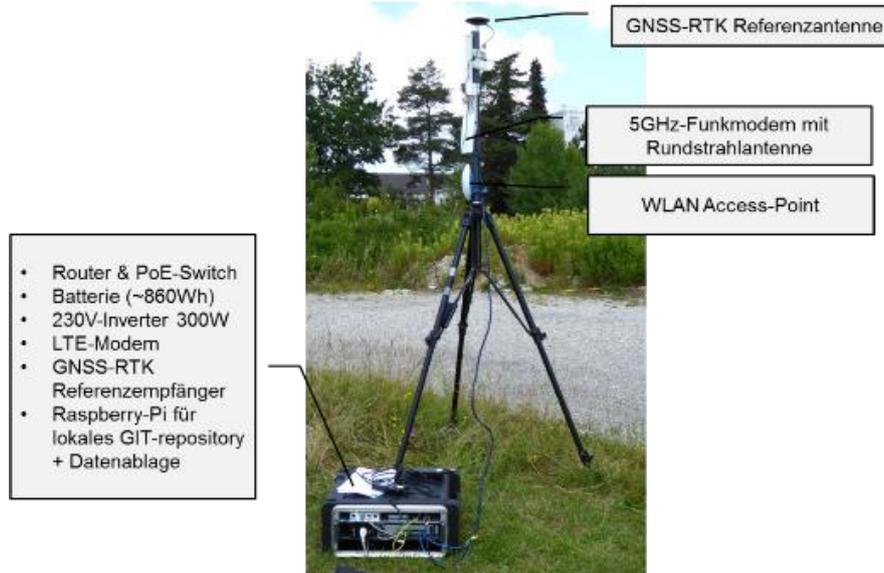


Abbildung 6.2.5: Funkbasisstation von DLR-KN

### 6.2.2 Systemintegration für visuelles SLAM und Sensorfusion

Zur Demonstration des visuellen SLAM- und Sensorfusionsalgorithmus verbesserte TUM-NAV das im Vorgängerprojekt VaMEx entwickelte und in das Schwarm-Rover-System des DLR-KN integrierte Sensorsystem [Abbildung 6.2.6]. Zunächst wurde eine Stereokamera (Bumblebee2, 20FPS, FireWire-Schnittstelle) auf dynamischen Rovern für visuelles SLAM montiert. Um Entfernungsmessungen zwischen Schwarm-Elementen zu erhalten, wurden ein UWB-Tag und Ankermodule (Decawave, USB-Schnittstelle) auf dynamischen und statischen Rovern montiert.

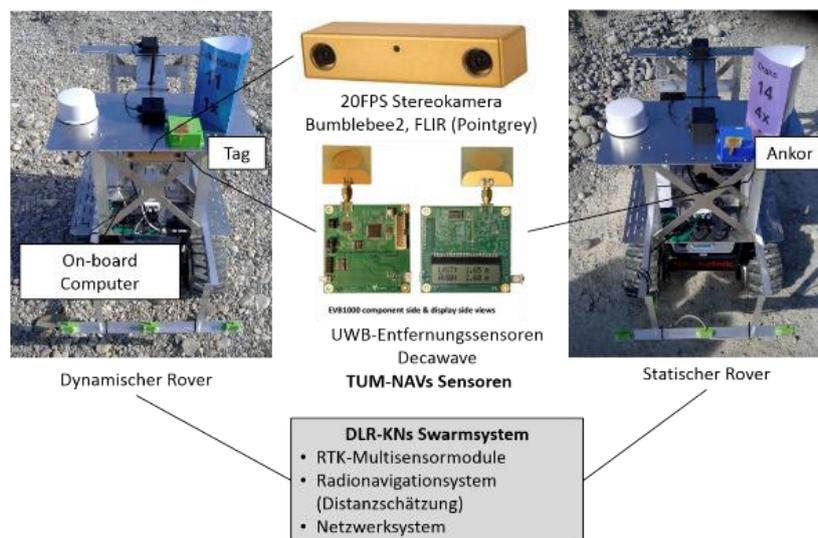
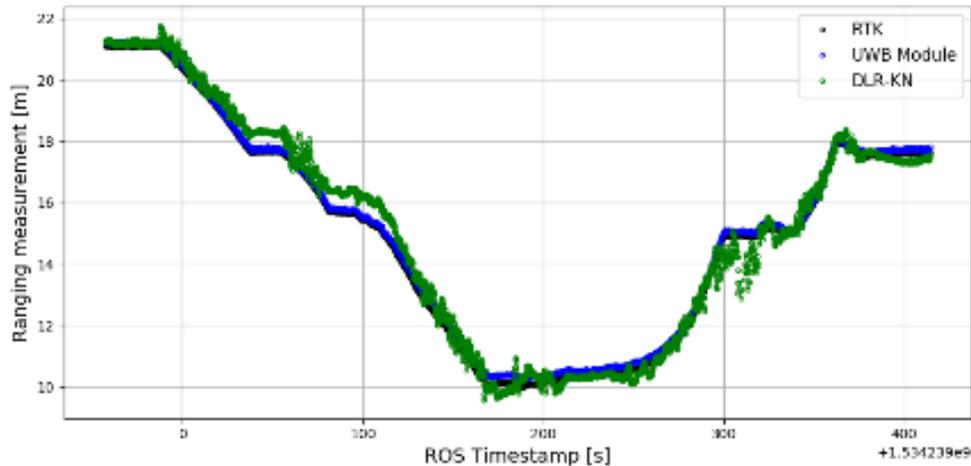


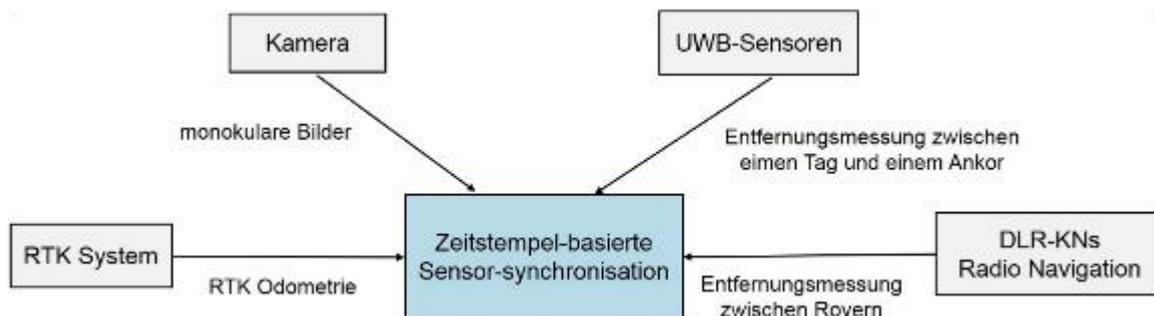
Abbildung 6.2.6: Das integrierte System mit dem TUM-NAV Sensor und dem DLR-IKN Rover

Zusätzlich wurden für die Sensorfusion die Abstandsschätzungen aus dem System des DLR-IKNs oder den mit den ANavS RTK-Lösungen generierten synthetischen Entfernungsmessungen verwendet für den Fall dass die UWB-Module die Messungen nicht durchführen können (z.B. unter Hochtemperaturbedingungen). Die mit jedem System erzielten Messungen sind in [Abbildung 6.2.7] dargestellt.



**Abbildung 6.2.7: Ranging measurements with the ANavS RTK system (synthetic, black); UWB modules (blue); DLR-KN's navigation system (green)**

Als Bordcomputer-Laptop wurde ein Laptop (i7, 16GB RAM) verwendet. Dieser empfing die Bild- und Messdaten von den Sensoren, synchronisierte diese durch Vergleich der Zeitstempel [Abbildung 6.2.8] und führte gleichzeitig die Lokalisierung und Kartierung (SLAM) mit dem Sensor-Fusionsalgorithmus durch. Alle Schätzungen (Fahrzeugposition und -orientierung und Kartenpunkte) können mit der aktuellen Kameraansicht an der Bodenstation per Remote-Computing überwacht werden.



**Abbildung 6.2.8: Datensynchronisierung durch Zeitstempel-Vergleich**

### 6.2.3 Experimentalsysteme automatischer LiDAR SLAM

Für die Versuchsdurchführung wurden die vorhandenen Schwarmteilnehmer modifiziert und in ihrer Leistungsfähigkeit erweitert, um eine automatisierte Exploration mittels LiDAR SLAM durchführen zu können. Die grundlegende Hardwarearchitektur wurde dabei auf fahrenden und fliegenden Schwarmteilnehmern identisch aufgebaut, um eine hohe Softwarekompatibilität zu erreichen.

Basis für die Automatisierung ist eine x86-basierte Rechereinheit, auf der die freie und modulare Robotiksoftware ROS läuft. Als Navigationssystem dienen eine Inertialmesseinheit (IMU), ein Relativortungssystem sowie ein LiDAR. Durch einen über ROS angesteuerten Aktuator kann das LiDAR um die Hochachse gedreht werden und ermöglicht somit eine

Rundumsicht.

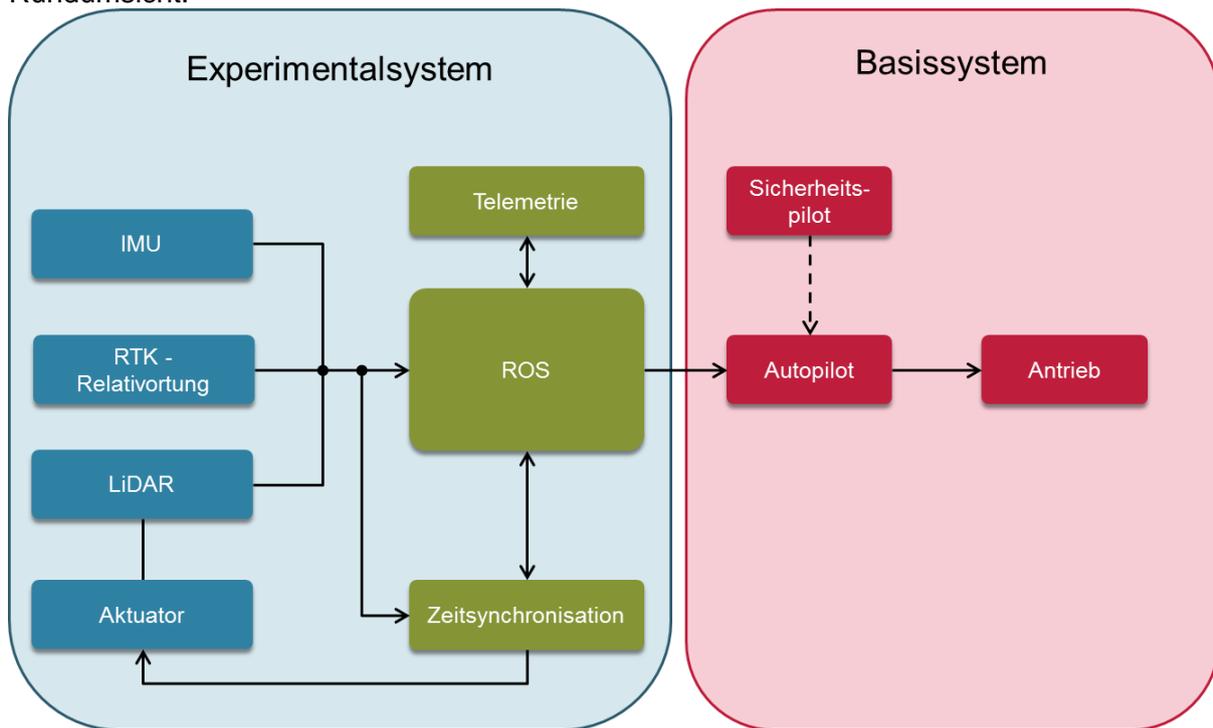


Abbildung 6.2.9: Systemaufbau von TUBS-IFF

Aus Sicherheitsgründen wird der Antrieb nicht direkt Angesteuert, sondern das System in ein Experimental- und ein Basissystem geteilt. Für das Basissystem wurde ein robuster, handelsüblicher Autopilot eingesetzt, der über eine Fernbedienung konventionell von einem Sicherheitspiloten gesteuert werden kann. Somit bleiben die Vehikel auch bei Ausfall oder Fehlfunktion des Experimentalsystems manuell steuerbar.

Zur automatischen Exploration werden im Experimentalsystem Wegpunktkommandos von ROS an den Autopiloten weitergegeben. Eine Relativortung liefert das AnavS RTK-System.

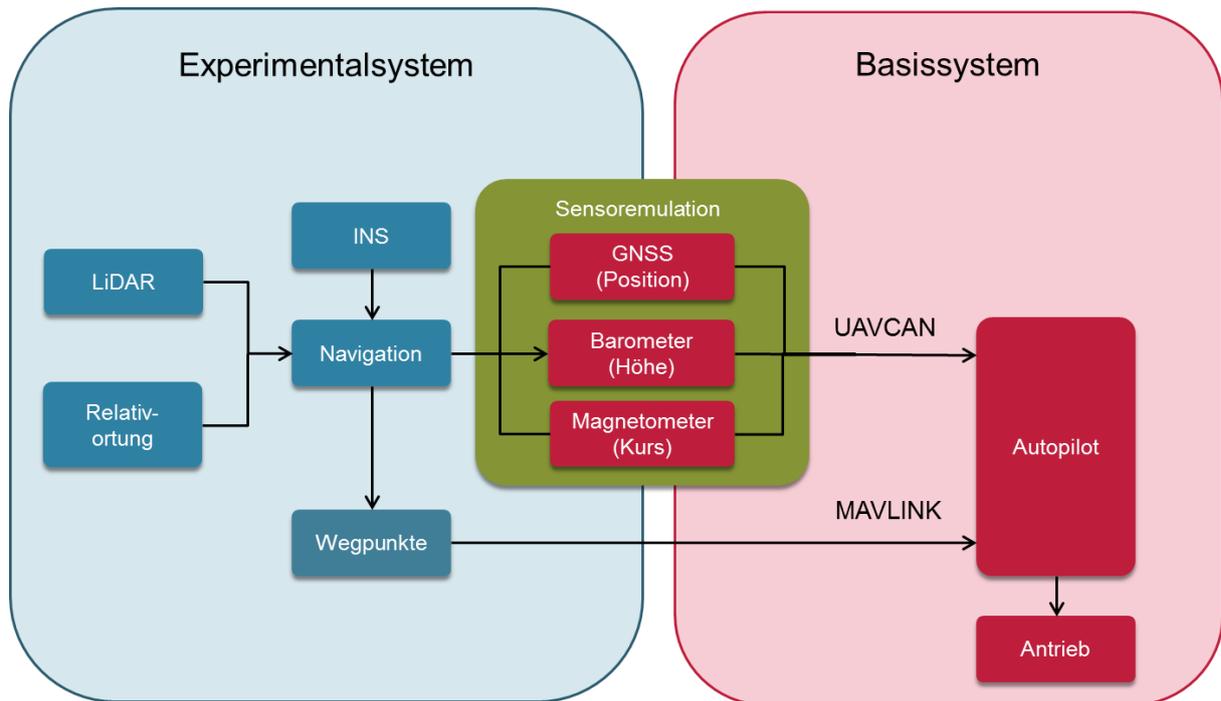
Als SLAM-Sensorik wurde der Hokuyo LiDAR weiterverwendet. Der Schwenkantrieb, um 3D Scans zu erhalten, wurde durch einen Aktuator mit Positionsrückmeldung ersetzt.

Daten der Inertialmesseinheit ADIS 16488 werden von einem echtzeitfähigen Microcontroller System eingelesen. Zur Zeitsynchronisation der Messdaten werden im Microcontroller basierten Zeitsynchronisationssystem die Zeitpulse von LiDAR, RTK-System und IMU erfasst. Zusätzlich wird vom Microcontroller auch das Servo in Echtzeit angesteuert und die Positionsrückmeldung ausgelesen. Durch Laufzeitmessung auf der USB-Verbindung zwischen Microcontroller und ROS lassen sich die Daten auf etwa 200us genau mit der Systemzeit von ROS synchronisieren. Auftretende Latenzen gepufferter Daten aus dem USB-System können dadurch im überarbeiteten System direkt erkannt werden.

### Sensorstützung des Basissystems

Durch die interne Inertialmesseinheit im Autopilotensystem ist eine stabilisierte Bewegung gewährleistet. Der Autopilot des Basissystems ist darauf ausgelegt, Stützinformationen von GNSS, Magnetometer und Barometer zu verwenden, um neben der stabilisierten Bewegung auch navigieren zu können. Dazu lässt sich das Basissystem über absolute Wegpunkte kommandieren. Geschieht dies vom Experimentalsystem aus, muss das Basissystem mit den gleichen Stützinformationen wie das Experimentalsystem arbeiten. Da im Experiment SLAM-basiert navigiert werden soll, was sich je nach Güte von einer GNSS-Lösung unterscheiden kann, kann es bei der Wegpunktnavigation zu Unterschieden zwischen Experimental- und Basissystem kommen, welche in falschen Bewegungsrichtungen resultieren. Um diesem Problem zu begegnen, werden von der Navigationslösung im Experimentalsystem die

benötigten Stützgrößen für den Autopiloten berechnet. Neben verschiedenen Punkt zu Punkt Verbindungen unterstützt der Autopilot auch UAVCAN als Bussystem.



**Abbildung 6.2.10: Sensoremulation zur Stützung des Autopiloten**

Hier können über eine einzige physikalische Schnittstelle die benötigten emulierten Sensordaten zur Stützung an den Autopiloten übertragen werden. Die Position wird durch eine GNSS-Nachricht mit entsprechender geographischen Länge und Breite emuliert. Die Höhenstützung erfolgt durch Emulation eines Barometers mit der entsprechenden Nachricht des statischen Druckes nach dem Modell der Standardatmosphäre. Die Kursstützung erfolgt über Nachbildung der dem Erdmagnetfeld entsprechenden Magnetometerdaten. Durch diesen Aufbau werden die Wegpunktkommandos des Experimentalsystems konsistent ausgeführt. Bei einem Absturz des Experimentalsystems, fehlerhafter Wegpunkte oder Stützinformationen kann mit dem weiterhin stabilisierten Autopiloten manuell sicher gelandet werden.

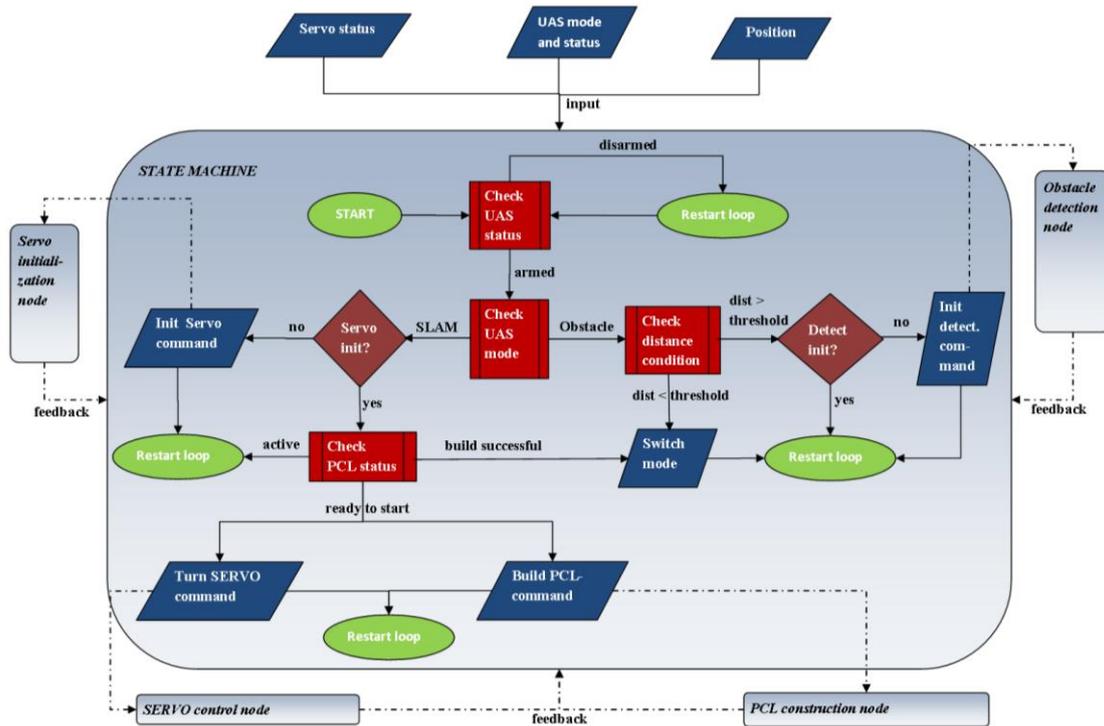
## 6.3 Ortung

### 6.3.1 Automatisierungen für LiDAR SLAM

Zur Realisierung einer vollständig automatisierten Mars-Exploration, als speziellen Anwendungsfall eines Navigationsszenarios, ist es notwendig alle relevanten Subprozesse ebenfalls zu automatisieren. Im untersuchten Einsatzszenario, unter Verwendung eines LiDAR und dessen Datenverarbeitung zur Realisierung eines SLAM zur Stützung der INS-Ortungslösung, wurde sowohl eine Automatisierung der Datenerfassung, der Datenverarbeitung als auch des Navigationsprozesses, welcher das Führen und die Ortsbestimmung des Systems umfasst, notwendig.

Die Automatisierung wurde mit Hilfe einer plattformunabhängigen Ablaufsteuerung, im folgenden auch State-Machine genannt, umgesetzt, die die Bewegungsabläufe des robotischen Systems, die Datengenerierung durch das LiDAR-Servo-Modul und die softwareseitige Datenverarbeitung durch den LiDAR-SLAM überwacht und steuert.

Der Ablauf für eine Explorationsmission und die damit verbundenen Schritte der State-Machine zur Automatisierung des LiDAR SLAM sind im folgenden Schaubild dargestellt.



**Abbildung 6.3.1: TUBS-IFF State-Machine zur Automatisierung der robotischen Systeme**

Zu Beginn der Mission wird zunächst das inertielle Navigationssystem (INS) initialisiert und mit einer Positionslösung versorgt. Die Generierung dieser globalen und initialen Positionslösung wird im Rahmen des Vorhabens VaMEx - LAOLa auf Basis eines Systems, bestehend aus Sternensensoren und Funkbarken realisiert, wobei letztere zur Ortsbestimmung der robotischen Systeme in dem durch die Sternensensoren aufgespannten Koordinatensystem dienen.

Nach der Initialisierung des INS erfolgt noch vor der ersten Bewegung ein initialer LiDAR Scan der Umgebung. Das Resultat eines jeden Scans ist eine Punktwolke. Diese erste Punktwolke bildet die Grundlage für die Verarbeitung nachfolgender, mit Hilfe des LiDAR-Servo-Moduls erzeugter Punktwolken durch den SLAM Algorithmus. Durch die erste Punktwolke wird somit der SLAM - Algorithmus initialisiert. Die Initialisierung kann auch auf Basis einer zuvor in den SLAM - Algorithmus eingespeisten Karte erfolgen. Befindet sich die Position des Systems innerhalb dieser Karte, kann die erste erzeugte Punktwolke durch den initialen Scan bereits zur Ortsbestimmung innerhalb der vorhandenen Karte herangezogen werden. Liegt die Karte im Koordinatensystem des durch die Sternensensoren aufgespannten Systems vor, so ist die durch den SLAM erhaltene Positionslösung im Idealfall, unter Ausschluß von Messfehlern des LiDAR, lokalen Minima im SLAM Prozess und Doppeldeutigkeiten beim Map-Matching, redundant.

Nach erfolgreicher Initialisierung der beiden Navigationssysteme versetzt die State-Machine das robotische System in den Fortbewegungsmodus. In diesem Modus kann das System entlang eines geplanten und durch Wegpunkte abstrahierten Pfades die Verfolgung seines aktuellen Explorationszielpunktes fortsetzen.

Die Unterbrechung des Fortbewegungsmodus zur Initiierung der Datenerfassung durch den LiDAR, der sich daran anschließenden Datenverarbeitung und SLAM erfolgt durch die State-Machine in Abhängigkeit von unterschiedlichen Kriterien. Diese Kriterien sind zeitabhängig, entfernungsabhängig oder aber abhängig von der Prozesskovarianz des Algorithmus zur Sensordatenfusion, welche durch einen Extended Kalman-Filter umgesetzt wurde. Durch Überschreiten eines oder mehrerer Schwellwerte der zuvor genannten Kriterien wird der Fortbewegungsmodus unterbrochen und die Datenerfassung des LiDAR zur Generierung einer Punktwolke eingeleitet. Sobald diese erzeugt, die Verarbeitung durch den SLAM

vollzogen ist, also eine Positionslösung durch das SLAM - Verfahren vorliegt, und die Sensordatenfusion vollzogen ist, wird das robotische System zurück in den Fortbewegungsmodus versetzt. Zeitgleich erfolgt eine Aktualisierung der Werte der Kriterien zur Unterbrechung des Fortbewegungsmodus. Das heißt die während der Fortbewegungsphase akkumulierten Werte für die entfernungs- und zeitabhängigen Kriterien werden durch die State-Machine zurück auf Null gesetzt. Ein Sonderfall ist das Kriterium der Prozesskovarianz. Diese ist ein Resultat des Fusionsfilters und wird nicht von der State-Machine manipuliert. Folglich kann die Prozesskovarianz des Extended Kalman-Filter noch Werte enthalten die auch nach dem Einspeisen des Mess-Updates in Form des SLAM-Resultates über dem Schwellwert liegen. Folglich wird sofort eine neue LiDAR-Datenerfassung eingeleitet. Sollten die Ergebnis des SLAM wiederholt in ein unzureichendes Update der Prozesskovarianz führen, kann dieser Zyklus nur durch ein adäquates Mess-Update aus einer anderen Quelle als dem SLAM, beispielsweise aus den Daten des Relativortungssystems oder aber der Funkbarken, durchbrochen werden.

Letztendlich werden die durch die Mechanismen der State-Machine generierten Punktwolken aus den Daten des LiDAR/Servo - Moduls und die Ergebnisse des SLAM Algorithmus gespeichert. Dadurch stehen diese auch für eine eventuelle manuelle Nachverarbeitung zur Verfügung.

Weiterhin wird das Ergebnis des SLAM, bestehend aus einer Punktwolken und der Ortungslösung des Systems zur kontinuierlichen Erweiterung der Kartendatenbank genutzt.

### 6.3.2 Visuelles SLAM und Distanzschätzungsfusion

Für die Selbstlokalisierung und Kartenerstellung von Fahrzeugen in unbekanntem Umgebungen (*Simultaneous Localization and Mapping*, SLAM) implementierte TUM-NAV einen Datenfusionsalgorithmus mit monokularem Sehen und Entfernungsmessung. Zunächst wird eine merkmalsbasierte visuelle SLAM-Methode angewendet, um die Posen des Rovers (Positionen und Orientierungen) zu schätzen und Kartenpunkte in Echtzeit zu erstellen. Um sowohl eine hohe Geschwindigkeit als auch eine hohe Genauigkeit bei verrauschten Bildern zu erreichen, wurde der „*Oriented FAST und Rotated BRIEF*“ (ORB) Detektor eingesetzt. FAST (*Features from Accelerated Segment Test*) und BRIEF (*Binary Robust Invariant Elementary Feature*) sind skalierungs- und rotationsinvariante Detektoren mit hoher Rechengeschwindigkeit. Der ORB-Detektor extrahiert die anfänglichen Merkmale mit der FAST-Methode, und der BRIEF-Algorithmus wird für die Rotationsinvarianz angewendet. Darüber hinaus werden Multiskalenmerkmale mit Pyramidenschichten erzeugt. Abbildung 6.3.2 zeigt ein Beispielbild (EuRoC da-taset, V101 easy) mit erkannten ORB-Merkmalen.

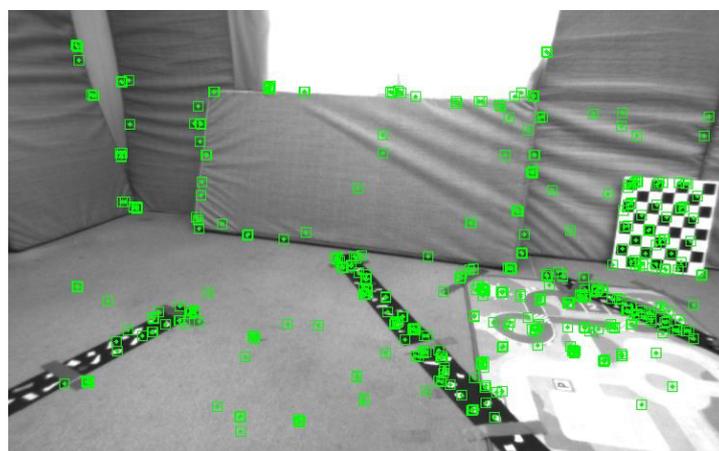
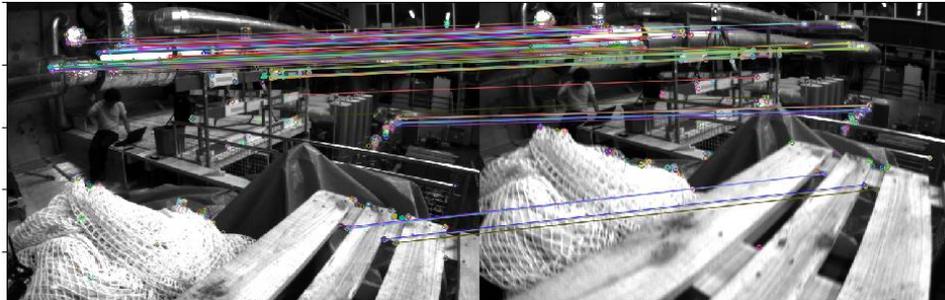


Abbildung 6.3.2: Ein Bild-Beispiel mit detektierten ORB Merkmalen

Nach der Merkmalsextraktion werden aufeinanderfolgende Bilder verglichen, um anhand der Deskriptoren der Bilder (z.B. Bildintensität, Intensitätsgradient) passende Merkmale zu finden. Ein Beispiel für korrespondierende Merkmale ist in Abbildung 6.3.3 dargestellt. Die verbundenen Merkmale dienen der kontinuierlichen Verfolgung der aktuellen Fahrzeugposition sowie der lokalen Kartenverfeinerung.

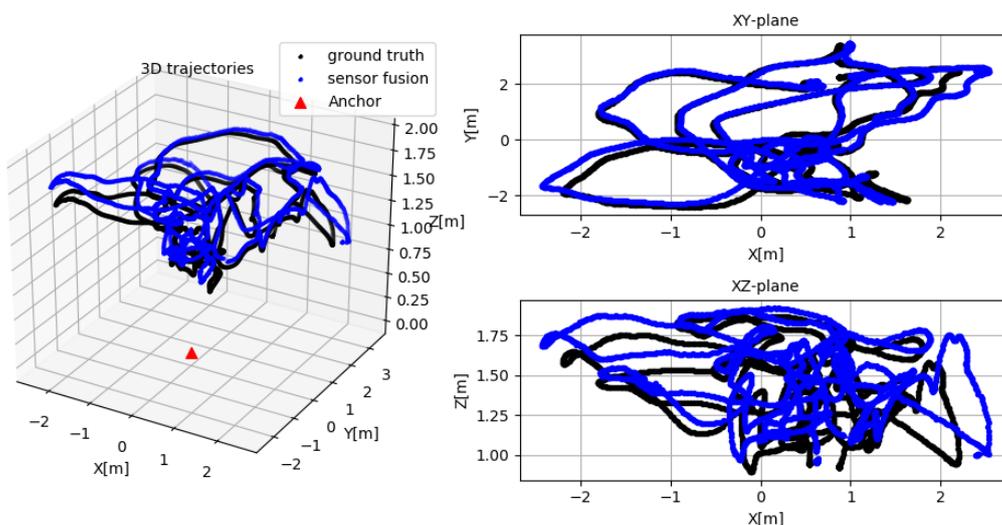


**Abbildung 6.3.3: Ein Beispiel für die Merkmalszuordnung in aufeinanderfolgenden Bildern**

Darüber hinaus wird die „Loop Closing“ Technik angewendet, um Driften im Dead-Reckoning-Prozess zu minimieren. Dazu werden die Übereinstimmungen zwischen den im aktuellen Bild beobachteten Merkmalen und den in der globalen Kartendatenbank gespeicherten Merkmalen gesucht. Wenn eine Wiedererkennung detektiert wird, werden die akkumulierten Fehler der Kameraposen durch eine Ähnlichkeitstransformation bestimmt und die Posen mit der Methode der kleinsten Fehlerquadrate korrigiert.

Im nächsten Schritt wird die mehrdeutige, globale Skala der monokularen reinen Bildkarte wiederhergestellt, indem die Entfernungsmessungen zwischen dynamischen und statischen Rovern fusioniert werden. Darüber hinaus werden, wenn kein Loop Closing verfügbar ist, Abweichungen in reinen Bildverarbeitungsschätzungen durch ein zusätzliches globale Least-Squares Verfahren reduziert, wodurch sowohl Wiedergabefehler in den Bildrahmen als auch Entfernungsfehler minimiert werden.

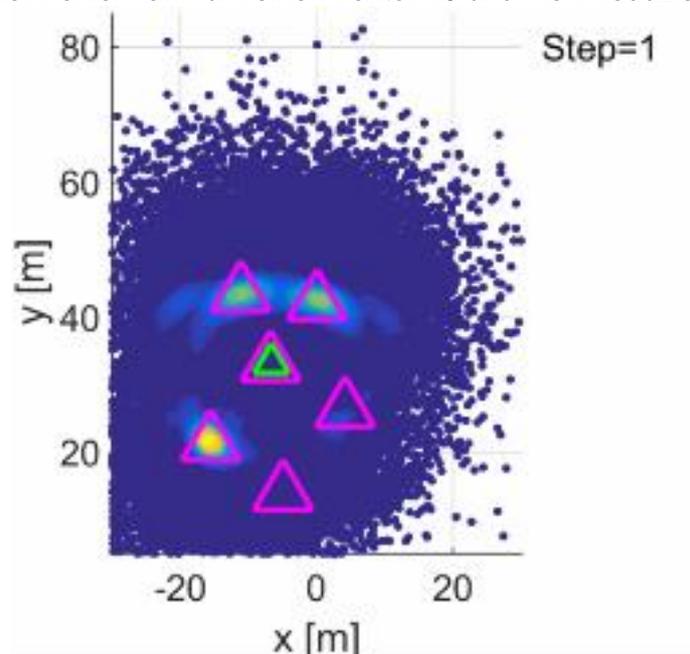
TUM-NAV demonstrierte den Fusionsalgorithmus mit der Sequenz V101 des EuRoC-Datensatzes, bei dem es sich um einen frei-verfügbaren Bilddatensatz handelt, der mit Hilfe eines MAV in verschiedenen Innenräumen gewonnen wurde. Die Positionsreferenz des EuRoC-Datensatzes wurde mit einem Vicon 6D-Bewegungserfassungssystem ermittelt. Wie in Abbildung 6.3.4 gezeigt, liefert das visuelle und Entfernungsmessungs-basierte System genaue Trajektorien-Schätzungen, mit einem RMSE von 0,21 m.



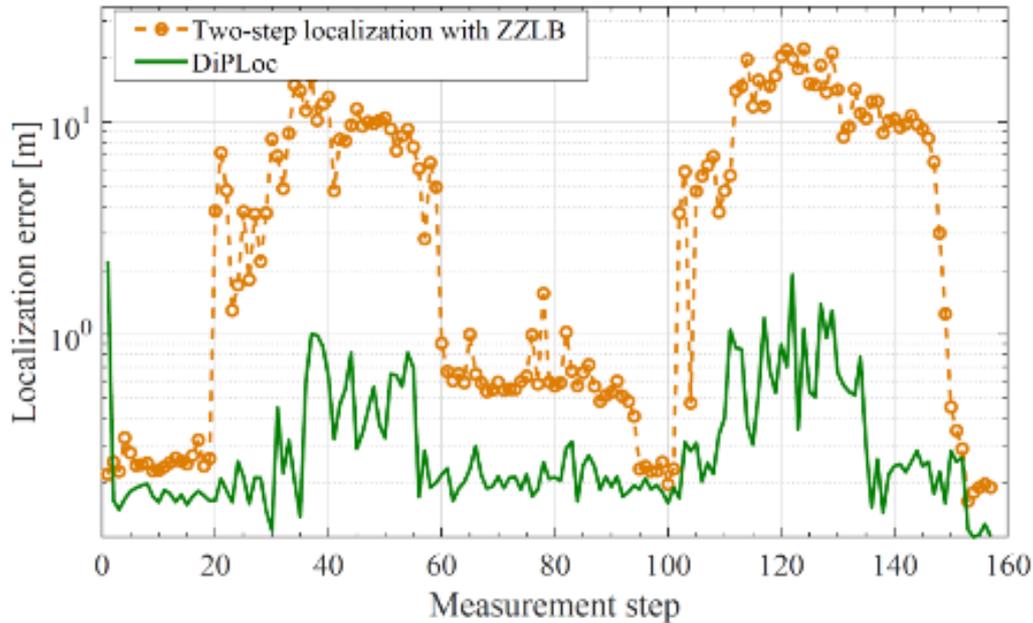
**Abbildung 6.3.4: Schätzung der Trajektorie mit der TUM-NAV-basierten Sensorfusion (blau) und Vergleich mit der Referenzlösung (schwarz)**

### 6.3.4 Relativpositionierung

Am Ende des Vorgängerprojekts VaMEx stellte DLR-KN ein Konzept für ein neues Verfahren zur Positionsschätzung vor. Dieses Konzept namens Direct Particle Filter Localization (DiPLoc) wurde zu Beginn von VaMEx-CoSMiC weiter untersucht. Im Allgemeinen wird ein sequentieller Bayes'scher Filter realisiert als Partikelfilter für die Positionsschätzung verwendet. Die Positionsschätzung selbst kann als Fusionierung von verschiedenen Distanzschätzungen zwischen Agenten im Netzwerk betrachtet werden. Für eine optimale Fusionierung muss die Güte der Distanzschätzung, z.B. die Varianz, der Bias, etc., bekannt sein bzw. wird diese mitgeschätzt um über geeignete Modelle, z.B. untere Schätzschranken, berücksichtigt zu werden. Dieser mehrstufige Ansatz hat einen entscheidenden Nachteil: die Güte ist eine schlechte Approximation der tatsächlichen Fehler, da z.B. sehr ungünstige Empfangsbedingungen (sehr geringes SNR), sowie Distanzschätzfehler verursacht durch starke Mehrwegeausbreitung (Multipath) nicht korrekt abgebildet werden können. Das neue Verfahren DiPLoc kürzt dieses mehrstufige Verfahren ab: das gesamte Empfangssignal vor dem Korrelator wird direkt zur Lokalisierung herangezogen. Die Komplexität von DiPLoc ist in derselben Größenordnung wie mehrstufige Verfahren Stand der Technik, mit einem entscheidenden Vorteil: durch die Nutzung des gesamten Empfangssignals (der Waveform) sind sämtliche Hypothesen enthalten und werden im Partikelfilter durch entsprechende Gewichtung automatisch aufgelöst. Dieser Ansatz ist in etwa vergleichbar mit Hard- und Soft-decision Dekodierung bei Kommunikationsempfänger, wobei DiPLoc dem Soft-decision Dekodierer entspräche. Letzterer hat eine deutlich bessere Performance. Abbildung 6.3.5 zeigt den ersten Zeitschritt von DiPLoc basierend auf den Messdaten aus dem Vorgängerprojekt VaMEx. Abbildung 6.3.6 zeigt den Lokalisierungsfehler über mehrere Zeitschritte, in welchen ein Rover eine Trajektorie fuhr. Der Ansatz Stand der Technik mit der Modellierung des Distanzschätzfehlers über eine Ziv-Zakai untere Schätzschranke (ZZLB) zeigt einen hohen Lokalisierungsfehler von mehrere Metern in Bereichen mit sehr ungünstigen Ausbreitungsbedingungen für das Funksignal. Gerade in diesen Bereichen verbessert DiPLoc die Lokalisierung: der Fehler kann um einen Faktor 10 und mehr reduziert werden.



**Abbildung 6.3.5: Erster Zeitschritt von DiPLoc: Punkte zeigen die Hypothesen des Partikelfilters (blau geringe Hypothese, gelb/rot starke Hypothese). Pinke Dreiecke zeigen die tatsächliche Position und das grüne Dreieck die Schätzung.**

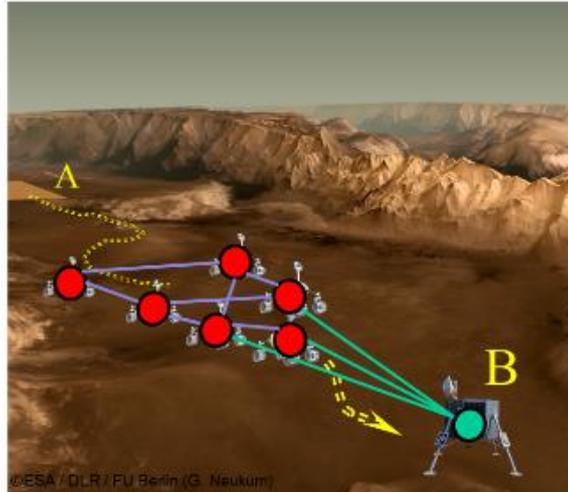


**Abbildung 6.3.6: Vergleich der Lokalisierungsfehler zwischen DiPLoc und dem modellbasierten Ansatz basierend auf Messdaten aus VaMEx.**

Der beiden Verfahren zur Relativpositionierung wurden in einem JAVA-Framework umgesetzt. Dieses Framework besitzt eine Schnittstelle zu den SDRs der Agenten um die Distanzschätzungen zu erhalten. In VaMEx-CoSMiC wurde der mehrstufige Ansatz für die erste praktische Umsetzung gewählt. Diese Realisierung erlaubt eine Schätzung der Positionen und eine grafische Ausgabe in Echtzeit. Weiterhin wurde eine Schnittstelle für das GNSS-RTK realisiert, um direkt den Lokalisierungsfehler berechnen und darstellen zu können. Dabei wird davon ausgegangen, dass das GNSS-RTK im freien Feld als Referenz betrieben wird.

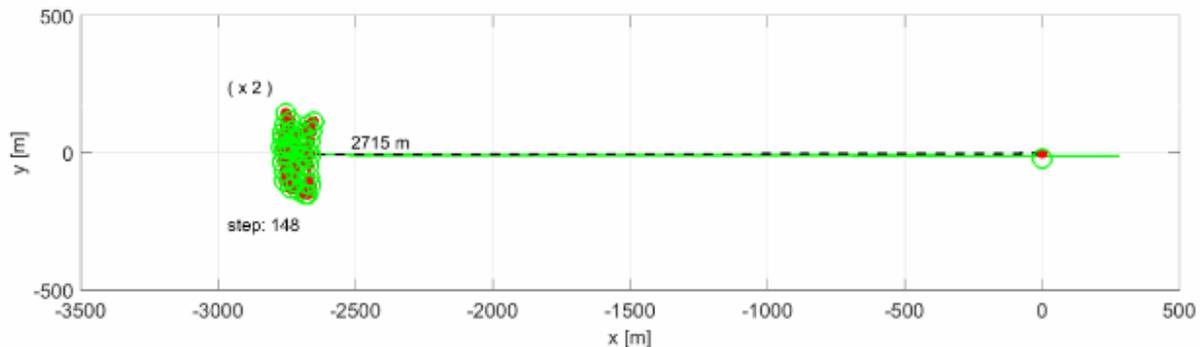
Die Güte der Relativpositionierung ist nicht nur von der Güte der Distanzschätzungen und der Anzahl der Nachbaragenten abhängig, sondern auch von der geometrischen Anordnung des Schwarms (GDOP). Die Robotersteuerung wurde mathematisch mit der Relativpositionierung verknüpft. Daraus ergibt sich eine Navigation für verschiedene Anwendungsfälle: der Schwarm kann selbstständig optimale Konstellationen (Geometrien) zur optimalen Selbstlokalisierung oder optimale Sensor-Aperturen einnehmen. Hierbei wurde eine Anwendung näher untersucht: die Return-to-Base Navigation, siehe Abbildung 6.3.7. Ein Schwarm ist sehr weit von der ursprünglichen Basis (Ort B) entfernt. Die Funksignale der Schwarmlokalisierung können nicht von der Basis empfangen werden. Daher sendet die Basis ein niederfrequentes Radiosignal aus, mit welchem eine deutlich größere Distanz überbrückt werden kann. Der Schwarm empfängt dieses Radiosignal und nutzt es für die Rückführung zur Basis. Dabei muss der Schwarm drei nicht-konvexe Optimierungsprobleme lösen, welche zum Teil sogar gegensätzlich sind:

1. Der Schwarm soll eine optimale Konstellation zur Selbstlokalisierung einnehmen.
2. Der Einfallswinkel des niederfrequenten Radiosignals muss über eine verteilte Apertur (verteilttes Antennenarray) geschätzt werden. Hierzu muss der Schwarm selbstständig eine Konstellation einnehmen, welche für die Schätzung des Einfallswinkels optimal ist.
3. Alle Agenten sollen in einer endlichen Zeit an der Basis ankommen.



**Abbildung 6.3.7: Return-to-Base Navigation für einen Schwarm.**

Die Abbildung 6.3.8 zeigt die Schwarmkonstellation nach wenigen Zeitschritten. Für die Winkelschätzung des niederfrequenten Radiosignals wäre es optimal, alle Agenten möglichst weit entlang der Y-Achse zu verteilen. Diese Anordnung ist jedoch ungünstig für die Selbstlokalisierung da sie zu einem sehr schlechten GDOP führt. Daher nimmt der Schwarm die Form eines linearen Antennenarrays mit Lattice-Struktur an, um sowohl die Selbstlokalisierungsgenauigkeit, als auch die Winkelschätzungsgenauigkeit zu optimieren.



**Abbildung 6.3.8: Return-to-Base Navigation: der Schwarm (tatsächliche Position als rote Punkte links) spannt eine optimale Konstellation für die Selbstlokalisierung und die Winkelschätzung für das niederfrequente Radiosignal auf. Die Basis ist als roter Punkt rechts eingezeichnet.**

Die in VaMEx-CoSMiC theoretisch erarbeitete Relativpositionierung setzt keine sogenannten Ankerknoten voraus. Man spricht daher von Anchor-Free-Localization. Ankerknoten sind Agenten/Sensoren im Netzwerk mit bekannter Position zur Fixierung des Koordinatensystems und zur Transformation des lokalen Koordinatensystems in ein Übergeordnetes.

Bei der Anchor-Free-Localization treten zwei Probleme auf, welche in einem ersten Schritt theoretisch untersucht wurden. Erstens existiert kein gemeinsames Koordinatensystem, sodass ein Konsensus um Netzwerk zwischen allen Agenten gefunden werden muss. Zweitens führen das Fehlen von Ankerknoten zur einer residualen Translation, Rotation, und auch Flipping der Gesamtkonstellation. Für eine robuste Schwarmnavigation (Einbeziehung der Steuerung) müssen diese zwei Probleme gelöst werden. Eine erste theoretisch untersuchte Lösung verwendet einen Konsens-basierten Ansatz, welcher verteilt im Netzwerk berechnet wird. Dabei konnte gezeigt werden, dass die residuale Translation und Rotation minimiert wird, und ein gemeinsames Koordinatensystem für die Navigation gefunden werden kann.

## 6.4 Kartierungen

### 6.4.1 Infrastruktur der Kartendatenbank

Wie bereits unter 6.3.1 beschrieben, werden durch die primären umgebungserfassenden Sensoren, den LiDAR und die Kameras, dreidimensionale Punktwolken erzeugt. Diese werden auf den jeweiligen sensortragenden Systemen nach der Vollführung des SLAM gespeichert. Das Resultat eines jeden SLAM, die Positionsbestimmung und die um die Punktwolke erweiterte Karten, werden ebenfalls auf dem System hinterlegt und können durch die anderen Schwarmteilnehmer oder durch einen anderen Nutzer, welcher Zugang zum Kommunikationsnetzwerk des Schwarms hat, abgefragt werden. Zwischen den Schwarmteilnehmer kann ein Zugriff auf die Karten in Form der Punktwolken sowohl durch das ROS-Kommunikationsnetzwerk als auch über das File Transfer Protocol (FTP) erfolgen. Letztere Variante stellt die einzige Option für Nutzer dar, die nicht Mitglied des Schwarms sind.

Die aus den dreidimensionalen Scans des LiDAR bzw. der Prozessierung der Kamerabilder generierten Punktwolken werden durch entsprechende SLAMs auch zur Kartierung genutzt. Das Ergebnis der Kartierung ist ebenfalls eine Punktwolke, die sich aus den einzelnen, durch den SLAM verarbeiteten Punktwolken zusammensetzt. Das Ergebnis der Kartierung durch die SLAM Verfahren erfolgt zyklisch, nach jedem erfolgreich durchgeführten SLAM in dem sogenannten Point Cloud Data (PCD) Dateiformat.

Die Datenbank besteht somit also aus der globalen Punktwolke als Ergebnis der Kartierung und den zyklisch erzeugten Punktwolken an den durch den Fusionsfilter bestimmten Position ihrer Generierung.

Letztere Punktwolken werden zur Erzeugung einer OctoMap [1] herangezogen, welche die Punktwolken abstrahiert und sowohl zu Navigationszwecken als auch taktischen Nahfeldnavigation herangezogen wird. Diese werden ebenfalls zyklisch gespeichert und stehen sowohl als Binary Tree als auch als Octree Dateiformat zur Verfügung.

Die Auswahl zugunsten einer OctoMap fiel auf Basis folgender recherchierten Eigenschaften:

- aktualisierbar
- erweiterbar
- speichereffizient
- kurze Zugriffszeiten
- modifizierbar zur Aufnahme von Metainformationen

Als Schnittstellen zwischen den Systemen und den einzelnen Programmen wird auf die ROS Infrastruktur zurückgegriffen. Dieses stellt bereits ein Kommunikationsinfrastruktur und kompatible Schnittstellendefinitionen für den Datentransport zwischen den einzelnen Programmen zur Verfügung. Unter Verwendung der vorhandenen ROS Schnittstellendefinitionen, den sogenannten ROS-Nachrichtentypen, sowie des ROS-Kommunikationsnetzwerkes konnte der Daten- und Informationsaustausch sichergestellt und ein modulares Softwareframework, bestehend aus einzelnen Programmen, zur Verarbeitung und Zugriff auf alle zur Erstellung der Karten relevanten Informationen realisiert werden.

### 6.4.2 Kartenfusion

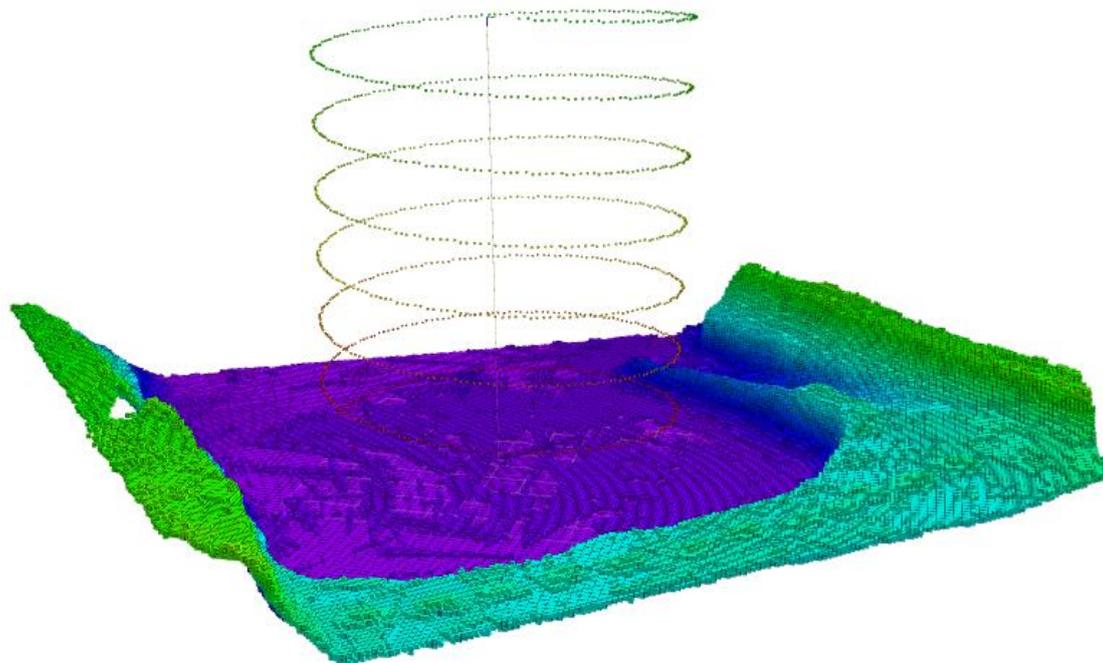
Durch das bereits im Abschnitt zur Infrastruktur der Kartendatenbank beschriebenen Kommunikationsnetzwerk ermöglicht bei einer bestehenden Verbindung zum jeweiligen Schwarmteilnehmer die Zugänglichkeit zu den gespeicherten Kartendaten in Form von Punktwolken oder den OctoMaps. Je nach Anwendungsfall kann somit auf die lokalen Kartendaten eines jeden Schwarmteilnehmers nicht nur zugegriffen sondern auch auf das anfragende System transferiert werden. Diese bevorzugte Variante liegt aber in dem automatisierten Transfer der Kartendaten bei jeder Aktualisierung der Umgebungsinformationen.

Durch die Kompatibilität der zur Kartierung herangezogenen Sensorinformationen, die alle als Punktwolken vorliegen, und des globalen und des über den Schwarm einheitlichen Koordinatensystems kann eine Fusion der lokalen Karten zu einer globalen Karte vollzogen werden. Eine Gewichtung der Daten erfolgt im Verhältnis 1:1. Das heißt die Daten aus unterschiedlichen bildgebenden Sensoren fließen gleichberechtigt in die Karte ein. Durch eine Zeitstempelung der Daten kann jedoch eine Gewichtung bezüglich der Aktualität erfolgen.

Zur Reduktion der übertragenen Datenmengen besteht die Option die Kartendaten in abstrahierter Form als OctoMap zu übertragen. Das die Kartendaten anfragende Schwarmteilnehmers kann dabei den Grad der Abstrahierung durch Vorgabe der Voxelgröße (Voxel: VOlumetric piXEL) beeinflussen.

Durch Ableitung der bestehenden OctoMap Klasse konnte eine neue und eigene Klasse abgeleitet werden, welche neben den Belegungswahrscheinlichkeiten eines Voxels auch Metainformationen aufnehmen kann. Die erweiterte bzw. abgeleitete Klasse kann Metainformationen wie Temperatur, Druck und Feuchte aufnehmen. Weiterhin wurde ein entsprechendes Plugin entwickelt um die Daten auch in der ROS- eigenen Visualisierung, namentlich RVIZ, darzustellen.

In der nachfolgenden Abbildung ist ein spiralförmiger Aufstieg eines Fluggeräts abgebildet.



**Abbildung 6.4.1: Darstellung der Metainformation „Luftdruck“ aus Daten eines spiralförmigen Aufstiegs eines UAS und Belegungswahrscheinlichkeiten in einer gemeinsamen OctoMap**

Während des Aufstiegs wurden kontinuierlich Messungen des Luftdrucks unternommen und in die bereits bestehende, durch Voxel abstrahierte Karte, eingepflegt. Sowohl die Erfassung der Umgebung mit einem durch ein LiDAR ausgerüstetes UAS als auch der spiralförmige Aufstieg wurden in Gazebo simuliert. Die Visualisierung der Daten erfolgte unter Zuhilfenahme von RVIZ. Basierend auf den bereits implementierten Änderungen zur Speicherung von Metainformationen wie Luftdruck, Temperatur und Feuchte können weitere Anpassungen an der abgeleiteten OctoMap Klasse vorgenommen werden um zusätzliche Metainformationen wie beispielsweise die Steigung oder die Bodenbeschaffenheit mit in die Karte aufzunehmen.

### 6.4.3 Aufbereitung und Datenbankabfragen

Das zwischen den Schwarmteilnehmern aufgespannte Kommunikationsnetzwerk bietet eine

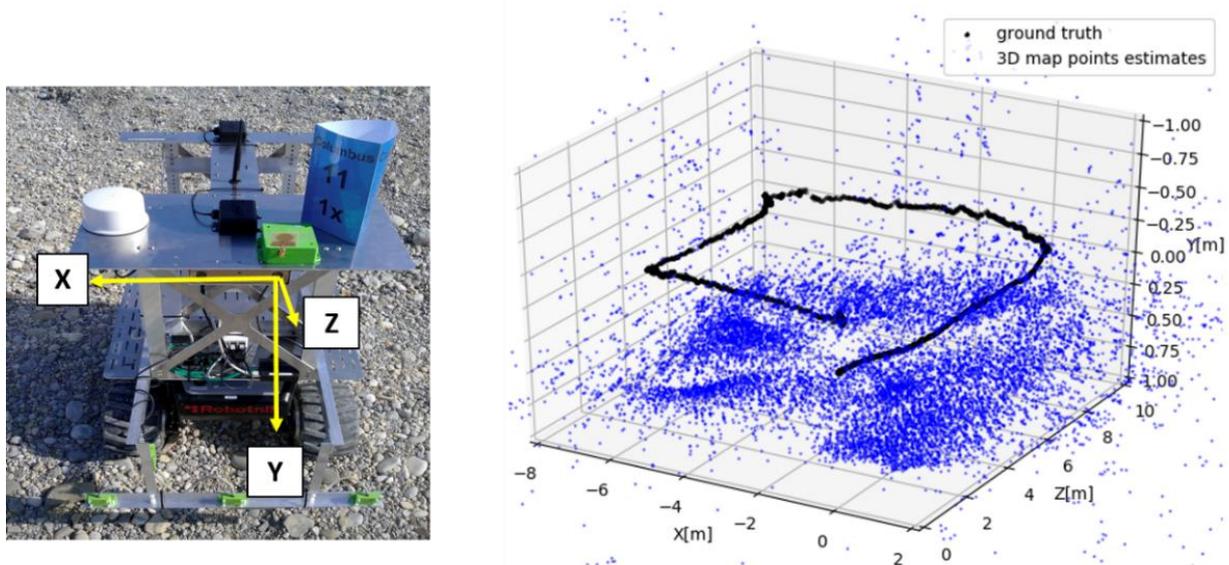
Vielzahl von implementierbaren Möglichkeiten zum Austausch der Kartendaten. Um das ROS – Kommunikationsnetz nicht zu überlasten wurde auf einen eher konservativen Ansatz für den Austausch der Kartendaten zurückgegriffen. Der Austausch wurde mit Hilfe des File Transfer Protocols (FTP) umgesetzt. Der Zugriff kann dabei jederzeit durch eines der Systeme, beispielsweise bei der Initiierung einer Mission oder aber auch periodisch erfolgen. Im Rahmen des Projektes wurde angenommen, dass die Kartendaten zentral auf einem der Systeme hinterlegt werden. Somit ist sichergestellt, dass alle Schwarmteilnehmer beim Abfragen der Kartendaten auf dieselbe Quelle zurückgreifen.

Eine Aufarbeitung der abgerufenen Kartendaten erfolgt im Anschluss auf dem abrufenden System selbst, da je nach Schwarmmitglied die Anforderungen an die finale Kartenrepräsentation, aufgrund des heterogenen Schwarms und möglicher Degradation des Systems, von den anderen Schwarmteilnehmern abweichen kann.

### 6.4.5 Vision-based sparse map database

In einem visuellen SLAM-Verfahren werden die Kartenpunkte mit einer Triangulation der extrahierten Merkmale bestimmt und die gefundenen 3D-Koordinaten in einer globalen Kartendatenbank gespeichert, einschließlich der Korrelationen zwischen den Kartenpunkten und den Merkmalen in Bildrahmen. Die Datenbank wird aktualisiert, wenn neue Kartenpunkte generiert oder die vorhandenen Daten korrigiert werden.

TUM-NAV testete die Kartengenerierung mit Bilddaten, die mit dem integrierten Rover-System erfasst wurden. Abbildung 6.4.2 zeigt die 3D-Koordinaten der geschätzten Kartenpunkte (blaue Punkte) und die Referenztrajektorie (schwarze Linie), wobei das Koordinatensystem dem ersten Kamerabild zugeordnet ist.



**Abbildung 6.4.2: Kartenpunkte, die durch Fusionierung der visuellen Positionierung und der abstandsmessungsbasierten Positionierung bestimmt wurden (rechts, blaue Punkte) und das Referenz-Koordinatensystem (links)**

## 6.5 Vehikelentwurf und Flugregelung

In diesem Arbeitspaket wird der Entwurf des Flugvehikels für die Marsumgebung behandelt. Dafür wird zuerst die Marsatmosphäre betrachtet. Anschließend wurden die Anforderungen an das Marsvehikel definiert, mit Hilfe dieser, verschiedene Konzepte betrachtet und bewertet wurden. Für die vielversprechendsten Konzepte wurde anschließend ein automatisiertes Auslegungstool in MATLAB entwickelt, welches eine schnelle Anpassung der Anforderungen an das Vehikel ermöglicht. Dadurch ist es möglich die Auswirkungen verschiedener Anforderungen und Randbedingungen auf das Vehikelkonzept zu untersuchen.

### 6.5.1 Untersuchung der Umgebungsbedingungen im Valles Marineris

Der Entwurf von Flugvehikeln ist sehr stark von den gegebenen Umgebungsbedingungen während des Fluges abhängig. Aufgrund der besonderen atmosphärischen Bedingungen am Mars, ist das Fliegen hier nur im Grenzbereich möglich. Daher ist es hier von besonderer Bedeutung möglich genaue Informationen über die Atmosphärenbedingungen im Valles Marineris zu kennen. Dabei ist wichtig zu beachten, dass sich aufgrund der Tiefe des Tales andere Bedingung als auf der Oberfläche vorhanden sind, wodurch sich andere Konzepte für die Flugvehikel ergeben. Die hier präsentierten Daten basieren auf der Mars Climate Database der ESA.

Eine Zusammenfassung der bestimmten Daten findet sich in Tabelle 6.5.1. Die Daten wurden für einen Referenzpunkt im Valles Marineris bestimmt, der sich auf einer Tiefe von 5120m befindet.

	Erde	Mars (Durchschnitt)	Mars (Tag)	Mars (Nacht)
Gravitationskonstante [ $m/s^2$ ]	9.81	3.71	3.71	3.71
Durchschnittliche atm. Temperatur [ $K$ ]	288	217.91	226.73	202.31
Durchschnittlicher atm. Druck [ $Pa$ ]	101325	895.54	882.52	918.58
Durchschnittliche atm. Dichte [ $kg/m^3$ ]	1.225	0.0216	0.0204	0.0238
Durchschnittliche Windgeschwindigkeit [ $m/s$ ]	n.a.	2.73	3.21	1.90
Max. Windgeschwindigkeit [ $m/s$ ]	n.a.	21.82	21.82	8.21
Sonneneinstrahlung [ $W/m^2$ ]	n.a.	157.21	246.11	0

**Tabelle 6.5.1: Atmosphärendaten für das Valles Marineris**

Die angegebenen Daten wurden über das Marsjahr gemittelt und sind sowohl für Marstage und –nächte, als auch gemittelt über Tag und Nacht gegeben.

## 6.5.2 Definition der Anforderungen an das Flugvehikel

Um ein gutes Design für das Flugvehikel zu erreichen, ist es notwendig eine ausreichende Menge an Anforderungen zu definieren. Mit Hilfe der definierten Anforderungen lassen sich anschließend verschiedene Konzepte für das Flugvehikel auf ihre Anwendbarkeit untersuchen. Für einen ersten Entwurf des Flugvehikels wurden folgende Anforderungen festgelegt:

Das Fluggerät muss/darf

1. eine maximale Abflugmasse nicht überschreiten,
2. nach der Landung selbstständig operativ werden können,
3. den Raketenstart und die Landung auf dem Mars überstehen,
4. den Schwebeflug und die Änderung des Schiebewinkels ermöglichen, um das Blickfeld der optischen Geräte anzupassen,
5. vertikal starten und landen können,
6. redundant gegen Ausfälle und Degradierung sein,
7. gegebene Performance-Parameter (z.B. Flughöhe, Reichweite und Flugzeit) erreichen,
8. in das Launcher-System passen,
9. die Möglichkeit besitzen, Höhlen zu erforschen,
10. bei Wind flugfähig sein,
11. Marsstürmen standhalten können.

Die hier formulierten Anforderungen sind sehr allgemein formuliert und müssen für das spätere Design mit Zahlenwerten konkretisiert werden. Für die spätere Entwicklung einer automatisierten Auslegung ist es jedoch nötig, die Anforderungen möglichst allgemeine festzulegen, um später eine leichte Anpassung der Anforderungen zu ermöglichen.

## 6.5.3 Bewertung und Auswahl verschiedener Konzepte

Um eine Vorauswahl möglicher Konzepte durchführen zu können, wurden zuerst alle möglichen Archetypen zusammengefasst (Abbildung 6.5.1).

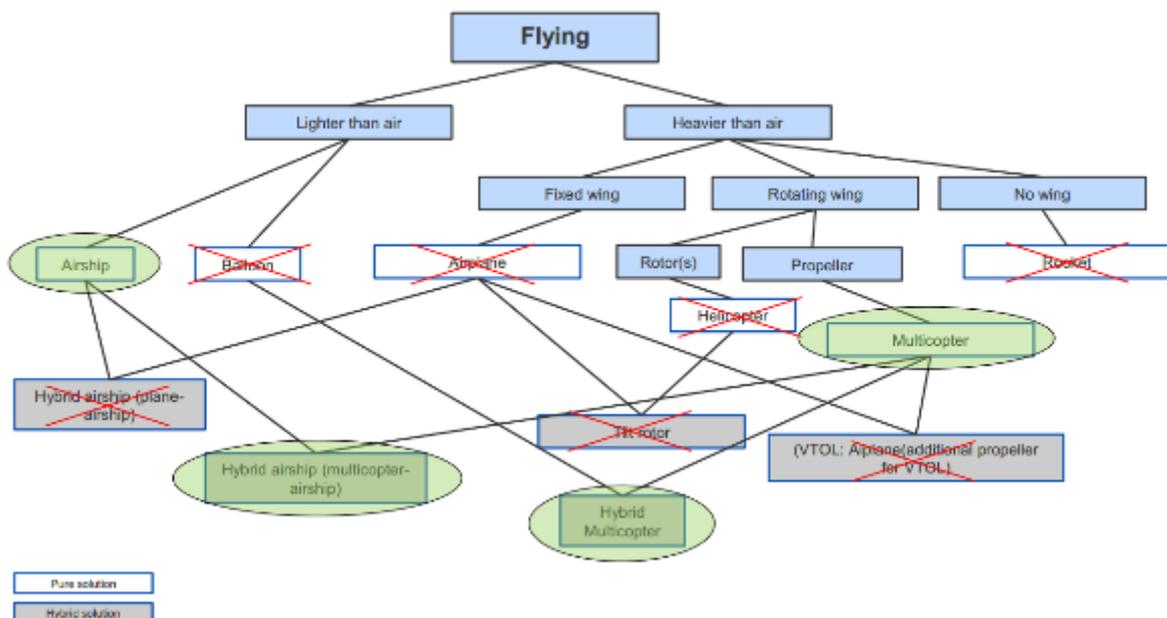


Abbildung 6.5.1: Mögliche Fluggerätdesigns

Verschiedene Konzepte konnten bereits durch eine erste einfache Betrachtung

ausgeschlossen werden, da diese eine oder mehrere Anforderungen nicht erfüllen:

- Ballon: Bietet keine Möglichkeit für eine kontrollierte Steuerung in der horizontalen Ebene.
- Flugzeug: Bietet keine Möglichkeit vertikal zu starten und landen
- Helikopter: Benötigt verstellbare Propellerblätter, welche eine aufwendige Mechanik erfordern. Dadurch ergeben sich zu viele mögliche Fehlerfälle, wodurch die nötige Redundanz nicht gewährleistet werden kann.
- Tilt-Rotor: Noch höhere Komplexität als bei einem Helikopter, wodurch sich wiederum Probleme bei der Redundanz ergeben. Zusätzlich ist für das gegebene Missionsprofil mit langen Zeiten im Schwebeflug die Fähigkeit für den effizienten Flächenflug nicht notwendig.
- VTOL Flugzeug: Analog zum Tilt-Rotor wird die Fähigkeit für einen effizienten Flächenflug für das Missionsprofil nicht benötigt.
- Luftschiff und hybrides Luftschiff: Überschlagsrechnungen haben ergeben, dass aufgrund der geringen Dichte bereits für kleine Nutzlasten sehr große Abmessungen des Luftschiffes benötigt werden, um genügend Auftrieb zu erzeugen. Dadurch kann auch dieses Konzept ausgeschlossen werden.

Die übrigbleibenden Konzepte, Multikopter (M) und hybrider Multikopter (HM), wurden zusätzlich noch um ein weiteres Konzept, Multikopter mit erweiterter Reichweite (HMER) ergänzt. Hierbei handelt sich, ähnlich wie beim hybriden Multikopter, um einen Multikopter mit einem Ballon, wobei hier der Ballon lediglich dazu dient, zusätzlich Akkus mitzunehmen und dadurch die Flugzeit zu erhöhen. Diese drei Konzepte wurden nun mit Bezug auf die definierten Anforderungen bewertet. Das Ergebnis der Bewertung ist in Tabelle 6.5.2 zusammengefasst.

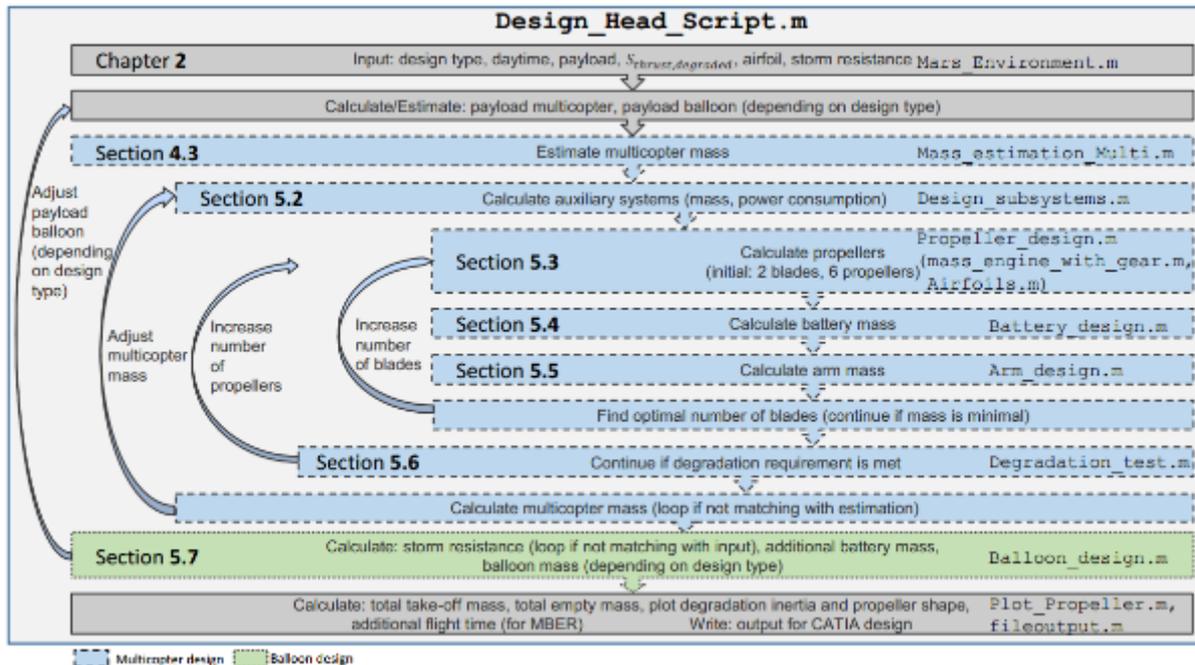
Anforderung	Gewichtung	M	HM	HMER
Max. Abflugmasse	10	3	4	2
Selbstständig operativ	4	4	2	2
Schwebeflug	9	3	5	4
Änderung des Blickfeldes	5	3	3	3
Vertikales Starten und Landen	8	3	4	3
Redundanz gegen Ausfälle	10	3	4	5
Resistenz gegen Degradierung	9	3	4	5
Max. Flughöhe	5	3	3	3
Flugzeit, Reichweite	10	3	4	5
Platz in Launcher	6	4	2	2
Flugfähigkeit bei Wind	6	4	2	2
Stürmen standhalten	9	4	2	3
Max. Geschwindigkeit	4	4	2	3
Größe, wenn operativ	4	4	3	2
<b>Gesamt</b>	-	<b>330</b>	<b>333</b>	<b>334</b>

Tabelle 6.5.2: Bewertung der Konzepte (5 (++) , 4 (+), 3 (0), 2 (-), 1 (--))

#### 6.5.4 Automatisiertes Design des Marsvehikels

Basierend auf den Ergebnissen der Bewertung der verschiedenen Konzepte wurde ein Tool zum automatisierten, vorläufigen Design des Marsvehikels erstellt. Da die Ergebnisse für die drei bewerteten Konzepte sehr nahe zusammenliegen, wurde das Tool so erstellt, dass es die

Möglichkeit bietet, alle drei Konzepte zu erstellen. In Abbildung 6.5.2 ist die allgemeine Struktur des Tools abgebildet.



**Abbildung 6.5.2: Programmstruktur des Tools zur automatisierten Berechnung eines vorläufigen Designs in MATLAB**

Für ein gegebenes Set an Eingangsdaten und je nachdem welcher Archetype gewählt wird, wird die zu tragenden Massen vom Multikopter und Ballon bestimmt. Anschließend wird das Multikopter-Design aufgerufen. Hierfür wird die Multikopter-Abflugmasse geschätzt und die einzelnen Subsysteme designt. Der Propeller wird für den benötigten Schub ausgelegt. Mit Hilfe des Propellers können nun die Masse der Motoren und der Getriebe bestimmt werden. Die Batteriemassen können mit Hilfe des benötigten Leistungsbedarfs für eine gegebene Mission bestimmt werden. Zusätzlich können passende Arme zur Aufnahme der Propeller ausgelegt werden. Anschließend wird in einer Schleife die optimale Anzahl der Propellerblätter bestimmt.

Das Degradationstool überprüft das berechnete Konzept auf die Einhaltung der Anforderung unter Degradierung oder Fehlern (Ausfall eines Rotors). Können diese Anforderungen nicht eingehalten werden, wird die Anzahl der Propeller erhöht. Die Masse des Multikopters kann nun bestimmt werden. Ist diese verschieden zur Schätzung vom Beginn, werden alle Schritte mit der neuen Masse wiederholt.

Anschließend wird die Masse des Ballons berechnet. Für den hybriden Multikopter als auch den hybriden Multikopter mit erweiterter Reichweite wird die Größe des Ballons abhängig von der Resistenz gegen Stürme berechnet. Wird die Resistenz gegenüber Stürmen nicht erreicht, wird die Massenaufteilung zwischen Multikopter und Ballon neu berechnet und alle vorherigen Schritte wiederholt. Ist die Lösung für das Design konvergiert, werden abschließend alle Massen addiert und das Ergebnis dargestellt.

Das entwickelte Tool bietet zusätzlich zur Auslegung den Export der Ergebnisse für CATIA. Dadurch können die gewonnenen Abmessungen des Designs schnell in eine 3D-Ansicht überführt werden, wodurch eine schnelle Bewertung der Ergebnisse ermöglicht wird (Abbildung 6.5.3).



**Abbildung 6.5.3: CATIA-Modell für ein hybrides Multikopter Konzept**

Das in Abbildung 6.5.3 dargestellte Modell wurde für eine durchschnittliche Dichte (Tag und Nacht) im Valles Marineris, einer Nutzlast von 400 Gramm und einer Flugzeit von 10 Minuten berechnet. Der Sicherheitsfaktor für den degradierten Zustand wurde zu 1.1 gewählt und es wurde eine Resistenz gegenüber Stürmen von 21.82 m/s angenommen. In Abbildung 6.5.4 ist ein Vergleich der drei verschiedenen Archetypen für dieselben Anforderungen dargestellt.



**Abbildung 6.5.4: CATIA-Modelle der verschiedenen Konzepte für die gleichen Anforderungen**

## 6.5.5 Prozessbasierter Flugregelungsentwurf

### 6.5.5.1 Prozessentwicklung und Anforderungsgenerierung

In diesem AP wird der Prozess für den terrestrischen Flugregelungsentwurf entwickelt. Zusammen mit dem in AP5220 entwickelten Regler sollen so Eigenschaften des Gesamtsystems garantiert werden. Ziel ist es, den Prozess so zu gestalten, dass er direkt auf den Marsregler anwendbar ist. Nur numerische Werte, also z.B. die Temperatur und systemrelevante Grenzen, unterscheiden sich. AP5400 soll dann untersuchen ob ein Mars-Regler durch eine Emulation experimentell auf der Erde validierbar ist.

Die extraterrestrische Exploration erfordert einen hohen Ressourceneinsatz und wird mit wenigen Trägersystemen ausgeführt. Um die Mission und damit auch die monetäre Investition nicht zu gefährden, muss sichergestellt werden, dass die fliegenden Schwarmteilnehmer auch unter widrigen Bedingungen einwandfrei funktionieren. Um dies zu gewährleisten wird der Flugregelungsentwurf gemäß eines vorher definierten Prozesses durchgeführt. Das Ziel des prozessbasierten Flugregelungsentwurfs ist es Aussagen über die Leistung und Robustheit des eingesetzten Reglers und Flugvehikels abzuleiten.

In der zivilen Luftfahrt spielt der prozessbasierte Regelungsentwurf eine zentrale Rolle bei der Zulassung. Hier sind allerdings nicht nur monetäre Interessen wichtig, sondern der Sicherheit wird die zentrale Rolle zugesprochen. Da in der zivilen Luftfahrt schon ein stringenter und erprobter Prozess für den Entwurf eines Flugreglers existiert, soll dieser soweit möglich und soweit sinnvoll verfolgt werden.

Zuständig für die kommerzielle Zulassung im Luftfahrtbereich sind die Behörden *European Aviation Safety Agency* (EASA) in Europa und *Federal Aviation Administration* (FAA) in den USA. Das Regelwerk für die Zulassung ist in den sogenannten *certification specifications* (CS) zusammengefasst. Die Klasse der „Normal, utility, aerobatic and commuter“ Flugzeuge einer gewissen Größe ist in der CS23 beschrieben. Die CS23 besteht aus 2 Teilen. Der erste Teil behandelt den „airworthiness code“, welcher die Anforderungen an das Flugzeug definiert. Der zweite Teil beinhaltet die *Acceptable Means of Compliance* (AMC), welche einen Weg vorgeben die Einhaltung der Anforderungen zu überprüfen und aufzuzeigen.

An sich ist der prozessbasierte Flugregelungsentwurf ein Teil des kompletten Flugzeugsystementwurfs. Für diesen gibt es empfohlene Vorgehensweisen, Richtlinien und Standards, die bei Einhaltung eine Zertifizierung erleichtern. Alle Abweichungen müssen einzeln begründet und geprüft werden. Den kompletten Flugzeugentwicklungsprozess beschreiben die erprobten Vorgehensweisen der SAE ARP 4754a ().

Der Prozess nach SAE ARP 4754a betrachtet das komplette Flugsystem, also Hardware und Software. Für das Projekt VaMEx CoSMiC stehen allerdings bisher nur Voruntersuchungen und Machbarkeitsstudien, die das Flugvehikel betreffen, zur Verfügung. Die Konzeption eines kompletten Flugvehikels erfordert allerdings rigores Systems Engineering, um sicherzugehen, dass das Vehikel auf dem Mars wunschgerecht funktioniert. Als Konsequenz wird im Folgenden nur ein funktional basierter Flugregelungsentwurfsprozess durchlaufen. Trotzdem besteht die Möglichkeit die Ergebnisse in späteren Entwürfen mit wenig Aufwand zu integrieren. Der folgende Abschnitt grenzt den prozessbasierten Flugregelungsentwurf im Projekt CoSMiC ab.

Die zentralen Dokumente des Entwicklungsprozesses beinhalten die „Environment Definition“, die „Concept of Operation“ (ConOps) sowie den „Anforderungskatalog“. Die „Environment Definition“ beinhaltet die im Abschlussbericht von VaMEx definierten Umgebungsbedingungen für den Mars. Für die Anwendung auf der Erde enthält dieses Dokument die terrestrischen Daten. Die ConOps beschreibt die Charakteristika des Systems. Alle für die Verwendung relevanten Daten über das System werden hier zusammengefasst. Dies beinhaltet unter anderem die Systemarchitektur, die auszuführenden Operationen sowie die Systemparameter. Der Anforderungskatalog enthält alle Anforderungen, die an ein System gestellt werden. Diese stammen aus verschiedenen Quellen. So können funktionale Anforderungen vom Auftraggeber an das System gestellt werden, während

Sicherheitsanforderungen aus den Sicherheitsanalysen stammen.

### 6.5.5.2 Einordnung des prozessbasierten Flugregelungsentwurfs in CoSMiC

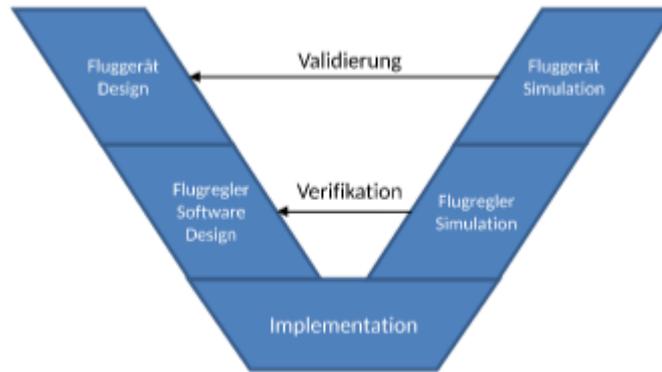
Der Aufwand für einen kompletten Flugsystementwurf gemäß SAE ARP 4754a übersteigt den Umfang von CoSMiC deutlich. Auch existiert noch kein komplett entworfenes System für das fliegende Marsvehikel. Dementsprechend wird der Prozess auf Flugregler eingeschränkt. Zentrale Bestandteile des Entwicklungsprozesses sind der Sicherheitsprozess und der funktionale Prozess. Der Sicherheitsprozess untersucht mögliche Fehlerfälle von Subsystemen und deren Auswirkung auf das Gesamtsystem bzw. den Erfolg der Operation. Die Methoden die hier zum Einsatz kommen beinhalten unter anderem das Functional Hazard Assessment, das Preliminary System Safety Assessment, das System Safety Assessment und die Common Cause Analysis. Der funktionale Prozess hat zur Aufgabe die übergeordneten Ziele in einzelne Funktionen zu unterteilen. Soll das System z.B. in der Lage sein Wegpunkten zu folgen, so ergeben sich nicht nur Anforderungen an den Positionsregler sondern auch an den Lageregler (andernfalls wäre z.B. ein rollenschlagendes Multirotorsystem akzeptabel, aber sicher nicht erwünscht). Das Ergebnis des Entwicklungsprozesses ist eine funktionale Struktur des Regelsystems sowie Anforderungen an das Flugregelungssystem, denen es genügen muss, um einen reibungslosen Einsatz auf dem Mars zu gewährleisten.

Ein kompletter Flugregler besteht aus Sensoren, Recheneinheiten, Aktuatoren, Energieversorgung und Schnittstellen. Wie beschrieben, existiert noch kein finales Flugvehikel für die Marsmission. Somit kann auch der beschriebene Sicherheitsprozess noch nicht vollständig durchgeführt werden. Die folgenden Abschnitte konzentrieren sich somit auf den funktionalen Prozess und die physikalischen Eigenschaften des Systems.

Es stellt sich die Frage wie man geeignete Anforderungen an das System stellt, bzw. diese von bereits gegebenen Anforderungen ableitet. Hierbei unterstützen bereits vorhandene und in der Luftfahrt breit genutzte Dokumente. Paragraph §CS23.1309 der CS23 beinhaltet Anforderungen an die Flugregelungsalgorithmik. Allerdings beinhaltet der zweite Teil der CS23 noch keine AMC für §CS23.1309. Für die AMC wurde zwar bereits ein Absatz in der CS23 reserviert, dieser befindet sich allerdings noch in Entwicklung. Stattdessen besagt die EASA CM SWCEH-002 (), dass der Standard DO-178B/C () akzeptiert wurde ausreichende AMC bereitzustellen. Die DO-178B/C stellt “[...] guidance for the production of software for airborne systems and equipment that performs its intended function with a level of confidence in safety that complies with airworthiness requirements” () zur Verfügung. Um dies zu schaffen, benötigt die DO-178B/C die vorherige, genaue Spezifikation der Software unter Verwendung von Anforderungen (). Passende Anforderungen sind unter anderem in AS94900 (), ARP94910 () und MIL-HDBK-1797 () zu finden.

Die in den genannten Dokumenten formulierten Anforderungen an den Flugregelungsalgorithmus sind basierend auf linearen Flugreglern entstanden. In VaMEx soll allerdings ein adaptiver Regler zum Einsatz kommen, da dieser auf unvorhergesehene Ereignisse reagieren kann. So sind z.B. Robustheitsanalysen mit Phasen- und Amplitudenrand nicht für adaptive Regler definiert. Da nichtlineare Regler zum Einsatz kommen werden die Anforderungen ausschließlich im Zeitbereich betrachtet und nicht im Frequenzbereich.

Der weitere Entwicklungsprozess orientiert sich an einem vereinfachten V-Modell (siehe Abbildung 6.5.5). Ausgehend vom Fluggerät und den funktionalen Anforderungen wird eine Regelungsstruktur entworfen. Diese wird anschließend implementiert. In Folge soll der implementierte Flugregler verifiziert werden. Die Verifikation soll überprüfen, ob die Anforderungen, die im Entwicklungsprozess erarbeitet wurden, erfüllt werden. Dazu zählt z.B. auch die Frage, ob der Regelungsalgorithmus richtig implementiert wurde. Im Anschluss an die Verifikation erfolgt die Validierung. Letztere soll überprüfen, ob das verifizierte System die an es gestellten Aufgaben erfüllen kann. Dem V-Modell liegt ein iterativer Prozess zu Grunde. Falls also z.B. eine Anforderung nicht erfüllt werden kann, wird das Software Design angepasst.



**Abbildung 6.5.5: Vereinfachtes V-Modell für funktionalen Flugreglerentwurf**

Die Verifikation und Validierung der Regelungsstruktur soll mit Hilfe von numerischen Simulationen erfolgen. Hierzu wird das in VaMEx 1 erstellte Simulationsmodell herangezogen. Des Weiteren wird in AP5400 untersucht inwiefern die Dynamik eines Marsvehikels mit Hilfe eines terrestrischen Vehikels experimentell emuliert werden kann. Als Vorbild dient das Forschungsflugzeug ATTAS des DLR, das kurze Zeit die Dynamik anderer Flugzeuge nachbilden kann. Gelingt dies, so wäre es möglich den Regelungsalgorithmus des Marsvehikels experimentell zu verifizieren und zu validieren.

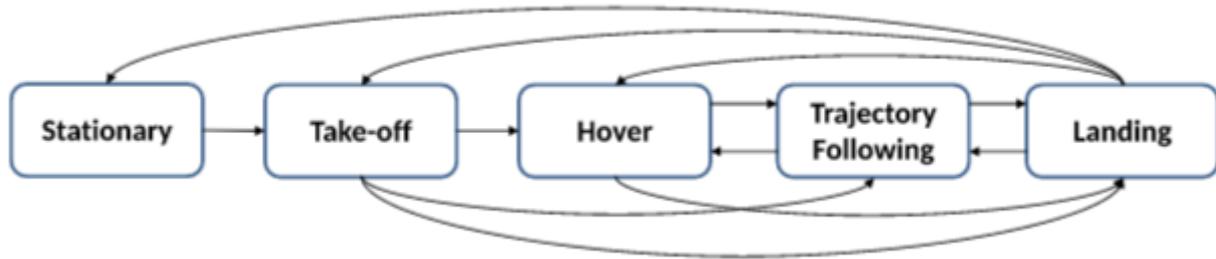
Die folgenden Abschnitte sollen einen kurzen Eindruck vermitteln, wie die Entwicklung des Flugregelungssystems abläuft. Dazu werden zunächst die Bewegungsprimitiven, die ein fliegender Schwarmteilnehmer erfüllen können soll, aufgezeigt. Daraus abgeleitet stellt Kapitel 6.5.5.4 zwei Anforderungen auf, die an die Bewegungsprimitiven gestellt werden.

### 6.5.5.3 Bewegungsprimitiven

Das fliegende Marsvehikel soll in der Lage sein verschiedene Manöver auszuführen. Dazu zählt unter anderem die Wegpunktfolge sowie die Inspektion eines interessanten Objekts. Die Manöver sind in der ConOps zusammengefasst. Aus den Manövern lassen sich Bewegungsbausteine, sogenannte Bewegungsprimitiven, ableiten, die jedes Manöver in verschiedene Subbewegungsmodulen unterteilen. So muss das Vehikel bei einer Inspektion zB. in der Lage sein zu hovern. Abbildung 6.5.6 zeigt die fünf in der ConOps abgeleiteten Bewegungsprimitiven der fliegenden Schwarmteilnehmer. Die einzelnen Modi sind in der ConOps genauer beschrieben. Jedes spezifizierte Manöver lässt sich durch geeignete Kombination der Primitiven nachstellen.

Dieses Vorgehen wird verwendet, da z.B. das automatische Starten in fast allen Manövern vorkommt und sonst die daraus abgeleiteten Anforderungen für jedes Manöver getestet werden müssten. Durch die Verwendung der Primitiven, muss das automatische Starten nur einmal getestet werden. Des Weiteren ermöglicht die Aufteilung in Bewegungsprimitiven die Berücksichtigung verschiedener Genauigkeiten und sogar Reglermodi. So ist zB. bei einer Trajektorienfolge eine höhere Abweichung von der Solltrajektorie erlaubt als bei der automatischen Landung. Dies schlägt sich oft in unterschiedlichen Reglerparametern nieder. In CoSMiC wird für jedes Bewegungsprimitiv ein Reglermodus vorgesehen. Die Pfeile in Abbildung 6.5.6 repräsentieren die Möglichkeiten des Algorithmus zwischen den Reglermodi umzuschalten. So kann z.B. vom Hover-Modus in die Trajektorienfolge, oder in die Landung geschaltet werden. Viele Schaltungen sind allerdings nicht erlaubt. So macht es z.B. keinen Sinn vom Stationärmodus in den Landemodus zu schalten.

Für jedes Bewegungsprimitiv lassen sich Anforderungen an die Genauigkeit ableiten. Das folgende Kapitel zeigt exemplarisch zwei Auszüge aus dem Anforderungskatalog.



**Abbildung 6.5.6: Bewegungsprimitiven der Flugvehikel und der modularen Zusammenhang im Regelungsalgorithmus**

#### 6.5.5.4 Auszug aus dem Anforderungskatalog

Im Folgenden sind zwei Anforderungen aus dem Anforderungskatalog aufgezeigt. Die erste Anforderung behandelt den Landevorgang. Die zweite Anforderung behandelt die zulässigen transienten Positionsfehler beim Umschalten der Reglermodi.

Tabelle 6.5.3 zeigt die oberste Anforderung an die automatische Landung. Wie in ConOps beschrieben besteht der Landevorgang aus zwei Phasen. Die erste Phase ist gültig solange sich das Flugvehikel zwischen der initialen Höhe und  $0,4[m]$  über dem designierten Landepunkt befindet. Das Ziel der ersten Phase ist eine zügige Verringerung der Höhe. Die zweite Phase beschreibt den finalen Landevorgang. Das fliegende Vehikel soll möglichst sanfte am Landepunkt aufsetzen und nur einen geringen horizontalen Positionsfehler am Ende des Landevorgangs besitzen. Störeinflüsse, wie z.B. der Bodeneffekt, erschweren die Regelaufgabe.

Mit dynamischen Manövern sind oft größere Abweichungen von Solltrajektorien verbunden. Somit sind in der ersten Phase höhere Positionsabweichungen zugelassen als in der zweiten. Das vorrangige Ziel hier ist die Höhenverminderung. In der zweiten Phase soll nur noch ein kleiner Positionsfehler zugelassen werden, um nicht zu weit vom Landepunkt entfernt aufzutreffen. Des Weiteren ist eine maximale Landegeschwindigkeit vorgegeben, um das Vehikel vor Schäden durch den Landestoß zu schützen. Aus den Anforderungen an die Positionsgüte lassen sich Anforderungen an die anderen Zustände und Reglerschleifen ableiten.

Tabelle 6.5.4 zeigt eine Anforderung an die Positionierungsgenauigkeit beim Umschalten von Reglermodi. Als Basis dienen die erlaubten Schaltungen aus Abbildung 6.5.6. Wird zwischen Reglermodi umgeschaltet, so werden oft intern Werte neu initialisiert. Meistens beruhen die Initialisierungen auf Messungen. Da Messungen verrauscht sind, kommt es somit zu Fehlinitialisierungen und der Flugregelungsalgorithmus reagiert darauf. Dies wiederum führt zu Transienten in den Zuständen des Systems. Während dieses transiente Verhalten zu erwarten ist, muss trotzdem sichergestellt werden, dass definierte Grenzen nicht überschritten werden.

<b>REQ ID</b>	<b>RQ-VME-2213-1102</b>		
<b>REQ Name</b>	<b>Automatic Landing</b>		
<b>Category</b>	DERIVED		
<b>Subcategory</b>	PERFORMANCE		
<b>Priority</b>	1		
<b>Description</b>	<p>The vehicle shall be able to automatically land at a designated landing point <math>(x, y, h)_{L,des}</math> (see ConOps for maneuver description) under the following conditions:</p> <ul style="list-style-type: none"> <li>the maximal horizontal deviation <math>\Delta(x, y)_{L,acc,M}</math> from the landing point <math>(x, y, h)_{L,des}</math> during the maneuver is 2 [m],</li> <li>the maximal altitude <math>(\Delta h_{L,1} + \Delta h_{L,2})</math> of 2 [m] above the final point shall not be exceeded during the maneuver,</li> <li>the maximal deviation <math>\Delta(x, y)_{L,acc,S}</math> from the final point at the <math>(x, y, h)_{L,des}</math> end of the maneuver in the horizontal plane is 0.5 [m],</li> <li>a final attitude <math>\theta_{inc}</math> up to 20 [°] about any axis in the horizontal plane is allowed,</li> <li>any final yaw angle is allowed,</li> <li>the maximal vertical speed of 1 [m/s] during the final stage of the maneuver shall not be exceeded.</li> </ul>		
<b>Rationale / Discussion</b>	<ul style="list-style-type: none"> <li>Integral part of every maneuver.</li> <li>The maximal landing velocity is limited by the structure.</li> <li>The landing area might have an inclination or the flying vehicle lands on rocks, which cause a non-zero attitude</li> </ul>		
<b>Scope of Validity</b>	Nominal environmental and degradation conditions.		
<b>Means of Compliance</b>	MoC 8 – The compliance is shown by analyzing a set of simulation test cases taking parameter variances into account.		
<b>Upstream Trace</b>	RQ-VME-2213-0101		
<b>Assumptions</b>	No obstacles within the described boundaries of the maneuver.		
<b>References</b>	<p><b>Applicable Documents:</b></p> <ul style="list-style-type: none"> <li>CoSMiC ConOps</li> <li>VaMEx environmental description</li> </ul> <p><b>Guidance Material:</b> None</p>		
<b>Validation Status</b>	DRAFT FOR REVIEW		
<b>Initial Version</b>	<b>Doc. Version</b>	<b>Date</b>	<b>By</b>
	0.1	2016-07-04	Mühlegg Maximilian
<b>Last Change</b>	0.3	2016-12-12	Mühlegg Maximilian

Tabelle 6.5.3: Anforderung an die Landeperformance eines fliegenden Schwarmteilnehmers

<b>REQ ID</b>	<b>RQ-VME-2213-1104</b>
---------------	-------------------------

<b>REQ Name</b>	<b>Transient Performance Bounds after mode switches</b>																																						
<b>Category</b>	DEIVED																																						
<b>Subcategory</b>	N/A																																						
<b>Priority</b>	1																																						
<b>Description</b>	<p>The transient position error occurring within <math>1 [s]</math> after a switch between modes as described in ConOps shall not be greater than specified in the matrix below. Units are in <math>[m]</math> and values are relative to the switching position. Switches only occur if the conditions stated in ConOps are met. Switches which are not allowed are labeled with 'Not allowed' in the following.</p> <table border="1"> <thead> <tr> <th>↓ From/To →</th> <th>Stationary</th> <th>Takeoff</th> <th>Hover</th> <th>Trajectory Following</th> <th>Landing</th> </tr> </thead> <tbody> <tr> <th>Stationary</th> <td>X</td> <td><math>\Delta(x,y) = 0[m]</math> <math>\Delta h = 0[m]</math></td> <td>Not allowed</td> <td>Not allowed</td> <td>Not allowed</td> </tr> <tr> <th>Takeoff</th> <td>Not allowed</td> <td>X</td> <td><math>\Delta(x,y) = 0.2[m]</math> <math>\Delta h = 0.2[m]</math></td> <td><math>\Delta(x,y) = 0.2[m]</math> <math>\Delta h = 0.2[m]</math></td> <td><math>\Delta(x,y) = 0.2[m]</math> <math>\Delta h = 0.2[m]</math></td> </tr> <tr> <th>Hover</th> <td>Not allowed</td> <td>Not allowed</td> <td>X</td> <td><math>\Delta(x,y) = 0.2[m]</math> <math>\Delta h = 0.2[m]</math></td> <td><math>\Delta(x,y) = 0.2[m]</math> <math>\Delta h = 0.2[m]</math></td> </tr> <tr> <th>Trajectory Following</th> <td>Not allowed</td> <td>Not allowed</td> <td><math>\Delta(x,y) = 0.2[m]</math> <math>\Delta h = 0.2[m]</math></td> <td>X</td> <td><math>\Delta(x,y) = 0.2[m]</math> <math>\Delta h = 0.2[m]</math></td> </tr> <tr> <th>Landing</th> <td><math>\Delta(x,y) = 0[m]</math> <math>\Delta h = 0[m]</math></td> <td><math>\Delta(x,y) = 0.2[m]</math> <math>\Delta h = 0.2[m]</math></td> <td><math>\Delta(x,y) = 0.2[m]</math> <math>\Delta h = 0.2[m]</math></td> <td>Not allowed</td> <td>X</td> </tr> </tbody> </table>			↓ From/To →	Stationary	Takeoff	Hover	Trajectory Following	Landing	Stationary	X	$\Delta(x,y) = 0[m]$ $\Delta h = 0[m]$	Not allowed	Not allowed	Not allowed	Takeoff	Not allowed	X	$\Delta(x,y) = 0.2[m]$ $\Delta h = 0.2[m]$	$\Delta(x,y) = 0.2[m]$ $\Delta h = 0.2[m]$	$\Delta(x,y) = 0.2[m]$ $\Delta h = 0.2[m]$	Hover	Not allowed	Not allowed	X	$\Delta(x,y) = 0.2[m]$ $\Delta h = 0.2[m]$	$\Delta(x,y) = 0.2[m]$ $\Delta h = 0.2[m]$	Trajectory Following	Not allowed	Not allowed	$\Delta(x,y) = 0.2[m]$ $\Delta h = 0.2[m]$	X	$\Delta(x,y) = 0.2[m]$ $\Delta h = 0.2[m]$	Landing	$\Delta(x,y) = 0[m]$ $\Delta h = 0[m]$	$\Delta(x,y) = 0.2[m]$ $\Delta h = 0.2[m]$	$\Delta(x,y) = 0.2[m]$ $\Delta h = 0.2[m]$	Not allowed	X
	↓ From/To →	Stationary	Takeoff	Hover	Trajectory Following	Landing																																	
	Stationary	X	$\Delta(x,y) = 0[m]$ $\Delta h = 0[m]$	Not allowed	Not allowed	Not allowed																																	
	Takeoff	Not allowed	X	$\Delta(x,y) = 0.2[m]$ $\Delta h = 0.2[m]$	$\Delta(x,y) = 0.2[m]$ $\Delta h = 0.2[m]$	$\Delta(x,y) = 0.2[m]$ $\Delta h = 0.2[m]$																																	
	Hover	Not allowed	Not allowed	X	$\Delta(x,y) = 0.2[m]$ $\Delta h = 0.2[m]$	$\Delta(x,y) = 0.2[m]$ $\Delta h = 0.2[m]$																																	
	Trajectory Following	Not allowed	Not allowed	$\Delta(x,y) = 0.2[m]$ $\Delta h = 0.2[m]$	X	$\Delta(x,y) = 0.2[m]$ $\Delta h = 0.2[m]$																																	
	Landing	$\Delta(x,y) = 0[m]$ $\Delta h = 0[m]$	$\Delta(x,y) = 0.2[m]$ $\Delta h = 0.2[m]$	$\Delta(x,y) = 0.2[m]$ $\Delta h = 0.2[m]$	Not allowed	X																																	
<b>Rationale / Discussion</b>	If the autopilot switches between modes, the controller shall be able to bound the position transients occurring due to the switch.																																						
<b>Scope of Validity</b>	Nominal environmental and degradation conditions.																																						
<b>Means of Compliance</b>	MoC 8 – The compliance is shown by analyzing a set of simulation test cases taking parameter variances into account.																																						
<b>Upstream Trace</b>	RQ-VME-2213-0101																																						
<b>Assumptions</b>	<ul style="list-style-type: none"> <li>• There are no obstacles within the described boundaries of the maneuver.</li> <li>• The vehicle is 100% functional</li> </ul>																																						
<b>References</b>	<b>Applicable Documents:</b> <ul style="list-style-type: none"> <li>• CoSMiC conops</li> <li>• VaMEx environmental description</li> </ul>																																						
	<b>Guidance Material:</b> None																																						
<b>Validation Status</b>	DRAFT FOR REVIEW																																						
<b>Initial Version</b>	<b>Doc. Version</b>	<b>Date</b>	<b>By</b>																																				
	0.1	2016-07-04	Maximilian Mühlegg																																				
<b>Last Change</b>	0.3	2016-12-13	Maximilian Mühlegg																																				

**Tabelle 6.5.4: Anforderung an die Positionierungsgenauigkeit beim Umschalten von Reglermodi**

## 6.5.6 Entwicklung Flugregler

Auf Basis der Ergebnisse der VaMEx 1 Projektes und aus den Ergebnissen der Prozessbasierten Flugregelungsentwicklung in den vorhergegangenen Kapiteln wurde nun eine Flugregelungsalgorithmus entwickelt. Die implementierte Regelungsstruktur basiert auf einem kaskatiertem Aufbau, bestehend aus einer Lageregelung und einer übergeordneten Positionsregelung.

Weiter Anforderungen an den Regelungsalgorithmus ergeben sich aus dem Anwendungsszenario auf dem Mars. Hier ist einerseits die lange Einsatzzeit zu nennen. Dadurch ergeben sich Verschleiß oder Degradierung des angewendeten Systems. Dadurch muss der entwickelte Algorithmus eine hohe Robustheit gegenüber Ausfällen einzelner Systemkomponenten haben. Die am kritischsten zu bewertende Komponente ist die die Antriebseinheit, bestehend aus Propeller und Motor. Bei Ausfall eines Propellers ergeben sich starke Einschränkungen bzw. ein Komplettausfall der Flugfähigkeit bei klassischen Flugregelungsalgorithmen. Daher wurde für den Flugregler im VaMEx-CoSMiC Projekt eine adaptive Komponente hinzugefügt. Dies ermöglicht Echtzeitschätzung von Ausfällen und Degradierung der Propeller und eine Anpassung des Regelungsalgorithmus während des Fluges.

Im Folgenden wird kurz zu dem Thema und den angewendeten Methoden verwandte Arbeiten eingegangen. Merheb et al. beschäftigten sich bereits in mit fehlertoleranter Positionsregelung. Diese Algorithmen wurden jedoch hauptsächlich in Simulationen validierte, wohingegen die für dieses Projekt entwickelten Algorithmen mit Schwerpunkt in Flugversuchen getestet wurden. legten in ihrer Arbeit den Schwerpunkt auf Fehlererkennung und Identifikation, beschäftigten sich Anpassung der Regelungsstrategie unter Annahme das der Regler bereits erkannt wurde. Mit der speziellen Anwendung auf Hexakopter-System beschäftigten sich unter anderem [29], [30], [31] und [32].

Die im Folgenden präsentierten Herleitung und Struktur des entwickelten Reglers wurde bereits in veröffentlicht. Die präsentierten Ergebnisse beziehen sich auf diese Veröffentlichung.

Der Regler kann in vier unterschiedlich Sub-Komponenten unterschieden werden: Lageregelung, Positionsregelung, Adaptive Control Allocation und Control Effectiveness Schätzung (Steuereffektivitätsschätzung). Der Lage- und Positionsregler wird in Kapitel 6.5.6.1 vorgestellt und hergeleitet. Kapitel 6.5.6.2 beschäftigt sich mit der Adaptiven Control Allocation und Kapitel 6.5.6.3 beschreibt die Control Effectiveness Schätzung.

### 6.5.6.1 Hexakopter-Dynamik und Baseline-Flugregler

Zu Beginn dieses Kapitels wird die für die Flugregelung verwendete Dynamik des Multirotorsystems hergeleitet. Zur Herleitung dieser wird das inertielle Koordinatensystem verwendet, welches den Ursprung im Erdmittelpunkt hat. Verwendet man das zweite Newtonsche Gesetz und vernachlässigt man den aerodynamischen Widerstand erhält man

$$m\dot{v} = t + mg. +$$

Hier beschreibt  $g$  den Erdbeschleunigungsvektor im körperfesten Koordinatensystem,  $m$  ist den Masse des Hexakopters,  $t$  gibt die gesamte Schubkraft und  $v$  beschreibt die Geschwindigkeit des Hexakopters relativ zum inertialen System.

Die Rotationsdynamik kann mit Hilfe der Eulergleichung

$$J\dot{\omega} = -\omega \times J\omega + M_p$$

hergeleitet werden. Hier beschreibt  $J$  das Massenträgheitsmoment des Multikoptersystems,  $\omega$

gibt die Winkelgeschwindigkeit des körperfesten relativ zum inertialen Koordinatensystem.  $M_P$  gibt die durch Schub erzeugten Momente. Für die weiteren Herleitung wird der Schubvektor mit  $t = -Tz_B$  angenommen. Dadurch kann die Bewegungsgleichung beschrieben werden als

$$\begin{aligned} t &= R_{IB} T_u(T) \begin{pmatrix} \omega_x \\ \omega_z \\ T \end{pmatrix} \\ &= -\dot{T}z_B + R_{IB} T_\omega(T) \omega. \end{aligned}$$

Die Matrix  $R_{IB}$  gibt dabei die Rotation vom körperfesten ins inertielle System. Die Matrizen  $T_\omega$  und  $T_u$  sind definiert durch:

$$T_\omega = \begin{bmatrix} 0 & -T & 0 \\ T & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

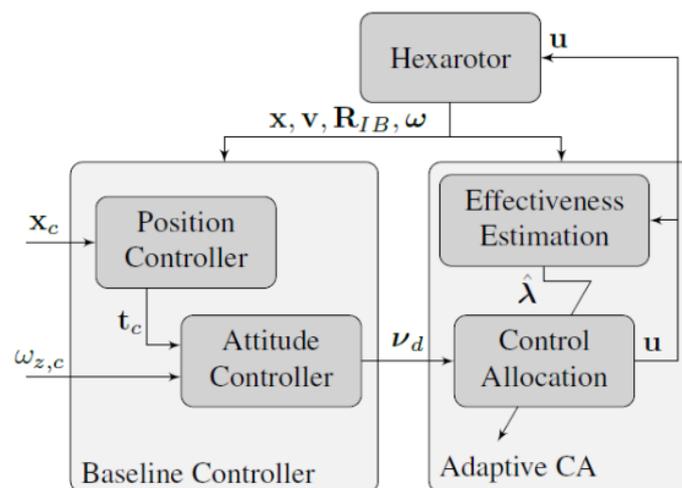
$$T_u = \begin{bmatrix} 0 & -T & 0 \\ T & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix}$$

Eine ausführlichere Herleitung der Bewegungsgleichung kann in gefunden werden.

In Abbildung 6.5.7 ist nun die Struktur des entwickelten Flugreglers gegeben. Die kommandierten Größen der Positionsschleife sind die gewünschte Position  $x_c$  im Regelungskoordinatensystem und die gewünschte Gierrate  $\omega_{z,c}$  ebenfalls im Regelungskoordinatensystem.

Die Positionsschleife berechnet aus den gewünschten Kommandos eine benötigten Schubvektor  $t_c$  welcher als Eingang für den Lageregler dient. Dieser berechnet daraus die gewünschten Pseudo-Controls  $v_d$  welche der Eingang der Adaptiven Control Allocation sind. Die Adaptive Control Allocation besteht aus zwei Teilen: der Effektivitätsschätzung und der eigentlichen Control Allocation. Die Control Allocation berechnet aus den gewünschten Pseudo-Controls die Drehzahl-Kommandos der einzelnen Propeller  $u$  des Hexakopters.

Der Positions- zusammen mit dem Lageregler werden als Baseline-Regler bezeichnet und werden als nächstes präsentiert.



**Abbildung 6.5.7: Struktur des entwickelten Flugreglers**

Als erstes soll die Positionsregelungsschleife kurz hergeleitet werden. Eine ausführliche Herleitung kann in [33] gefunden werden. Zur Bestimmung der gewünschten Position  $x_d$  aus

der kommandierten Position  $x_c$  wird ein Referenzmodell zweiter Ordnung verwendet:

$$\begin{pmatrix} \dot{x}_d \\ \dot{v}_d \end{pmatrix} = \begin{bmatrix} \mathbf{0}_3 & \mathbf{I}_3 \\ -\mathbf{K}_p & -\mathbf{K}_d \end{bmatrix} \begin{bmatrix} x_d \\ v_d \end{bmatrix} + \begin{bmatrix} \mathbf{0}_3 \\ \mathbf{K}_p \end{bmatrix} x_c$$

Hier beschreibt  $\mathbf{I}_3$  und  $\mathbf{0}_3$  die Einheitsmatrix und Nullmatrix.  $\mathbf{K}_p, \mathbf{K}_d$  sind positiv definite Matrizen, so gewählt, dass das System exponentiell stabil ist.

Um die Zustände des Referenz-Modells zu tracken wird die Fehler-Dynamik des Tracking-Fehlers definiert:

$$e_p := \begin{pmatrix} e_x \\ e_v \end{pmatrix} = \begin{pmatrix} x_d - x \\ v_d - v \end{pmatrix}$$

Unter Verwendung des linearen Regelungsgesetz

$$t_c := m(\dot{v}_d - g + \mathbf{K}_x e_x + \mathbf{K}_v e_v + \mathbf{K}_i e_{x,i})$$

kann Stabilität der Positionsschleife bewiesen werden. Die Matrizen  $\mathbf{K}_x, \mathbf{K}_v, \mathbf{K}_{x,i}$  werden wieder positiv definit gewählt, wodurch exponentielle Stabilität des Tracking-Fehlers erreicht wird.

Nachdem die Positionsschleife hergeleitet wurde, werden nun die wichtigsten Gleichungen für die Herleitung des Lagereglers präsentiert.

Als Eingang des Lagereglers dienen der kommandierte Schubvektor des Positionsreglers und die gewünschte Gierrate.

Als ersten Schritt werden wiederum Referenzmodelle benötigt. Für den absoluten Schub wird ein Referenzmodell erster Ordnung verwendet, welches durch die Gleichung

$$\dot{T}_d = -K(T_d - T_c)$$

gegeben ist. Hier ist  $T_d$  der gewünschte Schub und  $K$  die Eigenfrequenz des Referenzmodells. Das Referenzmodell der Lage wurde mit Ordnung zwei gewählt und wird beschrieben durch

$$\begin{pmatrix} \dot{\mathbf{R}}_{IR} \\ \dot{\boldsymbol{\omega}}_r \end{pmatrix} = \begin{pmatrix} \mathbf{R}_{IR} \boldsymbol{\omega}_r^\times \\ -\mathbf{K}_d \boldsymbol{\omega}_r + \mathbf{K}_p \mathbf{R}_{RI} (z_c \times z_d) \end{pmatrix}.$$

Hier beschreibt  $\mathbf{R}_{IR}$  eine Rotationsmatrix, welche einen Vektor von Referenzmodell-Koordinatensystem in das inertielle System transformiert.  $\boldsymbol{\omega}_r$  ist die Winkelgeschwindigkeit des Referenzmodell-Systems zum inertialen System. Die Matrizen  $\mathbf{K}_p, \mathbf{K}_d$  werden wiederum so gewählt, dass das Referenzmodell asymptotisch stabil ist.

Ziel des Regelungsalgorithmus ist es den Tracking-Fehler im Schubvektor und den Winkelgeschwindigkeiten zu null zu bringen. Dafür werden die Tracking-Fehler definiert:

$$e_t := t_d - t$$

$$e_\omega := \boldsymbol{\omega}_d - \boldsymbol{\omega}$$

Der angewendete Ansatz ist sogenanntes Block-Backstepping. Im Folgenden werden nur die Ergebnisse für die Regelgesetze des Lagereglers gegeben. Die genauen Herleitung können in und gefunden werden. Unter Anwendung des Block-Backstepping Ansatzes ergeben sich die Regelgesetze nun zu:

$$u_2 = T_u^{-1} \mathbf{R}_{BI} (\dot{t}_d + \mathbf{K}_t e_t)$$

$$M_d = \omega \times J\omega + J \left( T_\omega^T R_{BI} e_t + \dot{\omega}_d + K_\omega e_\omega \right)$$

$$T_d = \int_{\tau=t_0}^t u_{2,z} d\tau$$

Diese Regelgesetze garantieren unter geeigneter Wahl der Matrizen  $K_t$  und  $K_\omega$  exponentielle Stabilität der Regelschleife. Die Drehzahl-Kommandos an die Motoren können mit Hilfe der Moore-Penrose Pseudo-Inversen  $B_{v\omega}^+$  berechnet werden:

$$u = B_{v\omega}^+ v_d$$

### 6.5.6.2 Adaptive Control Allocation

Die bisher hergeleiteten Methoden des Baseline-Regler berücksichtigen noch keine Ausfälle oder Degradierung von Propeller. Hierfür wird die Adaptive Control Allocation verwendet. Aufgabe der Control Allocation ist es die gewünschten Pseudo-Controls  $v_d$  optimal auf Motor-Kommandos zu allokieren. Die Adaptive Control Allocation besteht daher aus zwei Komponenten, einer optimierungsbasierten Control Allocation und einer Control Effectiveness Schätzung. Die Control Effectiveness Schätzung wird genauer in Kapitel 6.5.6.3 beschrieben. Die im Folgenden beschriebenen Ergebnisse basieren auf der Veröffentlichung. Um ein Modell mit unbekanntem Aktuatoren-Ausfällen zu ermöglichen, wird das bisherige Modell erweitert:

$$v = B_{v\omega} \Lambda u$$

Hier gibt die Diagonal-Matrix  $\Lambda$  die aktuelle Control Effectiveness der einzelnen Aktuatoren zwischen 0 und 1. Da für reale Flug die Drehzahl nach oben als auch unten limitiert ist, ergibt sich die mögliche Menge aller möglichen Drehzahlen als

$$U := \left\{ u \in \mathbb{R}^6 \mid \omega_{i,\min}^2 \leq u_i \leq \omega_{i,\max}^2, \forall i = 1 \dots 6 \right\}.$$

Daraus ergibt sich die Menge aller mögliche Pseudo-Controls:

$$V := \left\{ v \in \mathbb{R}^4 \mid v = B_{v\omega} \Lambda u, u \in U \right\}$$

Hierbei ist wichtig zu beachten, dass die Größe der Menge  $V$  stark von der Control Effectiveness  $\Lambda$  abhängt.

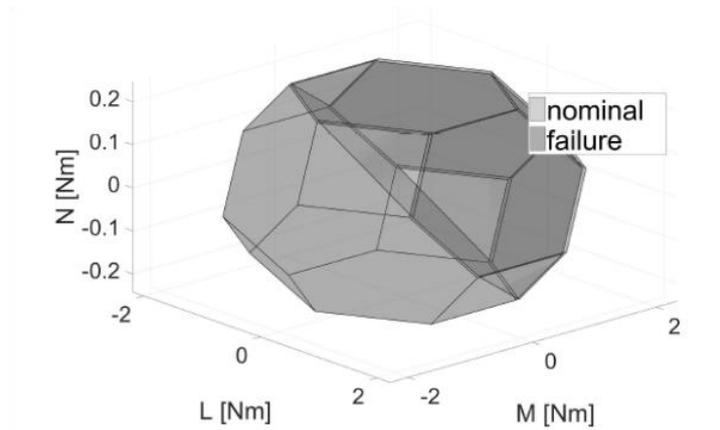


Abbildung 6.5.8: Vergleich der Menge aller möglichen Pseudo-Controls für Schwebeflug für Nominalfall und Ausfall des hinteren rechten Propeller

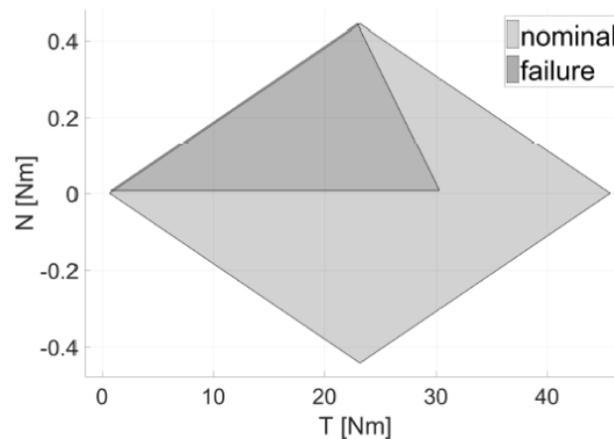


Abbildung 6.5.9: Vergleich der Menge aller möglichen Pseudo-Controls für Schwebeflug für Nominalfall und Ausfall des hinteren rechten Propeller (L=M=0)

In Abbildung 6.5.8 und Abbildung 6.5.9 ist als Beispiel ein Vergleich zwischen den Nominalfall (alle Propeller voll funktionsfähig) und einem degradierten Fall (rechter-hinterer Propeller ausgefallen) dargestellt.

Um auch für einen Ausfall eine optimale Control Allocation zu ermöglichen wurde eine Control Allocation entwickelt, welche auf Optimierungsverfahren basiert. Ziel dieser ist es einen optimalen Input  $u_{cmd}^*$  zu finden, für welchen gilt

$$v_d = B_{v\omega} \hat{\Lambda} u_{cmd}^*$$

Zu beachten ist, dass ab hier die geschätzte Control Effectiveness  $\bar{\Lambda}$  anstatt  $\Lambda$  verwendet wird. Als Optimierungsmethode wird ein gradientenbasiertes Verfahren, sogenanntes Backtracking Line Search, angewendet. In Abbildung 6.5.10 ist eine Ablaufplan des Algorithmus gegeben. Als Kostenfunktion für den Algorithmus wurde

$$J = \frac{1}{2} (\bar{v}_d - \bar{B}_{v\omega} \hat{\Lambda} \bar{u}_{cmd})^T W (\bar{v}_d - \bar{B}_{v\omega} \hat{\Lambda} \bar{u}_{cmd}) + \sum_{i=1}^6 \left( \frac{\frac{\mu}{2}}{(1 - \bar{u}_{i,cmd})^2} + \frac{\frac{\mu}{2}}{\bar{u}_{i,cmd}^2} \right)$$

gewählt.

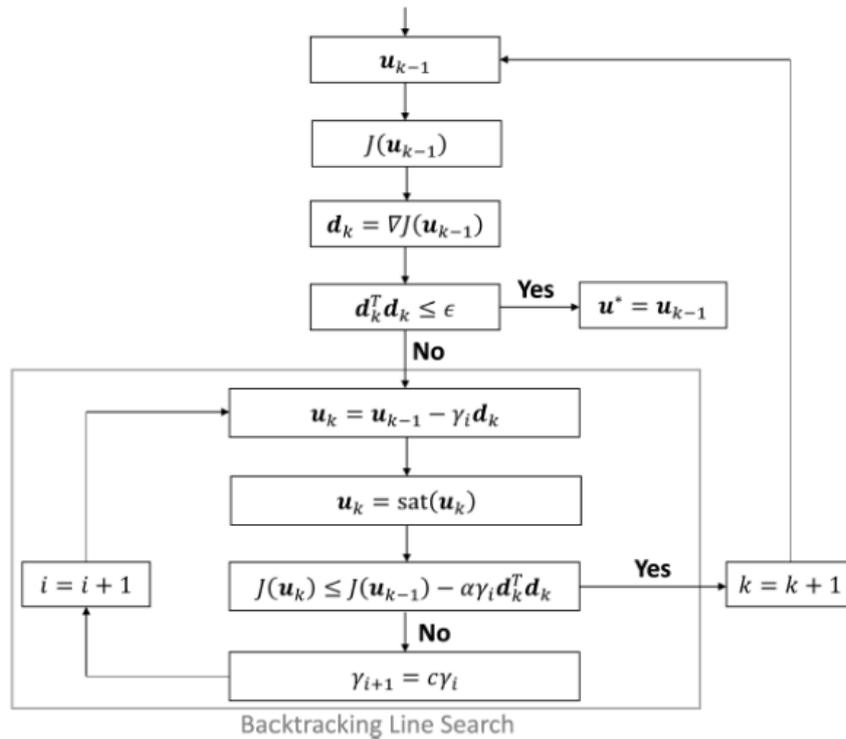


Abbildung 6.5.10: Überblick über Backtracking Line Search

### 6.5.6.3 Control Effectiveness Schätzung

Für die Adaptive Control Allocation, welche in vorherigen Kapitel präsentiert wurde, wird eine Schätzung der aktuellen Control Effectiveness benötigt. Das Verfahren zur Online-Schätzung dieser soll nun in diesem Kapitel präsentiert werden.

Die Schätzung der Control Effectiveness basiert auf dem Concurrent Learning Ansatz, präsentiert in und . Hierfür muss zuerst das Aktuator-Modell umgeschrieben werden:

$$v = B_{v\omega} U \lambda$$

Hier beschreibt  $U = \text{diag}(\omega_1^2, \omega_2^2, \omega_3^2, \omega_4^2, \omega_5^2, \omega_6^2)$  eine Diagonalmatrix und der Vektor  $\lambda$  enthält die Einträge auf der Diagonale von  $\Lambda$ .

Die Update-Regel für den Schätzung der Control Effectiveness ergibt sich unter Anwendung des Concurrent Learning Ansatzes zu:

$$\begin{aligned} \hat{\lambda} &= -\lambda \sum_{i=1}^p U_{f,i}^T B_{y\omega}^T (B_{y\omega} U_{f,i} \hat{\lambda} - y_{f,i}) + Q_\xi w \\ &= -\lambda \sum_{i=1}^p U_{f,i}^T B_{y\omega}^T (B_{y\omega} U_{f,i} \hat{\lambda}) + Q_\xi w \end{aligned}$$

Auf die genaue Herleitung und den Stabilitätsbeweis soll in diesem Bericht mit Rücksicht auf Lesbarkeit verzichtet werden. Diese können in genauer nachvollzogen werden.

## 6.6 Simulation Flugvehikel

Dieses Kapitel teilt sich in zwei Hauptteile auf. In Kapitel 6.6.1 wird ein Simulationsmodell des in Kapitel 6.5 entworfenen Flugvehikels präsentiert. In Kapitel 6.6.2 wird basierend auf den Ergebnissen des VaMEx-Projektes das Simulationsmodell für die Erdatmosphäre weiterentwickelt. Zusätzlich wird der in Kapitel 6.5.5 hergeleitete Flugregler implementiert und getestet.

### 6.6.1 Simulation Flugvehikel Mars

Da Testen von Flugalgorithmen vor Beginn einer Mission am Mars nicht möglich ist, müssen andere Methoden gefunden werden, um einen zuverlässigen Ablauf der Mission zu ermöglichen. Besonders bei Fluggeräten nimmt der Regelungsalgorithmus zur Stabilisierung des Fluggeräts eine zentrale Rolle ein. Um die entwickelten Algorithmen testen zu können, wird daher eine möglichst genaue Abbildung der Dynamik in einer Simulation benötigt.

Dieses Kapitel beschäftigt sich nun mit der Herleitung eines solchen Simulationsmodells anhand des in Kapitel 6.5 entwickelten Vehikelkonzepts. In Kapitel 6.6.1.1 werden die Umgebungsbedingungen im Valles Marineris hergeleitet. Anschließend werden in Kapitel 6.6.1.2 und 6.6.1.3 die Modelle des Flugvehikels und des Ballons erstellt. Kapitel 6.6.1.4 beschäftigt sich mit der Modellierung der Kopplung zwischen Flugvehikel und Ballon.

#### 6.6.1.1 Modellierung der Umgebungsbedingungen im Valles Marineris

Zur Modellierung der Marsumgebung wurden Daten der Online-Datenbank der ESA verwendet. Die Modellierung der Mars Umgebung beinhaltet die Modellierung des Geoids, der Atmosphäre und der Gravitation (Abbildung 6.6.1). Für den Entwurf und die Analyse des Flugreglers sind besonders die Atmosphäre und Gravitationskräfte von Bedeutung. Daher wird in diesem Bericht hauptsächlich auf die Modellierung dieser eingegangen. Für eine ausführlichere Modellierung soll auf die Arbeit verwiesen werden.

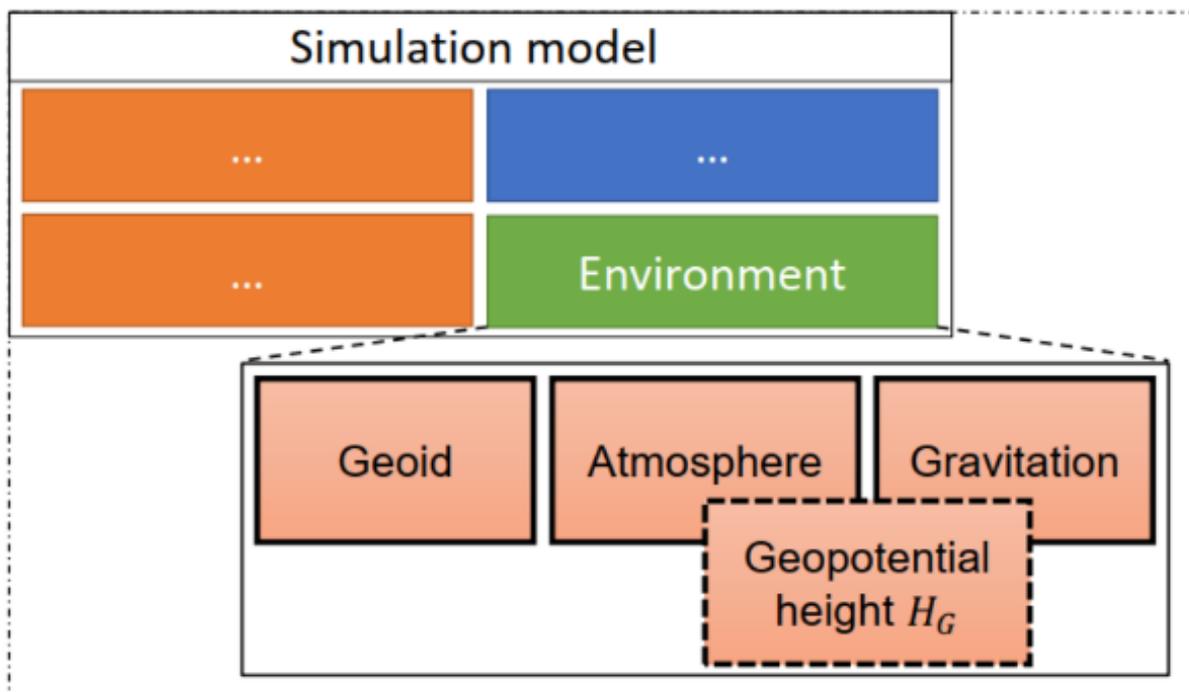


Abbildung 6.6.1: Struktur der Modellierung der Umgebungsbedingungen

Für die Gravitationsmodellierung wurde ein 1-Dimensionales (1D) Modell verwendet:

$$(g^{CG})_O = \left[ g_0 \cdot \left( \frac{H_G}{h_M} \right)^2 \right]_O$$

Dafür wird die Gravitationskonstant  $g_0$  benötigt. Die Berechnung der Gravitationskonstante erfolgt mit zur Hilfenahme der Newtonschen Gravitationskonstant  $G$ :

$$g_0 = \frac{G \cdot M_{mars}}{r_M^2} = \frac{6.67408 \cdot 10^{-11} \frac{m^3}{kg \cdot s^2} \cdot 6.4171 \cdot 10^{23} kg}{(3394600 m)^2} = 3.717 \frac{m}{s^2}$$

Hier wurde für den Mars-Radius  $r_M$  der Radius am Valles Marineris verwendet, da dies für das Projekt der interessante Bereich ist. Vergleicht man die Gravitationskonstant des Mars mit der der Erde

$$\frac{g_{0,Mars_{eq}}}{g_{0,Earth_{eq}}} = \frac{3.713 m/s^2}{9.80665 m/s^2} = 0.379$$

erkennt man, dass die Gravitationskraft auf dem Mars nur etwa ein Drittel der der Erde ist. Zusammenfassend sind die Ergebnisse für ein 1D-Modell der Gravitation in Tabelle 6.6.1 gegeben.

Mars 1-D Graviationsmodell		
Parameter	Notation	Wert
Geocentric radius at Valles Marineris	$r_M$	3394600 m
Height above reference ellipsoid	$h_M$	arbitrary
Geopotential height	$H_G$	equation (3-2)
Universal gravitational constant	$G$	$6.67408 \cdot 10^{-11} m^3/kg s^2$
Mass of Mars	$M_{Mars}$	$6.4171 \cdot 10^{23} kg$

Tabelle 6.6.1: Parameter des 1-D Gravitationsmodell

Analog zur Standard-Atmosphäre der Erde soll nun ein Modell für die Standard-Atmosphäre des Mars hergeleitet werden. Im Anschließendenden werden nur die Ergebnisse dargestellt. Für genauere Herleitungen soll wiederum auf verwiesen werden. Eine Zusammenfassung der Ergebnisse ist in Abbildung 6.6.2 gegeben. Die Mars-Atmosphäre wurde dabei in zwei Bereiche (untere und obere) unterteilt. Dadurch können vereinfachende Formeln angegeben werden, wobei trotzdem eine hohe Qualität der Modellierung erreicht werden kann.

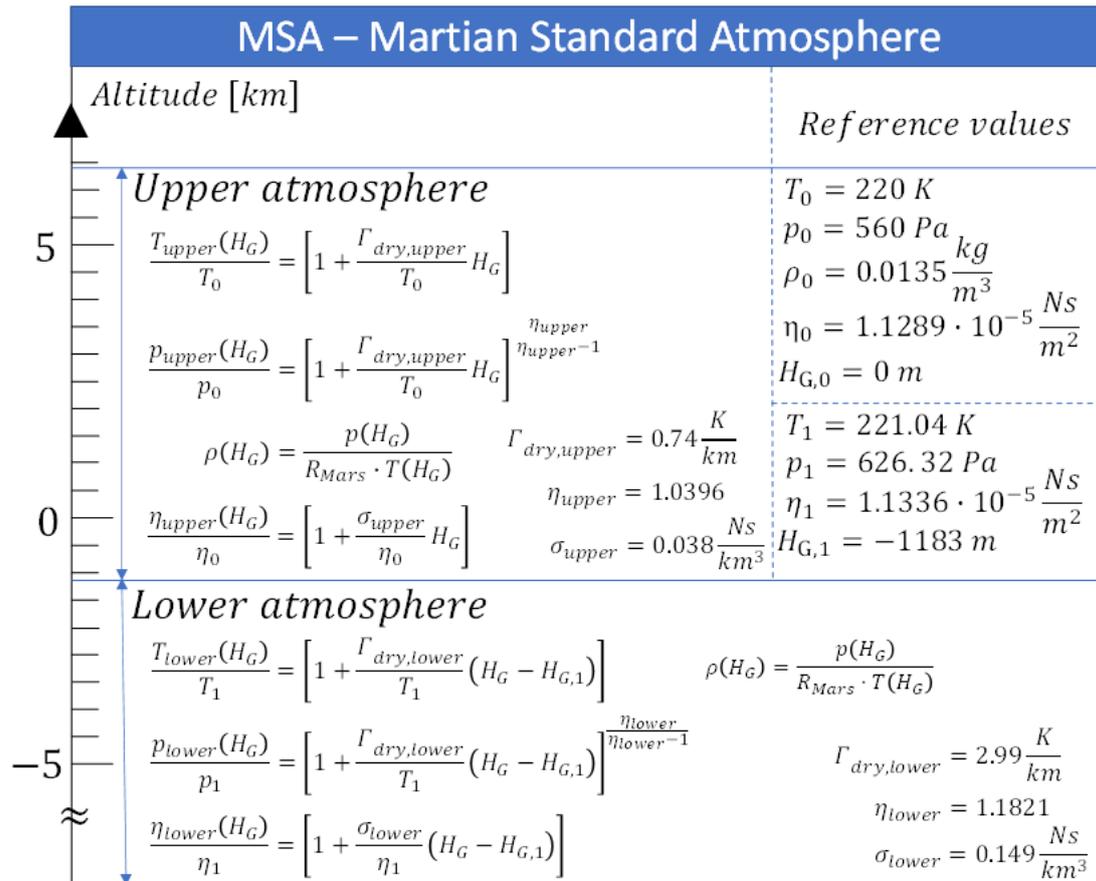


Abbildung 6.6.2: Atmosphären-Modell der Marsatmosphäre

Vergleicht man das Modell mit Messdaten der Mars Climate Database für die Dichte (Abbildung 6.6.3), erkennt man, dass hier eine gute Korrelation gegeben ist.

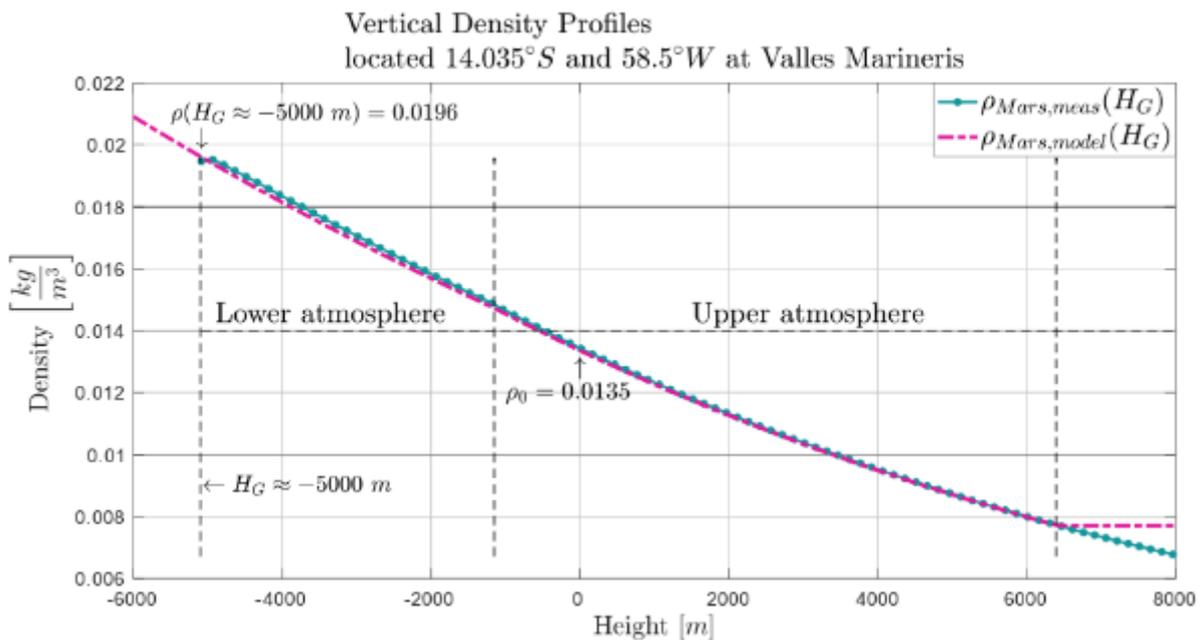


Abbildung 6.6.3: Vergleich der Dichte für das hergeleitete Modell mit der Mars Climate Database

### 6.6.1.2 Modellierung des Flugvehikels

Für die Modellierung müssen alle Kräfte und Momente bestimmt werden, welche auf das Flugvehikel wirken:

$$\sum \left( \mathbf{F}^G \right)_B = \left( \mathbf{F}_G^G \right)_B + \left( \mathbf{F}_A^G \right)_B + \left( \mathbf{F}_P^G \right)_B + \left( \mathbf{F}_{Rope}^G \right)_B$$

$$\sum \left( \mathbf{M}^G \right)_B = \underbrace{\left( \mathbf{M}_G^G \right)_B}_{=0} + \left( \mathbf{M}_A^G \right)_B + \left( \mathbf{M}_P^G \right)_B + \left( \mathbf{M}_{Rope}^G \right)_B$$

Hierbei können folgende Haupteinflüsse bestimmt werden:

- Gravitation (Index  $G$ )
- Aerodynamik (Index  $A$ )
- Propeller (Index  $P$ )
- Seil (Index  $Rope$ , siehe Kapitel 6.6.1.4)

#### 6.6.1.2.1 Gravitation

Die Gravitationskraft wird mit Hilfe des im vorherigen Kapitel hergeleiteten Gravitationsmodell bestimmt. Es wird dabei angenommen, dass die Kraft im Schwerpunkt des Fluggeräts liegt und dadurch kein Moment ausübt. Dadurch kann die Gravitationskraft mit

$$\left( \mathbf{F}_G^G \right)_B = T_{BO} \left( \mathbf{F}_G^G \right)_O = m \cdot g_0 \cdot \left( \frac{H_G}{h_M} \right)^2 \cdot T_{BO} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}_O$$

bestimmt werden.

#### 6.6.1.2.2 Aerodynamik

Die Aerodynamik des Fluggeräts kann in erster Näherung nur mit Hilfe von Widerstandskräften bestimmt werden. Diese Annahme kann durch das Form und den Aufbau des Flugvehikels gerechtfertigt werden. Bei dem hergeleiteten Design des Fluggeräts handelt es sich um ein Multikopter-System, welches über keine aerodynamischen Auftriebsflächen verfügt. Daher können die aerodynamisch wirkenden Kräfte bestimmt werden:

$$\left( \mathbf{F}_A^R \right)_B = -\bar{q} \begin{pmatrix} C_{D,x} & 0 & 0 \\ 0 & C_{D,y} & 0 \\ 0 & 0 & C_{D,z} \end{pmatrix} \left( \mathbf{V}_A^R \right)_B^M$$

Für das gegebene Design wurden die Koeffizienten bestimmt und es ergibt sich der aerodynamische Kraftvektor:

$$\left( \mathbf{F}_A^R \right)_B = -\frac{1}{2} \rho (h^R) \left\| \left( \mathbf{V}_A^R \right)_B^M \right\| \begin{pmatrix} 0.088 & 0 & 0 \\ 0 & 0.088 & 0 \\ 0 & 0 & 0.088 \end{pmatrix} T_{BA} \left( \mathbf{V}_A^R \right)_A^M$$

#### 6.6.1.2.3 Propeller

Zur Modellierung der Propellerkräfte wurde das Blattelement-Verfahren angewendet. Dieses

wird in Kapitel 6.6.2 hergeleitet und beschrieben. In diesem Kapitel soll kurz darauf eingegangen werden, wie die hergeleiteten Ergebnisse in die Simulation eingebunden werden können.

Die Blattelement-Methode liefert als Ergebnis Koeffizienten für die verschiedenen Krafrichtungen und Momente. Diese Koeffizienten sind nicht über den gesamten Flugbereich konstant und müssen für die jeweilige Flugbedingung berechnet werden. Diese Berechnung kann jedoch nicht in Echtzeit durchgeführt werden, was für Flugregelungssimulationen aber eine wichtige Voraussetzung ist. Daher wurde hier ein anderer Ansatz verwendet. Die Koeffizienten wurden für einen gegebenen Propeller und alle möglichen Flugbedingungen „offline“, das heißt vor dem Start der Simulation, berechnet und in Tabellen abgespeichert. Diese Tabellen werden anschließend in der Simulation integriert und die daraus resultierenden Kräfte und Momente bestimmt:

$$\begin{pmatrix} F_P^{P_i} \end{pmatrix}_{A_{proj}} = \begin{pmatrix} X_P^{P_i} \\ Y_P^{P_i} \\ Z_P^{P_i} \end{pmatrix}_{A_{proj}} = \frac{\rho(H_G) (2r_{prop})^4 (\omega_z^{P_i Rot})^2}{4\pi^2} \begin{pmatrix} C_x \\ C_y \\ C_z \end{pmatrix}_{A_{proj}}$$

$$\begin{pmatrix} M_P^{P_i} \end{pmatrix}_{A_{proj}} = \begin{pmatrix} L_P^{P_i} \\ M_P^{P_i} \\ N_P^{P_i} \end{pmatrix}_{A_{proj}} = \frac{\rho(H_G) (2r_{prop})^5 (\omega_z^{P_i Rot})^2}{4\pi^2} \begin{pmatrix} C_l \\ C_m \\ C_n \end{pmatrix}_{A_{proj}}$$

Hier sind die Koeffizienten  $C_x, C_y, C_z, C_l, C_m, C_n$  jedoch keine Konstanten sondern

$$C_x = C_x(\omega_{rot}, V_{abs}, \alpha)$$

$$C_y = C_x(\omega_{rot}, V_{abs}, \alpha)$$

$$C_z = C_x(\omega_{rot}, V_{abs}, \alpha)$$

$$C_l = C_x(\omega_{rot}, V_{abs}, \alpha)$$

$$C_m = C_x(\omega_{rot}, V_{abs}, \alpha)$$

$$C_n = C_x(\omega_{rot}, V_{abs}, \alpha).$$

Hier beschreibt  $\omega_{rot}$  die Rotationsgeschwindigkeit der Propellers,  $V_{abs}$  gibt die absolute Anströmgeschwindigkeit des Propellers und  $\alpha$  beschreibt den Anstellwinkel der Propellerebene relative zur Anströmgeschwindigkeit.

### 6.6.1.3 Modellierung des Ballons

Da die Ergebnisse der Design-Analyse des Flugvehikels in Kapitel 6.5 zu einem hybriden Konzept geführt haben, welches einen Ballon enthält, wird für die Simulation auch ein Modell des Ballons benötigt.

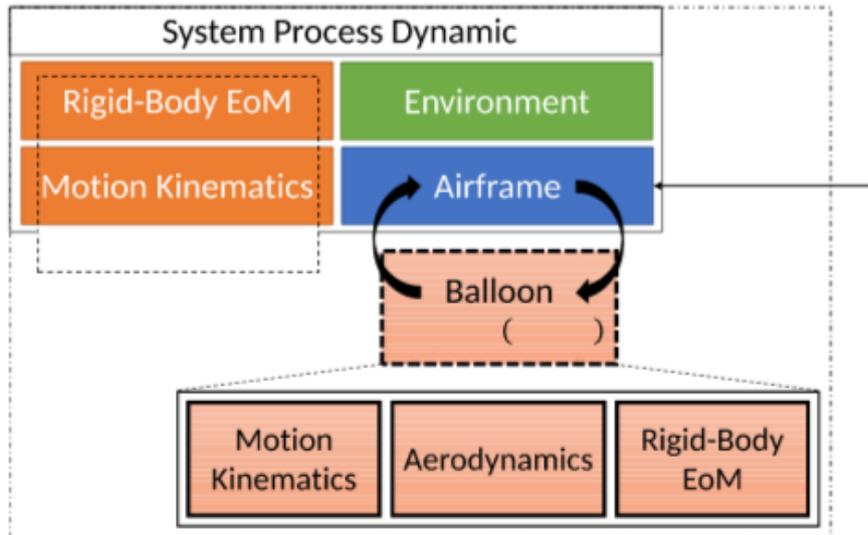


Abbildung 6.6.4: Einordnung des Ballons in das Gesamtsimulationsmodell

In Abbildung 6.6.4 ist eine Einordnung des Ballons in das Gesamtsimulationsmodell dargestellt. Bei dem Ballon handelt sich um einen unabhängigen Körper, welcher wiederum durch eine 12DOF (12 Freiheitsgrade) Dynamik beschrieben werden kann. Um diese Modellierung etwas zu vereinfachen, wurde die Rotationsdynamik des Ballons vernachlässigt und nur die Translationsdynamik modelliert. Dies kann gerechtfertigt werden, da die erzeugten Ballonkräfte und die Interaktion mit dem Flugvehikel hauptsächlich von der Relativposition und nicht von der Lage des Ballons abhängen.

Die Bewegungsgleichungen des Ballons sind in Abbildung 6.6.5 dargestellt. Hier ist zu erkennen, dass die aerodynamische Kraft (außer der Seilkraft, welche hier noch vernachlässigt wird – siehe Kapitel 6.6.1.4) als einzige die Bewegungsgleichungen beeinflusst. Sie kann in zwei Komponenten aufgeteilt werden:

$$\left( \mathbf{F}_{A_{Bal}}^Q \right)_{B_{Bal}} = \left( \mathbf{F}_{Lift}^Q \right)_{B_{Bal}} + \left( \mathbf{F}_{Drag}^Q \right)_{B_{Bal}}$$

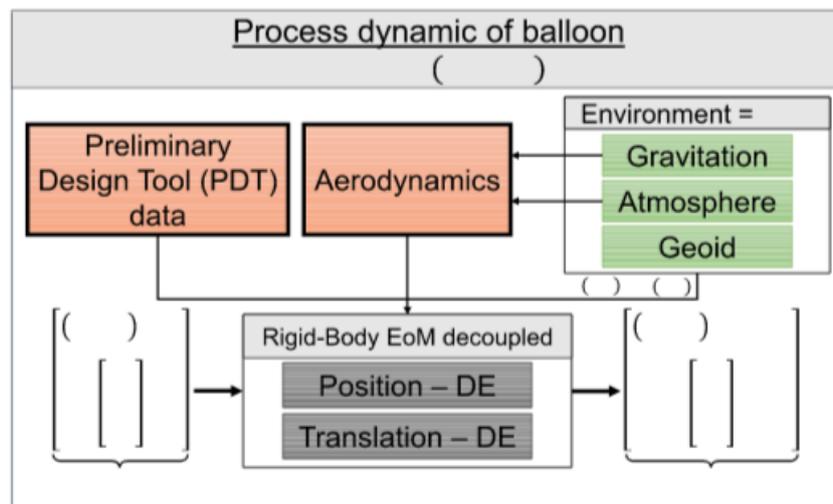


Abbildung 6.6.5: Bewegungsgleichungen des Ballons

Die Gesamt-Auftriebskraft des Ballons  $\left( \mathbf{F}_{Lift}^Q \right)_{B_{Bal}}$  ist gegeben durch die Netto-Auftriebskraft

$F_{L,net}^Q$  minus der Gravitationskraft der Hülle des Ballons  $F_{G,Envelope}^Q$ :

$$F_{Lift}^Q = F_{L,net}^Q - F_{G,Envelope}^Q$$

Diese kann nun bestimmt werden zu

$$F_{Lift}^Q = \left[ \left( \rho_{Mars}(h^R) - \frac{p(h^R)}{R_{He} \cdot T(h^R)} \right) \cdot V_{Bal} - \rho_{A,Bal} \cdot S_{Bal} \right] \cdot g_{Mars}(h^R).$$

Die Widerstandskraft des Ballons kann ähnlich zu Kapitel 6.6.1.2.2 bestimmt werden

$$\left( \mathbf{F}_{Drag}^Q \right)_{B_{Bal}} = -\bar{q} \begin{pmatrix} C_{D,x} & 0 & 0 \\ 0 & C_{D,y} & 0 \\ 0 & 0 & C_{D,z} \end{pmatrix} \left( \mathbf{v}_{A_{Bal}}^Q \right)_{B_{Bal}}^{M_{Bal}},$$

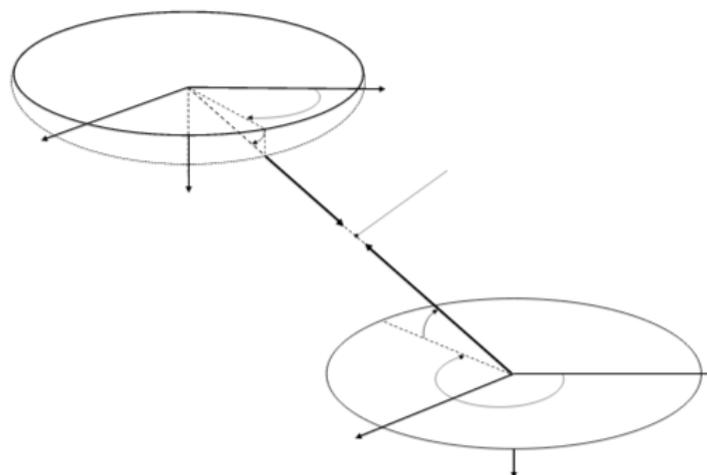
wobei der Staudruck  $\bar{q}$  durch

$$\bar{q} = \frac{1}{2} \rho(h^R) \sqrt{\left( \left( u_{A_{Bal}}^Q \right)_{B_{Bal}}^{M_{Bal}} \right)^2 + \left( \left( v_{A_{Bal}}^Q \right)_{B_{Bal}}^{M_{Bal}} \right)^2 + \left( \left( w_{A_{Bal}}^Q \right)_{B_{Bal}}^{M_{Bal}} \right)^2}$$

gegeben ist.

#### 6.6.1.4 Modellierung der Flugvehikel-Ballon-Kopplung

Nachdem in den vorherigen Kapiteln die Modellierung des Flugvehikels und des Ballons diskutiert wurden, soll nun auf die Interaktion beider eingegangen werden. Um die beiden Körper in der Simulation zu koppeln, wurde das Seil als Feder-Dämpfer System modelliert. Die Modellierung als elastisches Seil ist erforderlich, da eine starre Kopplung zu einem numerisch nicht lösbar System führen.



**Abbildung 6.6.6: Überblick der verwendeten Koordinatensystem zur Modellierung der Flugvehikel und Ballon Kopplung**

Zur Modellierung der Federkraft im Seil ist die relative Position von Fluggerät zu Ballon, das heißt der Punkt  $R$  und  $Q$  erforderlich (Abbildung 6.6.6). Zusätzlich wird für die Berechnung der

Dämpfungskraft im Seil die relative Geschwindigkeit dieser beiden Punkte benötigt. Damit kann die Seilkraft berechnet werden:

$$(F_{Rope}) = (F_{Spring}) + (F_{Damper}) = k_{Rope} \cdot \Delta l - d_{Rope} \cdot \Delta v$$

Da der Feder- und Dämpfungskoeffizient des Seils nicht gegeben sind, wurde ein Ablaufplan zu Bestimmung dieser Werte erstellt (Abbildung 6.6.7):

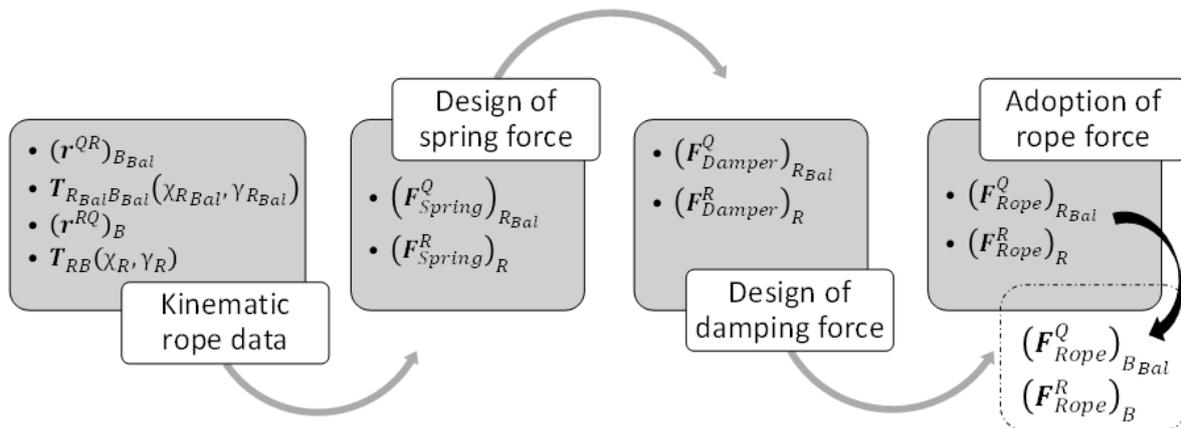


Abbildung 6.6.7: Ablaufplan zur Bestimmung der Feder- und Dämpfungskoeffizienten

Nachdem die kinematischen Daten wie Position und Geschwindigkeit bestimmt wurden, wird zuerst die Federkraft berechnet. Anschließend wird die Dämpfungskraft bestimmt und abschließend noch eine Transformation der Seilkraft vorgenommen. Die resultierenden, wirkenden Seilkräfte am Flugvehikel sind gegeben als

$$\begin{pmatrix} F_{Rope}^R \end{pmatrix}_B = \varepsilon \begin{pmatrix} \cos(\gamma_R) \cos(\chi_R) [F_{Spring}^R - F_{Damper}^R] \\ \cos(\gamma_R) \sin(\chi_R) [F_{Spring}^R - F_{Damper}^R] \\ -\sin(\gamma_R) [F_{Spring}^R - F_{Damper}^R] \end{pmatrix}_B$$

Im Körperfesten Koordinatensystem des Flugvehikel und im Koordinatensystem des Ballons als

$$\begin{pmatrix} F_{Rope}^Q \end{pmatrix}_{B_{Bal}} = \varepsilon \begin{pmatrix} \cos(\gamma_{R_{Bal}}) \cos(\chi_{R_{Bal}}) [F_{Spring}^Q - F_{Damper}^Q] \\ \cos(\gamma_{R_{Bal}}) \sin(\chi_{R_{Bal}}) [F_{Spring}^Q - F_{Damper}^Q] \\ \sin(\gamma_{R_{Bal}}) [F_{Spring}^Q - F_{Damper}^Q] \end{pmatrix}_{B_{Bal}}$$

Dabei berechnet sich die Federkraft zu

$$\begin{pmatrix} F_{Spring}^R \end{pmatrix}_R = \begin{pmatrix} \varepsilon \cdot F_{Spring}^R \\ 0 \\ 0 \end{pmatrix}_R,$$

wobei zusätzlich gilt

$$\begin{pmatrix} F_{Spring}^R \end{pmatrix} = k_{Rope} \cdot \left( \left\| \begin{pmatrix} r^{RQ} \end{pmatrix}_B \right\| - l_0 \right)$$

und

$$\varepsilon = \begin{cases} 1, & F_{Spring}^Q > 0 \wedge F_{Spring}^R > 0 \\ 0, & F_{Spring}^Q \leq 0 \wedge F_{Spring}^R \leq 0 \end{cases}.$$

Die Dämpferkraft ergibt sich zu

$$\begin{pmatrix} F_{Damper}^R \end{pmatrix} = \begin{pmatrix} F_{Damper}^Q \end{pmatrix} = d_{Rope} \cdot \left[ \begin{pmatrix} u_K^R \end{pmatrix}_R^M + \begin{pmatrix} u_{KBal}^Q \end{pmatrix}_{R_{Bal}}^M \right].$$

Auch hier wird die Kraft wieder nur aktiviert, wenn das Seil gespannt ist. Das heißt

$$\begin{pmatrix} F_{Damper}^R \end{pmatrix}_R = \begin{pmatrix} -F_{Damper}^R \cdot \varepsilon \\ 0 \\ 0 \end{pmatrix}_R.$$

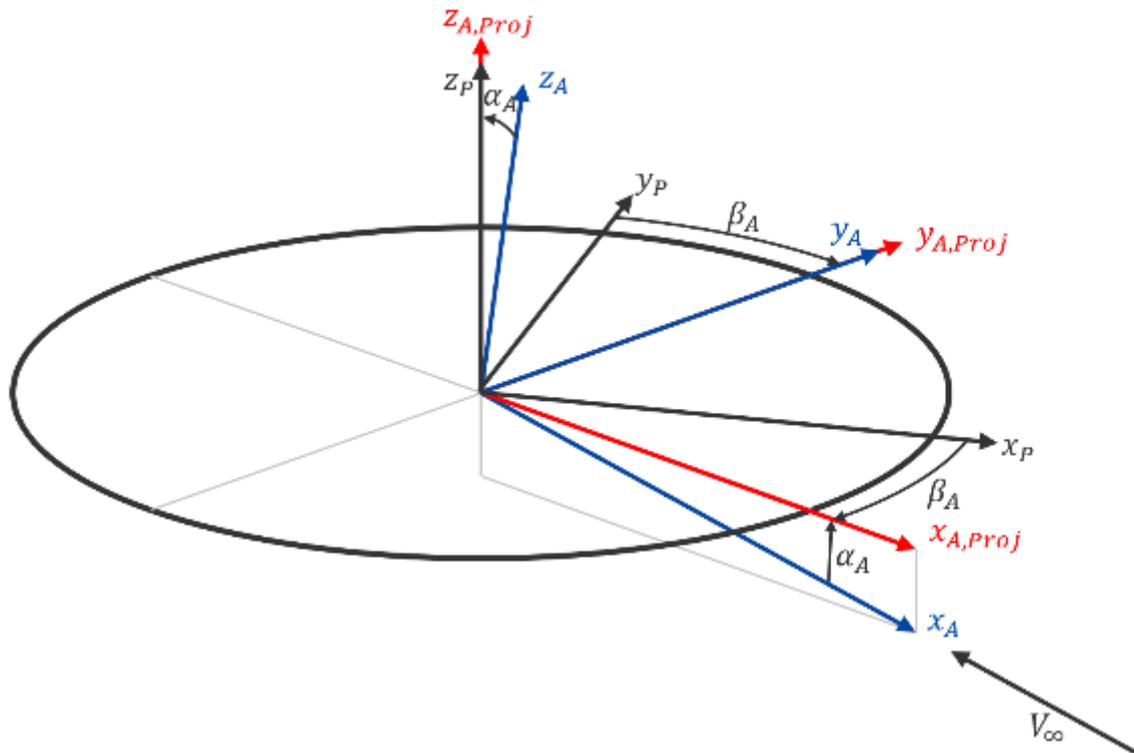
## 6.6.2 Simulation Flugvehikel Erde

Basierend auf den Ergebnissen des VaMEx-Projektes wurde die Simulationsumgebung weiter verfeinert und zusätzlicher Detailgrad implementiert. Da ein Haupteinfluss auf die Dynamik des Multikopters die Kräfte des Propellers sind wurde eine neues Tool zur Bestimmung der Propellerkräfte implementiert, welches in Kapitel 6.6.2.1 vorgestellt wird. Anschließend wird der Aufbau des Simulations-Frameworks, bestehend aus Simulationsmodell und Regler, dargestellt.

### 6.6.2.1 Modellierung von Propeller-Kräften

Für die Modellierung der Propeller-Kräfte wurde die Blatt-Element-Theorie mit einer Kopplung der Impulstheorie, bekannt unter dem Namen Blade-Element-Momentum-Theory (BEMT) verwendet. Die Grundlagen der verwendeten Theorien sind in und enthalten und die folgenden Herleitungen sind davon abgeleitet.

Bevor mit der Herleitung der Theorie begonnen werden kann, werden die benötigten Koordinaten Systeme definiert.



**Abbildung 6.6.8: Koordinatensysteme für die Propeller-Koeffizientenberechnung**

In Abbildung 6.6.8 sind alle benötigten Koordinatensysteme abgebildet.  $P$  beschreibt dabei das Propeller-feste System,  $A$  das aerodynamische und  $A,Proj$  das in die Propeller-Ebene projizierte System. Die dazugehörigen Winkel zwischen den Systemen sind mit dem aerodynamischen Anstellwinkel relativ zur Propeller-Ebene  $\alpha_A$  und dem aerodynamischen Schiebewinkel  $\beta_A$  gegeben.

Zusätzlich ist in Abbildung 6.6.9 das Blattkoordinatensystem abgebildet. Dieses rotiert mit dem Propellerblatt um die  $z_P$ -Achse des Propellersystems. Der dazugehörige Rotationswinkel  $\psi_{Bl}$  ist relativ zur  $x_{A,Proj}$ -Achse des projizierten aerodynamischen Systems gegeben.

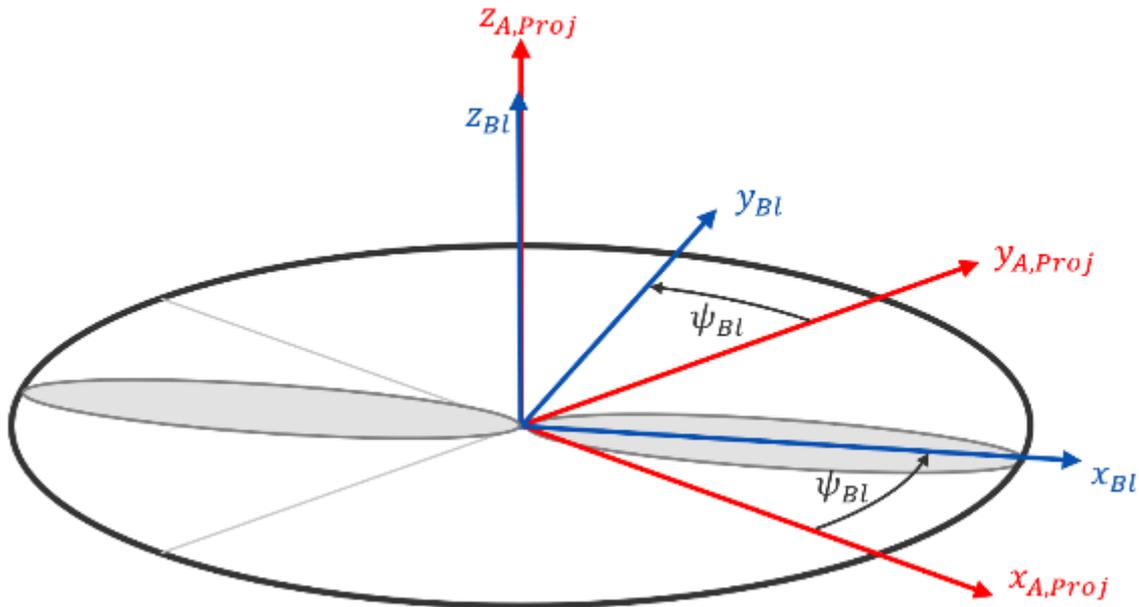


Abbildung 6.6.9: Definition des Blatt-Koordinatensystems

Bei der Blatt-Element-Theorie wird das Propeller-Blatt radial in unabhängige Elemente geteilt. Die Anströmgeschwindigkeit an einem Element ist abhängig von der freien Anströmung  $V_\infty$ , dem Anstellwinkel  $\alpha_A$  und der Rotationsgeschwindigkeit des Propellers  $\omega_z^{PBl}$ :

$$\begin{aligned} \left( \mathbf{V}_A^{Rij} \right)_{Bl} &= (V_A)_{Bl}^P + (\omega^{PBl})_{Bl} \times \left( \mathbf{r}^{PRij} \right)_{Bl} \\ \left( \mathbf{V}_A^{Rij} \right)_{Bl} &= M_{BlA_{Proj}}^j (V_A)_{A_{Proj}}^P + (\omega^{PBl})_{Bl} \times \left( \mathbf{r}^{PRij} \right)_{Bl} \\ \left( \mathbf{V}_A^{Rij} \right)_{Bl} &= M_{BlA_{Proj}}^j (\psi_{Bl,j}) \begin{bmatrix} V_\infty \cos \alpha_A \\ 0 \\ V_\infty \sin \alpha_A \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \omega_z^{PBl} \end{bmatrix}_{Bl} \times \begin{bmatrix} r_i \\ 0 \\ 0 \end{bmatrix}_{Bl} \end{aligned}$$

Mit Hilfe der Geschwindigkeit  $\left( \mathbf{V}_A^{Rij} \right)_{Bl}$  am Element  $ij$  können nun die Anströmbedingungen am Blattelement dargestellt werden (Abbildung 26).

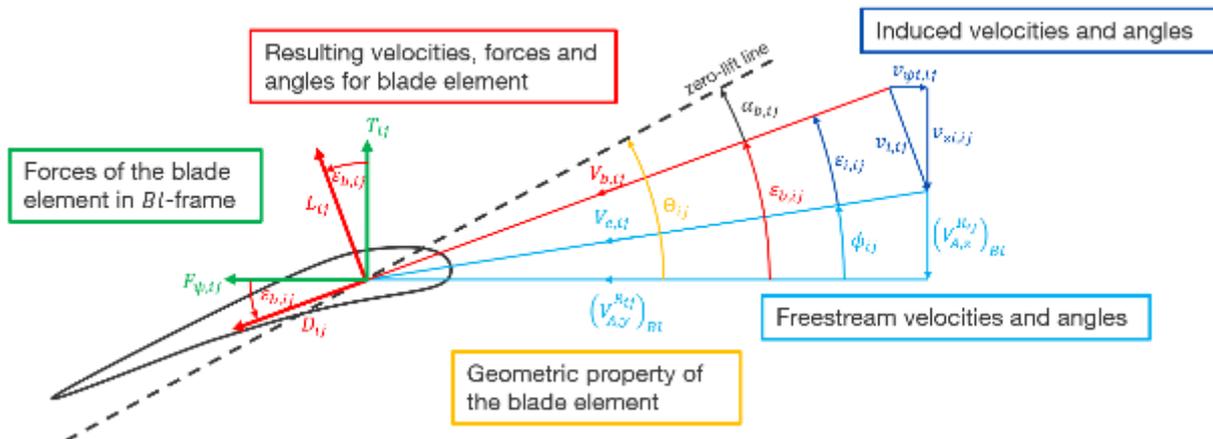


Abbildung 6.6.10: Anströmbedingungen an einem Blattelement des Propellers

In Abbildung 6.6.10 beschreibt  $v_i$  die induzierte Geschwindigkeit des Propellers und  $\Theta_{ij}$  den geometrischen Pitch-Winkel des Elements.  $L_{ij}$ ,  $D_{ij}$  geben die Auftrieb- und Widerstandskraft des Elements,  $T_{ij}$ ,  $F_{\psi,ij}$  die Schub- und Blattumfangskraft des Blattelements. Für die Anwendung der BEMT kann der Schub mit Hilfe der Impulstheorie eines Propellers zu

$$T_{MT,ij} = 2\rho A_{ij} v_{i,ij} \sqrt{\left(V_{A,x}^P\right)_{A_{proj}}^2 + \left(\left(V_{A,z}^P\right)_{A_{proj}} - v_{i,ij}\right)^2}$$

berechnet werden. Zusätzlich kann der Schub mit Hilfe der Blatt-Element-Theorie bestimmt werden:

$$T_{BET,ij} = \frac{N_B}{4\pi} \rho V_{b,ij}^2 \cdot c_{b,ij} \cdot \Delta r \cdot \Delta \psi \cdot \left( C_L(\alpha_{b,ij}, Re_{ij}, Ma_{ij}) \cdot \cos \varepsilon_{b,ij} - C_D(\alpha_{b,ij}, Re_{ij}, Ma_{ij}) \cdot \sin \varepsilon_{b,ij} \right)$$

Setzt man nun beide Gleichungen für den Schub gleich, erhält man:

$$\begin{aligned} & 2\rho r_i \Delta \psi \Delta r v_{i,ij} \sqrt{\left(V_{A,x}^P\right)_{A_{proj}}^2 + \left(\left(V_{A,z}^P\right)_{A_{proj}} - v_{i,ij}\right)^2} \\ &= \frac{N_B}{4\pi} \rho V_{b,ij}^2 c_{b,ij} \Delta r \Delta \psi \left( C_L(\alpha_{b,ij}, Re_{ij}, Ma_{ij}) \cos \varepsilon_{b,ij} \right. \\ & \quad \left. - C_D(\alpha_{b,ij}, Re_{ij}, Ma_{ij}) \sin \varepsilon_{b,ij} \right) \end{aligned}$$

In dieser Gleichung ist die einzige Unbekannte die induzierte Geschwindigkeit  $v_i$ , wobei zu beachten ist, dass gilt

$$\begin{aligned} V_{b,ij}(v_{i,ij}) &= \frac{\left(V_{A,y}^{R_{ij}}\right)_{Bl}}{\cos \varepsilon_{b,ij}(v_{i,ij})}, \\ \alpha_{b,ij}(v_{i,ij}) &= \Theta_{ij} - \varepsilon_{b,ij}(v_{i,ij}), \\ \varepsilon_{b,ij}(v_{i,ij}) &= \arctan \left( \frac{\left(V_{A,z}^{R_{ij}}\right)_{Bl} + v_{i,ij}}{\left(V_{A,y}^{R_{ij}}\right)_{Bl}} \right). \end{aligned}$$

Die sich ergebende, nicht-lineare Gleichung kann nun mit einem numerischen Gleichungslöser (z.B. Newton-Raphson) nach  $v_i$  gelöst werden. Anschließend kann unter Zuhilfenahme der Blatt-Element-Gleichungen die Kräfte und Moment am Propeller-Element gelöst werden:

$$\begin{aligned} L_{ij} &= \frac{1}{2} \cdot \rho \cdot V_{b,ij}^2 \cdot c_{b,ij} \cdot C_L(\alpha_{b,ij}, Re_{ij}, Ma_{ij}) \cdot \Delta r \\ D_{ij} &= \frac{1}{2} \cdot \rho \cdot V_{b,ij}^2 \cdot c_{b,ij} \cdot C_D(\alpha_{b,ij}, Re_{ij}, Ma_{ij}) \cdot \Delta r \end{aligned}$$

$$M_{ij} = \frac{1}{2} \cdot \rho \cdot V_{b,ij}^2 \cdot c_{b,ij}^2 \cdot C_m(\alpha_{b,ij}, Re_{ij}, Ma_{ij}) \cdot \Delta r$$

Der Schub und die Umfangskraft am Element ergeben sich zu

$$\begin{aligned} T_{ij} &= L_{ij} \cos \varepsilon_{b,ij} - D_{ij} \sin \varepsilon_{b,ij} \\ F_{\psi,ij} &= L_{ij} \sin \varepsilon_{b,ij} + D_{ij} \cos \varepsilon_{b,ij} \end{aligned}$$

Die Kräfte im Blatt-System eines Blatt-Elements können nun bestimmt werden:

$$(\mathbf{F}_{ij})_{Bl} = \begin{bmatrix} 0 \\ -F_{\psi,ij} \\ T_{ij} \end{bmatrix}, \quad (\mathbf{M}_{ij})_{Bl} = \begin{bmatrix} M_{ij} \\ 0 \\ 0 \end{bmatrix} + \begin{pmatrix} r_i^{PR_{ij}} \end{pmatrix}_{Bl} \times (\mathbf{F}_{ij})_{Bl}$$

Nachdem diese in das projizierte aerodynamische System transformiert wurden, können die Gesamtkräfte und -momente am Propeller durch Summation berechnet werden:

$$(\mathbf{F})_{A_{Proj}} = \frac{N_B}{2\pi} \sum_j \sum_i (\mathbf{F}_{ij})_{A_{Proj}}, \quad (\mathbf{M})_{A_{Proj}} = \frac{N_B}{2\pi} \sum_j \sum_i (\mathbf{M}_{ij})_{A_{Proj}}$$

Es ist zu beachten, dass zur Bestimmung der korrekten Kräfte und Momente noch mit der Blattanzahl des Propellers geteilt durch eine Umdrehung  $\frac{N_B}{2\pi}$  normiert werden muss.

### 6.6.2.2 Aufbau des Closed-Loop Simulations-Frameworks

Zur Entwicklung des Flugreglers wurde ein Modell-basierter Ansatz gewählt. Um diesen Ansatz gerecht zu werden, wird eine Simulationsumgebung benötigt. Diese wurde in MATLAB/Simulink implementiert. Der Aufbau dieser Simulationsumgebung ist in Abbildung 6.6.11 dargestellt.

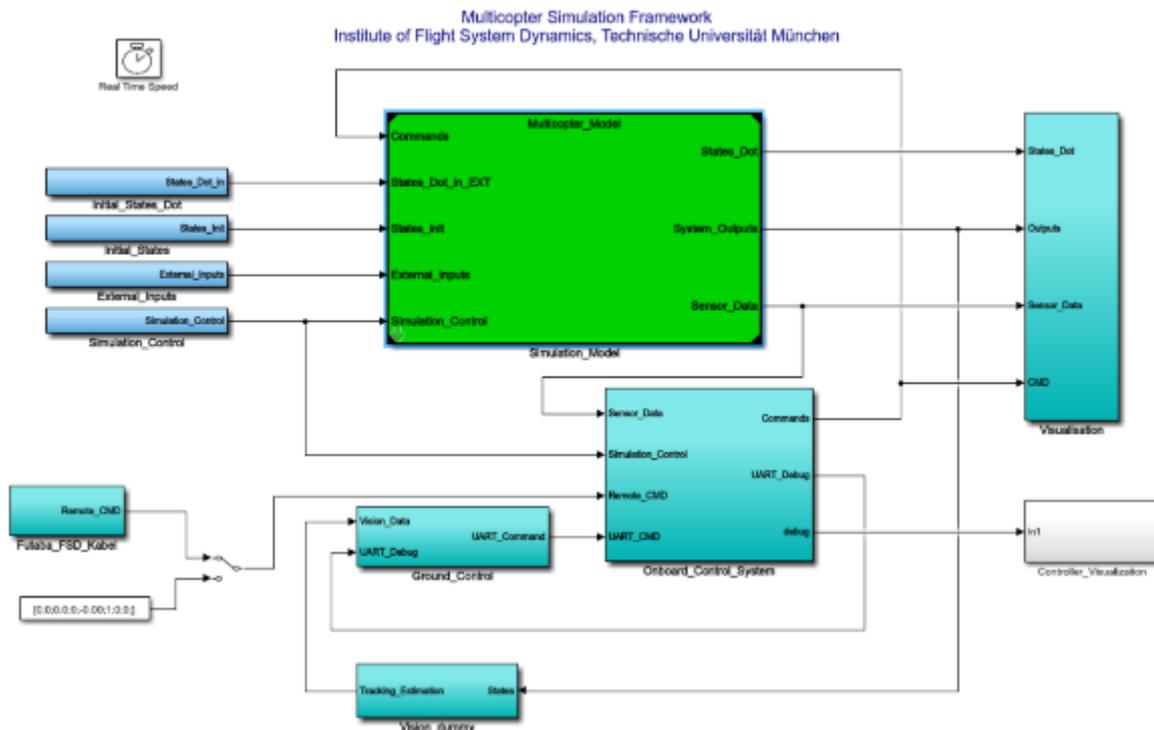


Abbildung 6.6.11: Aufbau des Simulations-Frameworks

Das Simulations-Framework besteht aus 5 Hauptkomponenten:

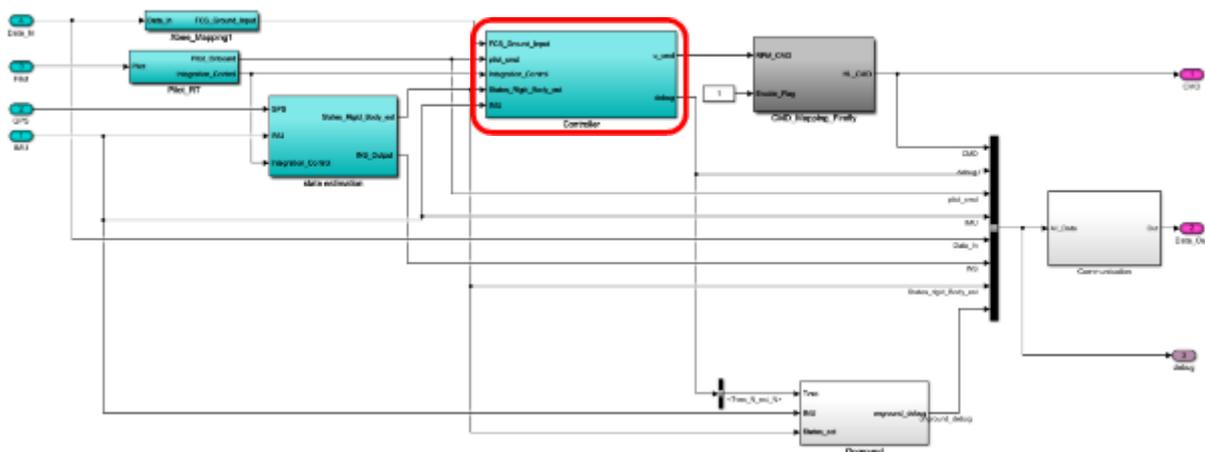
- Simulation Model
- Onboard-System
- Ground-Control
- Simulationssteuerung
- Visualisierung

Das Simulations-Modell wurde im Zug des VaMEx Forschungsprojektes entwickelt und um eine erweiterte Propeller-Aerodynamik (siehe Kapitel 6.6.2.1) erweitert. Die Komponente Simulationssteuerung dient zur allgemeinen Steuerung der Simulation und enthält unter anderem Reset und Run-Flags um die Simulation auf beliebige Startwerte zurückzusetzen zu können bzw. jederzeit anhalten zu können. Zusätzlich können darin die Initialwerte vorgegeben werden und externe Eingänge wie etwa Windgeschwindigkeiten eingestellt werden.

Die Ground-Control spiegelt die Bodenstation, welche bei Flugtests verwendet wird, wieder und dient zur Steuerung von Regler-Parametern. Zusätzlich können über diese Funktionen des Multikopters aktiviert bzw. deaktiviert werden. Als wichtigstes Beispiel ist hier das An- und Abschalten einzelner Propeller zu nennen.

Die Visualisierung dient zur Darstellung verschiedener Signale des Simulationsmodells und des Reglers. Dadurch kann in der Simulationsumgebung die Performance des Reglers evaluiert und verbessert werden.

Der Flugregelalgorithmus ist im Onboard-System implementiert. Durch den Modellbasierten Entwicklungsprozess kann aus diesem automatisiert Code erzeugt werden, welcher auf die den Multikopter portiert werden kann. Dadurch ist immer sicher gestellt, dass der in der Simulation getestete Flugregler mit dem auf der Hardware übereinstimmt. In Abbildung 6.6.12 ist der implementierte Flugregler dargestellt. Der in Kapitel 6.5.5 hergeleitete Algorithmus ist dabei im rot eingekreisten Block enthalten.



**Abbildung 6.6.12: Überblick über den implementierten Flugregler**

Die Implementierung des kaskadierten Reglers, bestehend aus Positions- und Lageschleife ist in Abbildung 6.6.13 dargestellt.

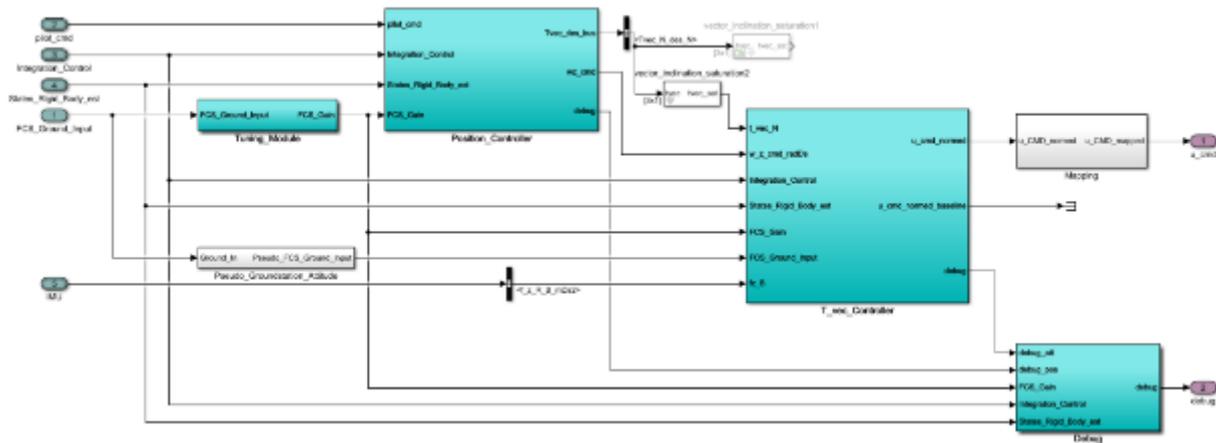


Abbildung 6.6.13: Implementierung des kaskadierten Flugregelungsalgorithmus

## 6.7 Untersuchung: Emulierte Flugvehikel-Dynamik

Dieses Kapitel beschäftigt sich mit der Frage, ob die Dynamik eines Flugvehikels auf einem anderen Fluggerät mit ähnlicher Architektur nachgebildet werden kann. Dies ist besonders für das VaMEx-CoSMiC Projekt von Bedeutung. Die Entwicklung von Flugreglern basiert in großen Teilen modellbasiert, es ist jedoch trotzdem unerlässlich den Flugregelungsalgorithmus in Flugversuchen zu testen. Dies ist jedoch für ein Fluggerät mit dem Einsatzgebiet Mars nicht möglich. Daher wurde die Frage untersucht, ob die Dynamik eines Multikopters auf einem anderen Multikopter nachgebildet werden kann. Zusätzlich wird ein möglicher Ansatz basierend auf Nichtlinearer Dynamischer Inversion (NDI) präsentiert.

Da für die Untersuchung die Modelle des Marsvehikels noch nicht zur Verfügung standen wurden zwei vorhandene Modelle verwendet. Die nachzubildende Dynamik basiert auf einem Quadrocopter (Guest). Diese soll auf einem Hexacopter (Host) nachgebildet werden.

Bei einem Multikoptersystem sind die rotatorische und translatorische Dynamik stark miteinander gekoppelt. Um in eine gewünschte Richtung zu beschleunigen und Geschwindigkeit aufzubauen, muss du erst die Lage des Multikopters geändert werden, um eine Kraft in die gewünschte Richtung zu erzeugen. Dies hat den Hintergrund, da die einzige beeinflussbare Kraftkomponente der Schub des Systems ist, welcher bei Standardkonfigurationen in die negative Richtung der z-Achse des Körperfesten Koordinatensystems zeigt.

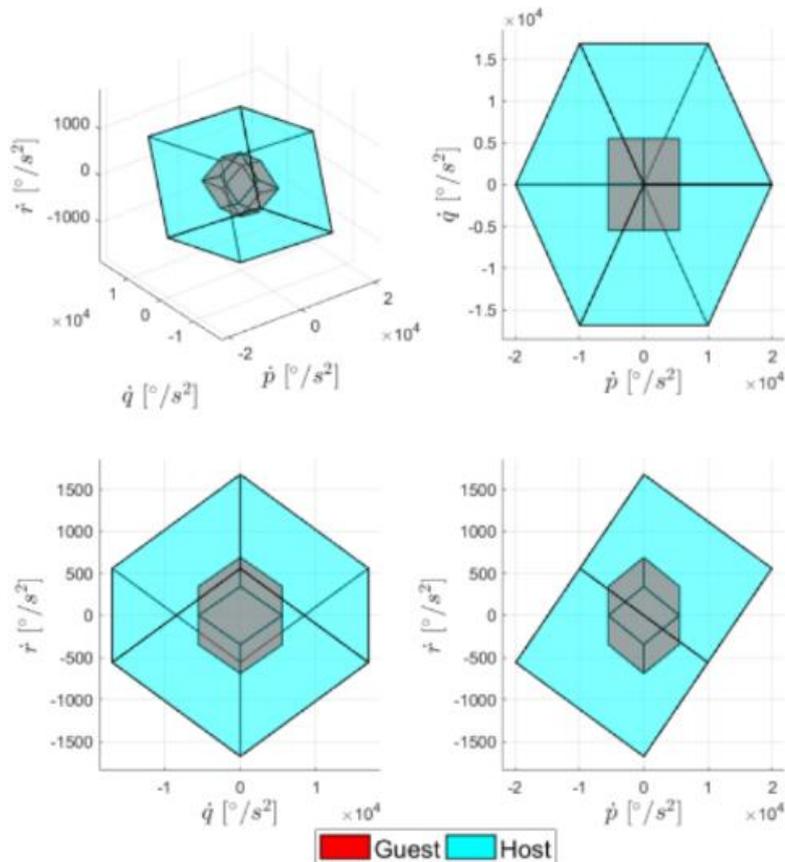
Neben der Schubkraft können noch die Momente um die Achsen des Körperfesten Systems beeinflusst werden. Durch diese Besonderheit muss die Emulation in zwei getrennte Konzepte aufgeteilt werden:

- Emulation der rotatorischen Dynamik
- Emulation der translatorischen Dynamik

### 6.7.1 Vergleich der Guest und Host Dynamiken

Bevor mit der Herleitung des Regelungsansatzes zur Emulation begonnen werden kann, muss untersucht werden, ob die Dynamik des Guest auf dem des Hosts abgebildet werden kann. Hier soll besonders auf die rotatorische Dynamik eingegangen werden, da dies die größte Einschränkung des Systems ist.

Für diese Untersuchung werden die maximalen Drehbeschleunigungen des Systems um die jeweiligen körperfesten Achsen bestimmt. Für die folgenden Darstellungen wurden die Toolbox von verwendet.

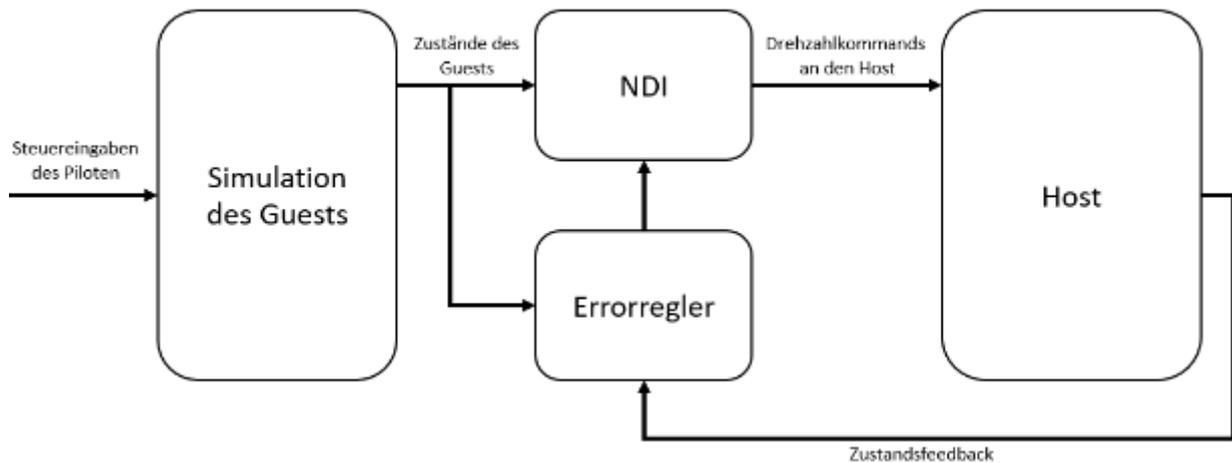


**Abbildung 6.7.1: Vergleich der Host und Guest Drehbeschleunigungen**

In Abbildung 6.7.1 sind für ein Beispiel die maximal möglichen Drehbeschleunigungen von Host und Guest dargestellt. Es ist zu erkennen, dass der Guest innerhalb des aufgespannten Bereichs liegt. Dadurch kann rückgeschlossen werden, dass die Emulation der rotatorischen Dynamik mit dem Host möglich ist. Würde der Guest außerhalb des Host-Bereichs liegen, müsste entschieden werden, ob die reduzierte Dynamik des Host für die gewünschten Fluguntersuchungen noch ausreichend ist.

### 6.7.2 Implementierung des Emulationskonzepts

Für die Implementierung des Emulationskonzepts wurden die Regelungsmethode NDI verwendet. Diese bietet sich an, da für ein optimales Relgerdesign, die Dynamik des Host rausgekürzt und die Dynamik des Guest vollständig auf dem Host abgebildet wird.



**Abbildung 6.7.2: Struktur des implementierten Reglers zur Dynamik-Emulation**

In Abbildung 6.7.2: ist die generelle Struktur des zur Dynamik-Emulation implementierten Reglers abgebildet. Die Hauptkomponenten sind dabei:

- Simulation des Guests
- NDI
- Errorregler

Die Simulation des Guest verwendet als Eingang die Steuerkommandos eines Piloten oder eines Flug-Management-System. Aus diesen Steuerkommandos werden die Zustände des Guests abgeleitet.

Der NDI-Algorithmus verwendet nun diese Zustände für die Berechnung der Drehzahlkommandos des Hosts, um die Dynamik des Guest möglichst exakt abzubilden. Da in der Realität Unsicherheiten von Parametern für den NDI Algorithmus vorhanden sind, wird ein Fehlerregler (Errorregler) hinzugefügt. Dieser sorgt dafür, dass die Zustände des Guest von Host auch beim Vorhandensein von Unsicherheiten getrackt werden.

Für die mathematische Implementierung des NDI Algorithmus sei auf und .

### 6.7.3 Simulationsergebnisse

Im Folgenden sollen einige Beispielmission zur Validierung der Performance der Emulationsmethode präsentiert werden.

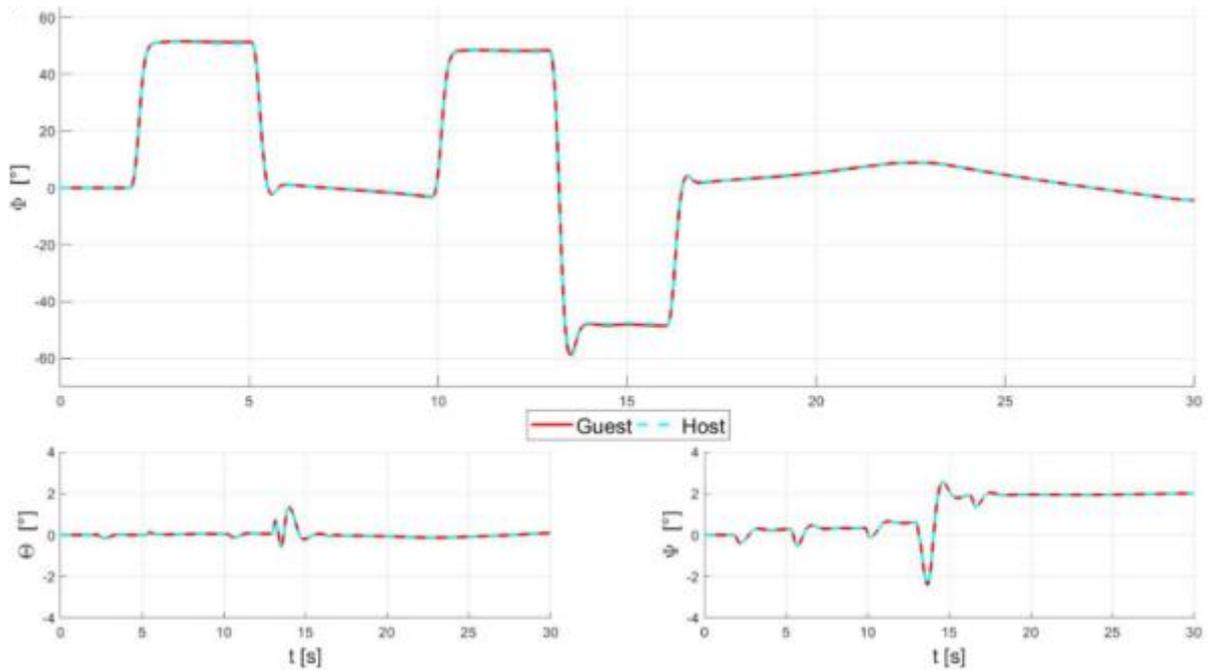


Abbildung 6.7.3: Tracking-Performance des Emulationsalgorithmus für Rotatorische Dynamik (Lage)

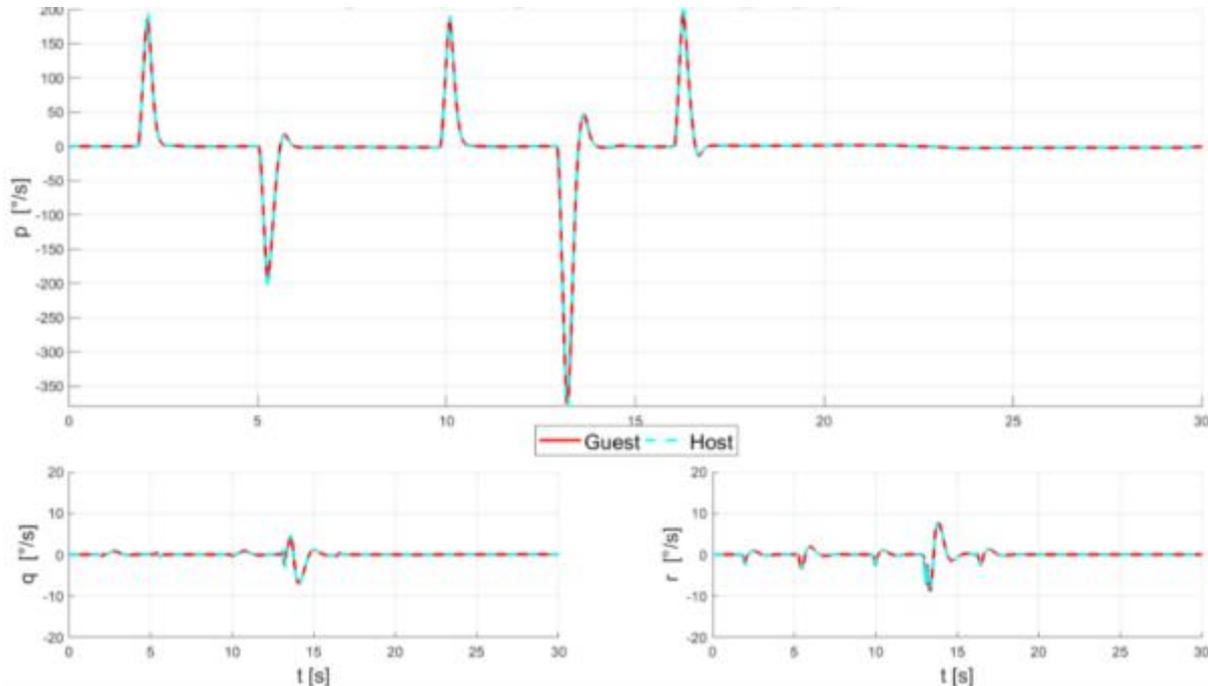
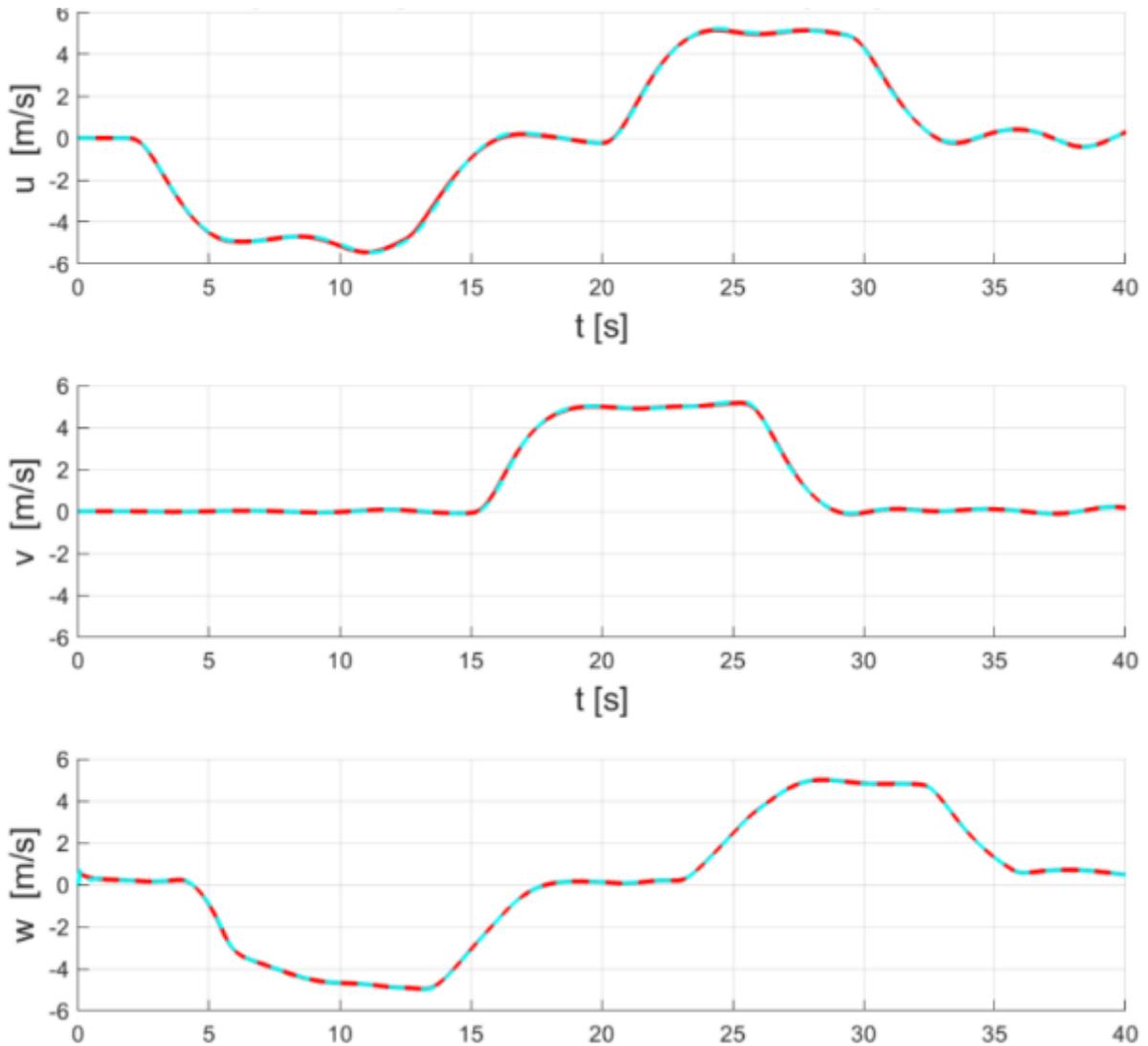


Abbildung 6.7.4: Tracking-Performance des Emulationsalgorithmus für Rotatorische Dynamik (Drehraten)

In Abbildung 6.7.3 und Abbildung 6.7.4 ist die Trackingperformance in Bezug auf rotatorische Emulation des Host für eine vorgegebene Guest-Dynamik gegeben. Es ist gut zu erkennen, dass sowohl die Drehraten als auch die Lage gut getrackt werden.



**Abbildung 6.7.5 Tracking-Performance des Emulationsalgorithmus für Translatorische Dynamik (Geschwindigkeiten)**

In Abbildung 6.7.5 und Abbildung 6.7.6 sind die Simulationsergebnisse für die translatorische Dynamikemulation dargestellt. Wie in beiden Fällen gut zu erkennen ist, wird sowohl die Geschwindigkeit als auch die Position zur Zufriedenheit getrackt.

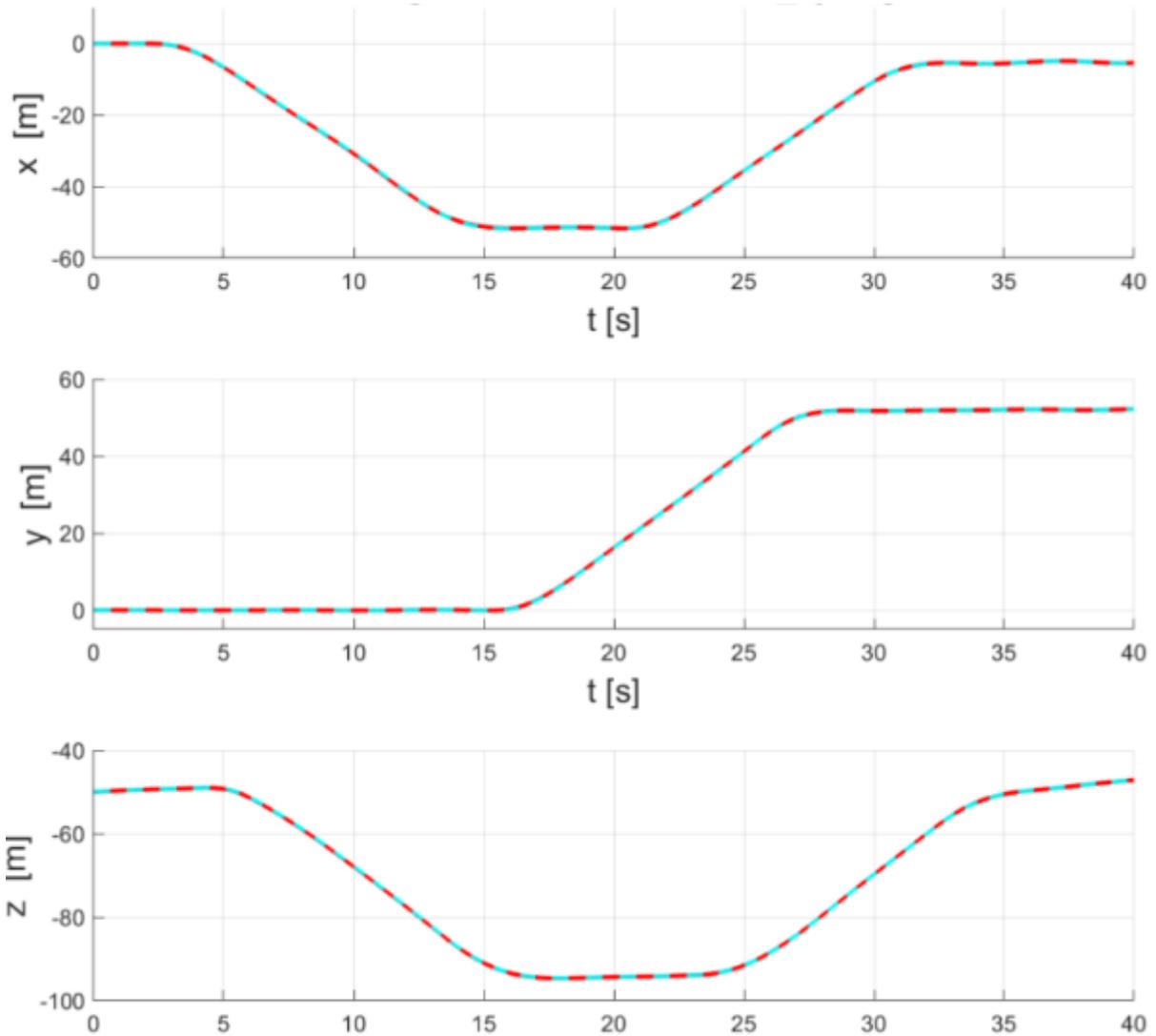


Abbildung 6.7.6: Tracking-Performance des Emulationsalgorithmus für Translatorische Dynamik (Position)

## 6.8 Automatische Schwarmnavigation

### 6.8.1 Selbstorganisierende Exploration

Unter Schwarmexploration verstehen wir datengetriebene und modellbasierte Algorithmen, welche Zielpunkte für die einzelne Agenten des Schwarms generieren, an denen zukünftige Messungen am informativsten sind. Dazu werden bereits gesammelte Daten von z.B. einem wissenschaftlichen Instrument automatisch ausgewertet, sowie Modellannahmen über den zu beobachtenden physikalischen Prozess berücksichtigt.

Dabei werden Zielfunktionen optimiert, die auf den Messdaten aller Schwarmelemente basieren. Erst die Kooperation im Netzwerk erlaubt diese Ansätze, daher ist die Kommunikation im Schwarm essentiell, sowie die Lokalisierung um die Messdaten den Orten im Erkundungsgebiet zuordnen zu können.

Abbildung 6.8.1 zeigt allgemein das Prinzip der Schwarmexploration. Ziel ist es, anhand von Messungen im Schwarm, die Parameter eines Prozessmodells bestmöglich zu schätzen. Als Prozessmodelle können sowohl bekannte Modelle, wie z.B. Differentialgleichungen und Transferfunktionen, als auch "Black-Box"-Modelle, wie z.B. Kernel-Methoden eingesetzt werden.

Die Schätzung der Parameter erfolgt im Netzwerk mit Hilfe verteilter Algorithmen, z.B. Message-Passing. Somit kann auch die Rechenlast im Netzwerk zwischen den Agenten verteilt werden und Synergien in der Berechnung ausgenutzt werden.

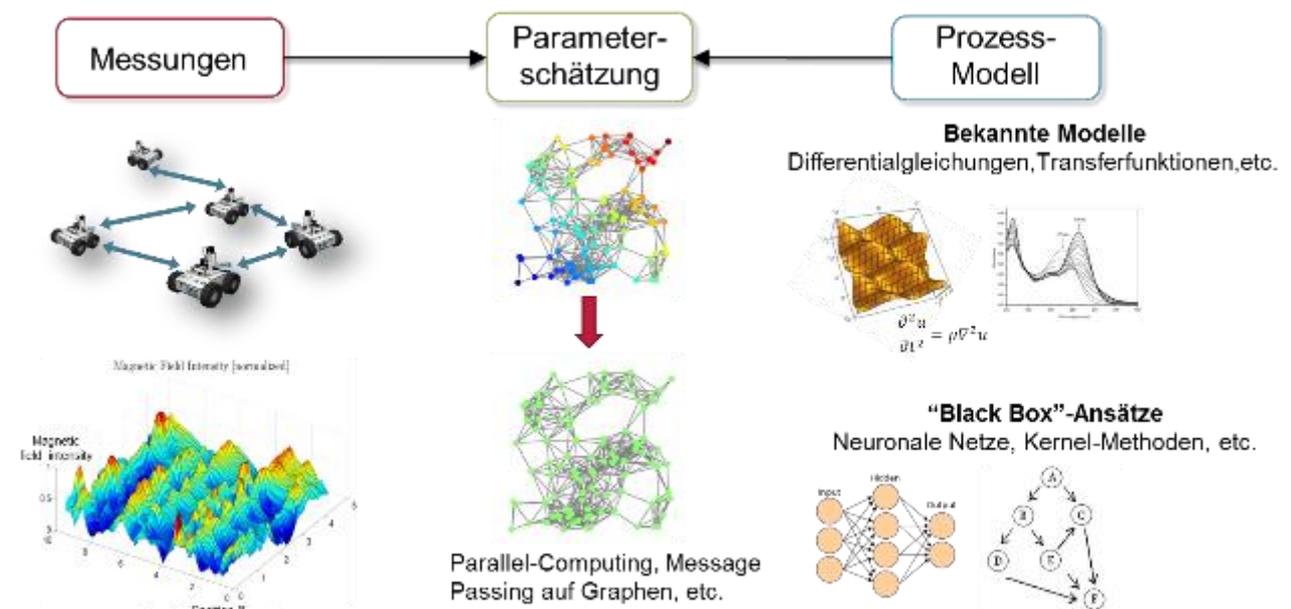


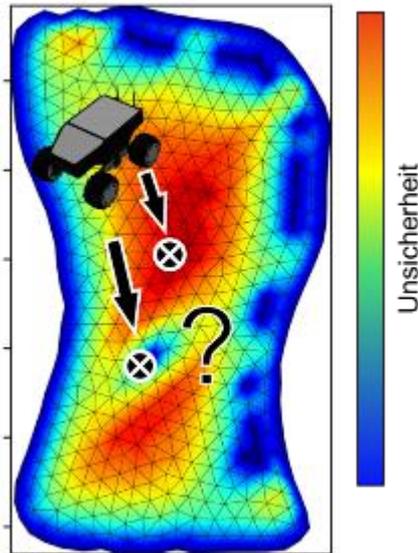
Abbildung 6.8.1: Datenprozessierung im Netzwerk für die Schwarmexploration.

In diesem Vorhaben wurde der Fokus auf die Gas-Exploration gesetzt, welche sowohl in der Raumfahrtforschung, als auch in terrestrischen Anwendungen wie der Geologie/Atmosphärenforschung und dem Katastrophenschutz Verwendung finden kann. Der zu beobachtende Prozess ist folglich die dynamische Gasausbreitung. Die Aufgabe des Schwarms ist die Exploration und Kartierung der Gaskonzentration und das Auffinden von Gasquellen. D.h. die Agenten sollen alle Gasquellen finden, wobei weder die Anzahl, noch die Positionen, noch die Stärke der Gasquellen bekannt sind. Die Agenten sind aber in der Lage an ihrer jeweiligen Position punktwise die Gaskonzentration zu messen.

Die Problemstellung liegt nun darin, eine Explorationsstrategie für den Schwarm zu finden. Die entwickelte Lösung basiert auf der Ausnutzung von physikalischen Modellen der

Gasausbreitung über partielle Differentialgleichungen (Diffusionsgleichungen). Die Diffusionsgleichung erlaubt drei wesentliche Punkte für die Gasexploration:

1. Inferenz, da die Quelle nicht direkt messbar ist und nur indirekt über Gaskonzentrationsmessungen bestimmt werden kann.
2. Die Diffusionsgleichung ist a-priori Information, welche die Exploration deutlich beschleunigt.
3. Unsicherheitsquantifizierung, welche die Basis für informationsgetriebene Exploration ist.



**Abbildung 6.8.2: Unsicherheitskarte**

Die Explorationsstrategie selbst legt fest, an welchen Stellen im Erkundungsgebiet die Gaskonzentration gemessen werden soll. Dabei wird eine informationsgetriebene (Entropie) Exploration eingesetzt. Regionen im Erkundungsgebiet mit hoher Unsicherheit in den Modellparametern werden für neue Messungen bevorzugt. Diese Messungen bedeuten den höchsten Informationsgewinn bezüglich der Gesamtschätzung der Modellparameter.

Für die Explorationsstrategie wird also eine räumliche Beschreibung der Unsicherheiten verwendet um die Regionen mit der höchsten Unsicherheit (dem höchsten Informationsgewinn) als nächste Wegpunkte für die Agenten des Schwarms auszuwählen. Abbildung 6.8.2 zeigt beispielhaft eine Unsicherheitskarte. Der Agent würde in diesem Beispiel den Wegpunkt in der roten Region wählen.

Zusätzlich zur Explorationsstrategie wird ein Koordinierungsverfahren benötigt, das die berechneten Wegpunkte den einzelnen Agenten zuordnet und während der Exploration Kollisionen der Agenten vermeidet. Hierzu müssen die Agenten im Schwarm zusätzlich ihre aktuellen Positionsinformationen austauschen.

Das Verfahren zur Gasexploration wurde als Hardware-in-the-loop (HIL) Simulation umgesetzt. Dabei werden Gasquellen und Luftströmungen bedingt durch Wind, als auch der Gassensor numerisch simuliert. In dieser Simulation werden die sechs realen Rover von DLR-KN mit der lokalen Kommunikation, der Lokalisierung, und der realen Hinderniskarte eingebunden. D.h. die Rovers bewegen sich real durch das Explorationsgebiet, der zu erkundende Prozess ist aber virtuell simuliert. Für die Implementierung des Explorationsalgorithmus wurde das Robot Operating System (ROS) verwendet. Zum Austausch der Daten wie z.B. die Ortsinformation der Rover, wurde ein ROS Multimaster-System verwendet, um ein verteiltes System ohne „Single-Point-of-Failure“ zu realisieren. Ergänzend wurden Schnittstellen zur Bodenstation der TU-Braunschweig zum Austausch von

Zustandsinformationen und der Hinderniskarte realisiert.

## 6.8.2 Exploration zur Kartenerstellung

Auf Grundlage einer Karte soll der Teilnehmer des heterogenen Schwarms gezielt nach ein vordefiniertes Gebiet vermessen und so beispielsweise die magnetische Flußdichte, die Feuchtigkeit im Boden oder aber die Konzentration und Verteilung von anderen Substanzen kartieren. Wie bereits erwähnt muss dazu zunächst eine entsprechende Karte des zu vermessenden Gebietes vorliegen. Die durch Satelliten oder Sonden verfügbaren Karten des Valles Marineris sind dazu ungenügend und bilden die Oberfläche nicht im benötigten Detailgrad ab. Ein Teil des Schwarms verfügt jedoch über Sensorik, wie LiDAR, um die für eine weitere Erkundung benötigte Karte mit dem gewünschten Detaillierungsgrad zu generieren.

Der hier vorgestellte Ansatz zur Erzeugung einer dreidimensionalen Karte basiert auf dem von Yamauchi [4] entwickelten Ansatz einer zweidimensionalen, grenzenbasierten Exploration. Dabei wird zunächst ein zu explorierendes, zunächst unbekanntes Areal als leere Karte vordefiniert, welches schrittweise durch die Sensoren des explorierenden Systems mit Umgebungsinformationen ergänzt wird. Dabei werden in den Randgebieten vom bekannten zum unbekanntem Areal Grenzen definiert, welche als Explorationsziele dem robotischen System als Zielpunkte zur Verfügung stehen. Dabei wählt das robotische System jenes Explorationsziel aus der Liste der möglichen Ziele als nächsten Wegpunkt an, welches mit dem kürzesten Pfad korrespondiert.

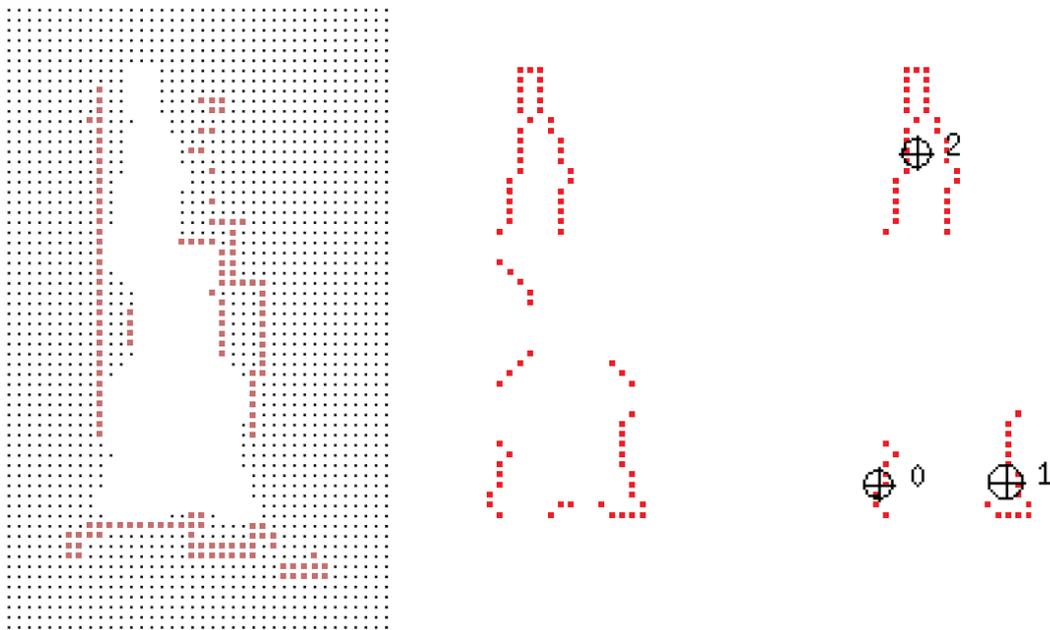
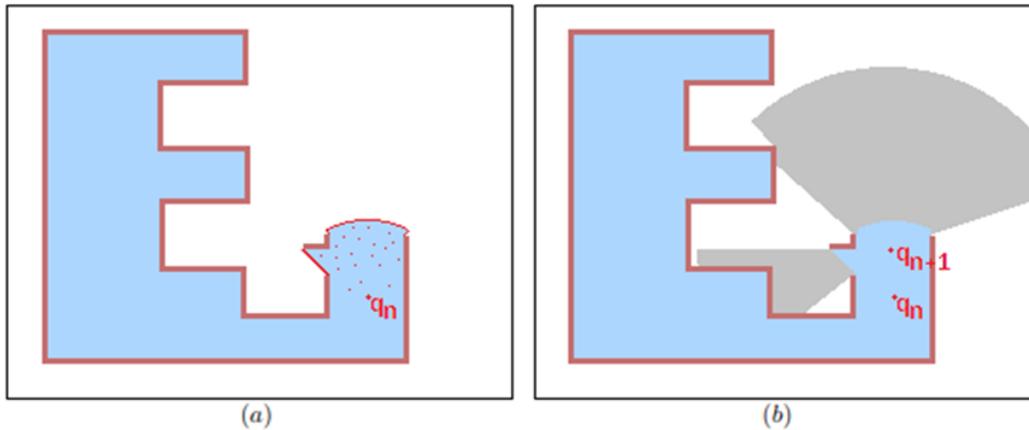


Abbildung 6.8.3: zweidimensionale, grenzenbasierte Exploration nach Yamauchi [4]

Die Karte wird durch die Sensormessungen aktualisiert und somit verschieben sich auch die Grenzen und Zielpunkte weiter. Dies wird solange fortgesetzt bis alle erreichbaren Zielpunkte erkundet wurden oder die gesamte, anfänglich leere Karte vollständig mit Informationen gefüllt ist. Nicht erreichbare Zielpunkte sind dabei solche, zu denen kein Pfad geplant werden konnte. In solch einem Fall kommt es vor, dass für Bereiche der Karte am Ende der Erkundung weiterhin keine Informationen vorliegen.

Dieser Ansatz wurde im Rahmen des Projektes für den dreidimensionalen Fall zur Erzeugung einer OctoMap umgesetzt. Dazu wurde ausgehend von einem anfänglichen Kartenausschnitt eine Bounding Box definiert. Diese Bounding Box enthält neben den bekannten Bereichen nun auch unbekanntes Areal. Auf Basis dieser Daten können Grenzen definiert werden. Neben der Identifikation der Grenzen wurde das Sensorverhalten, dessen Erfassungsbereich,

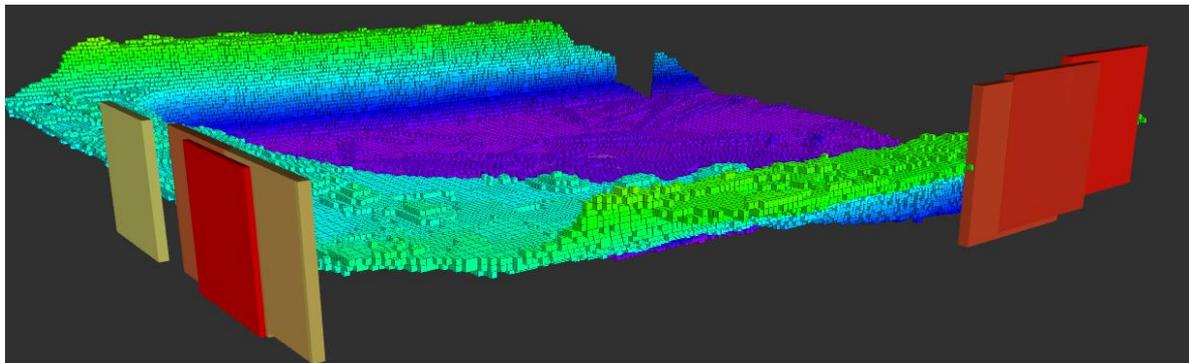
berücksichtigt. Somit inkoooperiert der Algorithmus auch Ansätze des „Next-Best-View“ Algorithmus, welcher Ziele favorisiert, für welche der Informationszugewinn durch einen Sensor am höchsten vermutet wird. Ihm liegt ein Modell des Sensors zugrunde mit Hilfe dessen der Zugewinn an einem beliebigen Punkt ermittelt werden kann. Der Ansatz des „Next Best View“ ist in Abbildung 6.8.4 für den zweidimensionalen Fall dargestellt.



**Abbildung 6.8.4: (a) Karte mit möglichen Positionskandidaten (rot); (b) Zielpunkt  $q_{n-1}$  mit maximalen Informationszugewinn**

Für eine Anzahl von Positionen  $q_n$  wird die Position bestimmt, für welche der Zugewinn an neuen Informationen durch das hinterlegte Sensormodell und dem aktuellen Wissen aus der Karte am größten ist. Dabei ist wichtig zu beachten, dass der Punkt immer im bekannten Areal liegt.

Ferner wurde festgelegt, dass der anzufliegende Grenzbereich größer ist als das Vehikel, sodass dieses die Zielposition auch ohne Kollision des sensortragenden Systems mit anderen Strukturen auch einnehmen kann. Das Resultat dieser zur Kartierung entwickelten Explorationsstrategie ist in Abbildung 6.8.5 dargestellt.



**Abbildung 6.8.5: vielversprechendste Explorationsziele bzgl. des Informationszugewinns in der Nähe des Flugsystems**

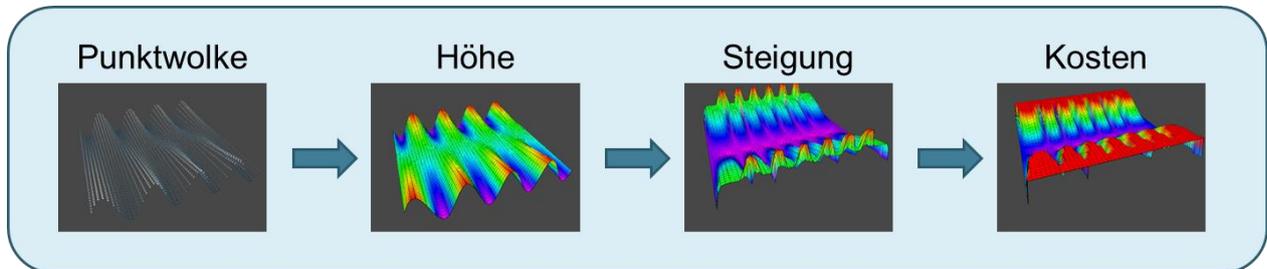
In der Abbildung sind nur die besten 7 Lösungen bezüglich des Informationszugewinns und der Nähe zur Position des explorierenden Flugsystems dargestellt. Eine Bewertung auf Basis der Pfadlänge erfolgte im Rahmen des Projektes nicht. Die Versuche wurden mit Hilfe einer Simulationsumgebung, namentlich Gazebo, realisiert.

### 6.8.3 Taktische Nahfeldnavigation

#### 6.8.3.1 Umsetzung für Unmanned Ground Vehicle (UGV)

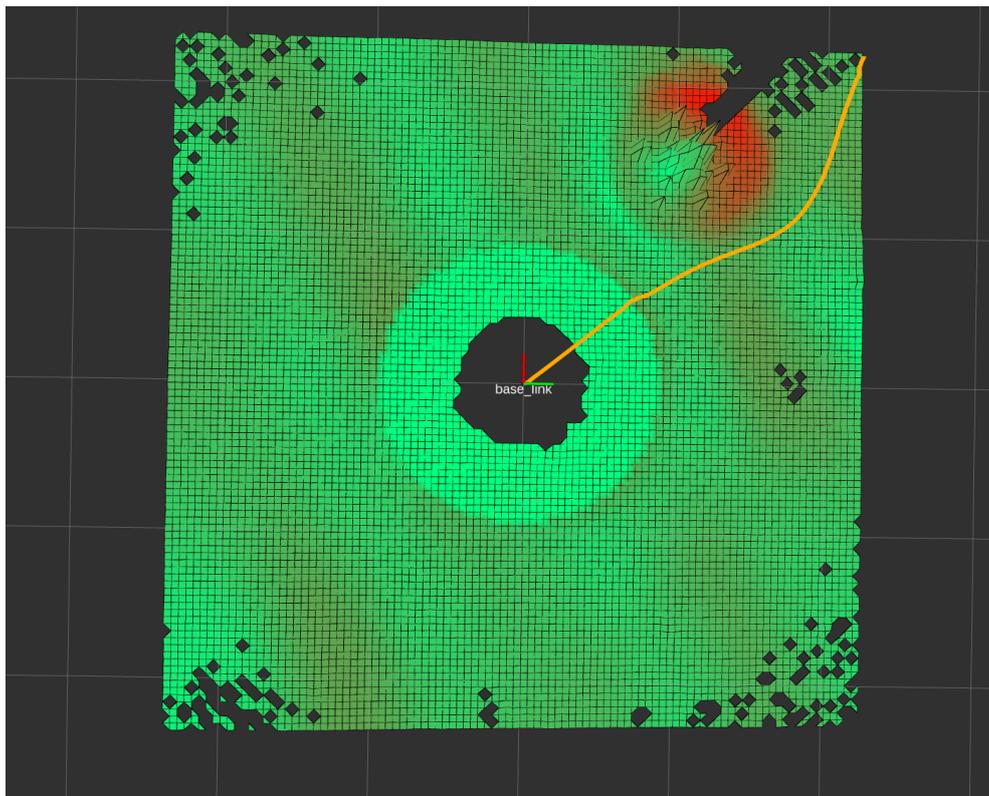
Um eine sichere Exploration in unbekanntem Gelände durchzuführen, darf sich ein Rover nur soweit Richtung nächstem Wegpunkt bewegen, wie die Hindernisfreiheit und Befahrbarkeit

gewährleistet werden kann. Die Datenbasis hierfür bilden LiDAR Punktwolken. Dabei muss die Punktdichte im Zentimeterbereich liegen, um auch Schlaglöcher relevanter Größe sicher zu erkennen. Prinzipbedingt durch die Position des LiDARs auf dem Rover sowie seiner konstante Winkelauflösung erhält man im Allgemeinen die erforderlichen Punktdichten im Umfeld von etwa fünf Metern um den Rover, welches dann im nächsten Schritt befahren werden kann. Die Punktwolke wird nach der Erfassung georeferenziert. Die erforderliche Positions- und Lageinformation wird von der parallel dazu ablaufenden Registrierung der Punktwolke im SLAM bezogen.



**Abbildung 6.8.6: Schrittweise Verarbeitung der Punktwolke hin zur Kosten-Karte**

Zur Berechnung einer Kostenkarte werden aus der Punktwolke Parameter wie Höhe, Steigung zellenweise ermittelt und in einer Gridmap (Gitterkarte) gespeichert. Zellen, für welche keine Messpunkte vorliegen können, je nach Strategie der Wegplanung, umfahren oder nach einer Annäherung des Rovers an die Stelle neu vermessen werden. Abbildung 6.8.7 zeigt die Umfahrung eines Hindernisses bis zum nächsten Messpunkt.



**Abbildung 6.8.7: Umfahrung eines Hindernisses durch Pfadplanung auf Basis der Kosten-Karte**

### 6.8.3.2 Umsetzung für Unmanned Aerial System (UAS)

Die Umgebungsinterpretation für ein fliegendes System entspricht nur geringfügig der eines bodengebundenen Systems. Einschränkungen, welchen das bodengebundene System unterliegt, berühren unter Umständen ein fliegendes Gerät nur unmerklich. Insbesondere im

Fall eines Multirotorsystems ist die Steigung nur von untergeordneter Relevanz, da ein Multirotorsystem im Bedarfsfall ein reines Vertikalmanöver, d.h. die verfügbare Antriebsleistung wird vollständig in eine Vertikalgeschwindigkeit umgesetzt, durchführen kann. Im Rahmen des Projektes wurden verschiedene Ansätze zur Umsetzung von Antikollisionsroutinen basierend auf den bereits vorhandenen LiDAR untersucht. Beide hier vorgestellten Ansätze nutzen als Grundlage eine OctoMap.

#### **6.8.3.2.1 Ansatz für dynamische Umgebung**

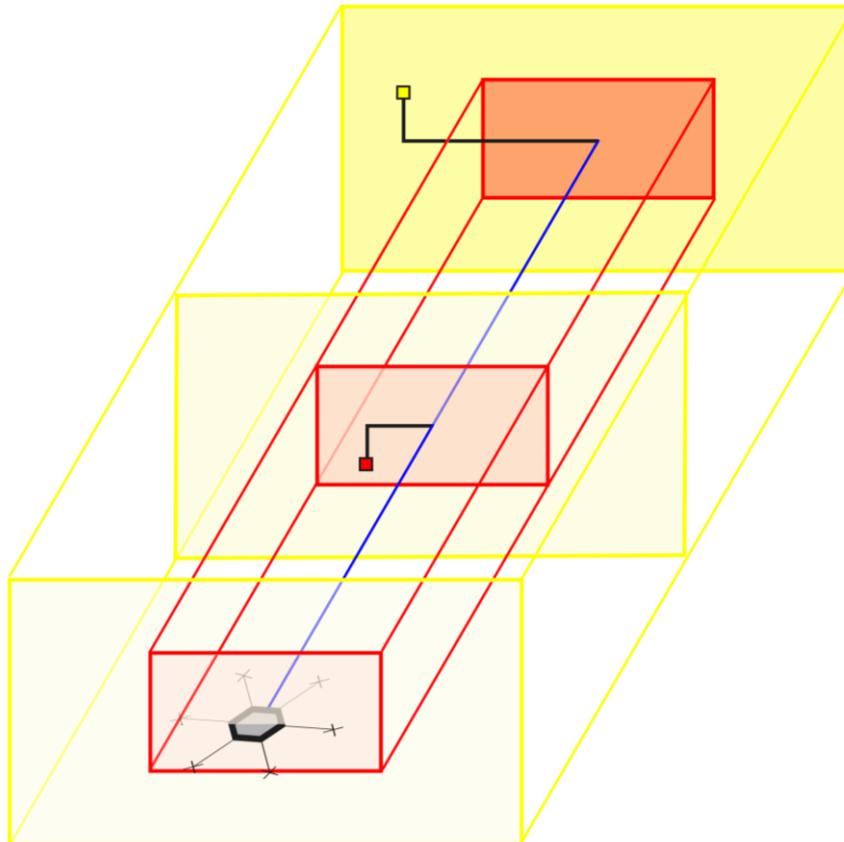
Auf dem Mars kann zunächst von einer statischen Umgebung ausgegangen werden. Dennoch muss auch die Dynamik des Schwarms, bestehend aus heterogenen Schwarmteilnehmern, berücksichtigt werden, die gleichzeitig agieren und auch miteinander interagieren sollen. Daher ist die Annahme einer statischen Umgebung unzureichend um eine sichere Navigation, ohne Kollision der Schwarmteilnehmer miteinander, zu gewährleisten. Daher wurde in diesem Ansatz untersucht ob die Daten eines LiDAR herangezogen werden können um rechtzeitig stumme Schwarmteilnehmer zu erkennen und eine Kollision zu vermeiden. Als stumme Schwarmteilnehmer werden hier Schwarmteilnehmer betrachtet, über die keine aktuellen Informationen über deren Position und Geschwindigkeit vorliegen. Im Idealfall ist jedem Schwarmteilnehmer die Position aller anderen Schwarmteilnehmer zu jedem Zeitpunkt bekannt. Jedoch kann durch einen vorübergehenden Ausfall der Kommunikationsstrecke keine aktuelle Information über den Zustand aller Schwarmteilnehmer vorliegen. Da die Schwarmteilnehmer dennoch ihre Aufgabe bis zu einem gewissen Maße ohne eine bestehende Kommunikationsverbindung weiter fortführen sollen, müssen auch für diesen Fall Mechanismen greifen, die eine Kollision der Schwarmteilnehmer untereinander verhindern. Dabei ist dem System jedoch seine eigene Position, Orientierung und Geschwindigkeit bekannt, sei dies nun durch ein Referenzsystem oder aber durch die Lösung des INS. Für die Realisierung des Ansatzes wurde das unbemannte Fluggerät hardwareseitig nicht modifiziert.

#### **Umsetzung**

Zur Umsetzung einer Kollisionsvermeidung in einer dynamischen Umgebung wurden zunächst zwei Strategien in Betracht gezogen. Zum einem die Speicherung der gesammelten Daten zur Erzeugung einer OctoMap mit einem anschließenden Raytracing zur Suche nach Hindernissen im Flugpfad und zum anderen eine Analyse der Scans in Bewegungsrichtung inklusive einer Unterteilung in Sektoren unterschiedlichen Gefahrenpotentials ein. Neben den beiden bereits erwähnten Ansätzen wurde weiterhin untersucht ob ein Vektor-Field-Histogramm unter Verwendung der als OctoMap vorliegenden Kartendaten zur Hindernisvermeidung herangezogen werden kann.

### Sektoreneinteilung

Die Einteilung der Sektoren und deren Verlauf ist in der nachfolgenden Abbildung 6.8.8 dargestellt.



**Abbildung 6.8.8: Prinzip der Sektoreneinteilung**

Die Sektoren sind dabei um den Flugpfad, in der Abbildung in blau dargestellt, angeordnet. Der innere rot umrandete Sektor besitzt in seiner Ausdehnung in etwa die Abmessungen des unbemannten Flugsystems. Der äußere Sektor umschließt den inneren vollständig und kann in seiner Ausdehnung ein Vielfaches des inneren Sektors betragen. Dieser dient vornehmlich als zusätzlicher Sicherheitsbereich und soll dazu dienen unvorhersehbare Abweichung vom Pfad, beispielsweise durch Navigationsfehler, zu berücksichtigen um auch für diese Fälle einen kollisionsfreien Flug sicherstellen zu können. Für jede Messung des Laserscanners erfolgt zunächst die Transformation in das körperfeste Koordinatensystem des unbemannten Flugsystems. Die Messpunkte werden dann über die Berechnung des Abstandes vom Flugpfad dem inneren oder äußeren Sektor zugeordnet. Messwerte, die außerhalb der Sektoren liegen, werden sofort verworfen. Dabei werden grundsätzlich nur die Entfernungsmessungen einer Scanzeile ausgewertet, die in Flugrichtung liegen. Anschließend wird für jeden Sektor jeweils der Wert mit dem kleinsten Abstand zum UAS zwischengespeichert, in das globale Koordinatensystem transformiert und abschließend mit dem aktuell gespeicherten Sektorminimum, durch Gegenüberstellung der aktuellen Abstände zum UAS, verglichen. Sollte der Abstand des aus dem aktuellen Scan errechneten Punktes für das Sektorminimum kleiner sein als der für den gespeicherten Punkt, wird letzterer durch diesen ersetzt.

Auf Basis der beiden gespeicherten Werte für die beiden Sektoren kann eine Bewertung der Situation erfolgen und Maßnahmen ergriffen werden.

Der Ansatz bietet den Vorteil der einfachen Umsetzung und eines geringeren Bedarfes an Speicher, da lediglich zwei Werte, je einer pro Sektor, hinterlegt werden. Ein Nachteil dieses

Ansatzes ist, dass alle anderen erfassten Daten eines Scans verworfen werden und bei einer Neuplanung eines Pfades oder Kollisionsanalyse bei einer abrupten Änderung der Flugrichtung nicht zur Verfügung stehen.

### OctoMap Ray-Casting

Im Gegensatz zum Ansatz der Einteilung in Sektoren werden beim hier verfolgten Ansatz keine Daten eines Scans mit dem LiDAR verworfen. Alle seine Informationen werden verarbeitet und fließen in eine Umgebungsdarstellung in Form einer OctoMap ein.

Da in dieser Karte alle zuvor gesammelten Informationen abgelegt werden, können diese auch in Ausnahmesituationen, beispielsweise einer unerwarteten und ungeplanten Kursänderung, herangezogen werden, um eine Analyse des momentanen Flugpfades zur Vermeidung von Kollisionen durchzuführen bis aktuellere Daten durch den LiDAR vorliegen.

Weiterhin ermöglicht es dieser Ansatz, bei hinreichend genauer Information über die Orientierung und Position des LiDAR-tragenden Systems eine erste, grobe und schnelle Kartierung eines Areals ohne die Zuhilfenahme eines SLAMs voranzutreiben. In der nachfolgenden Abbildung 6.8.9 ist eine, im Zuge eines solchen Fluges auf einem Modellflugplatz, erzeugte OctoMap in unterschiedlichen Auflösungen dargestellt. Der zugrundeliegende Flugpfad ist in Abbildung 6.8.10 dargestellt.

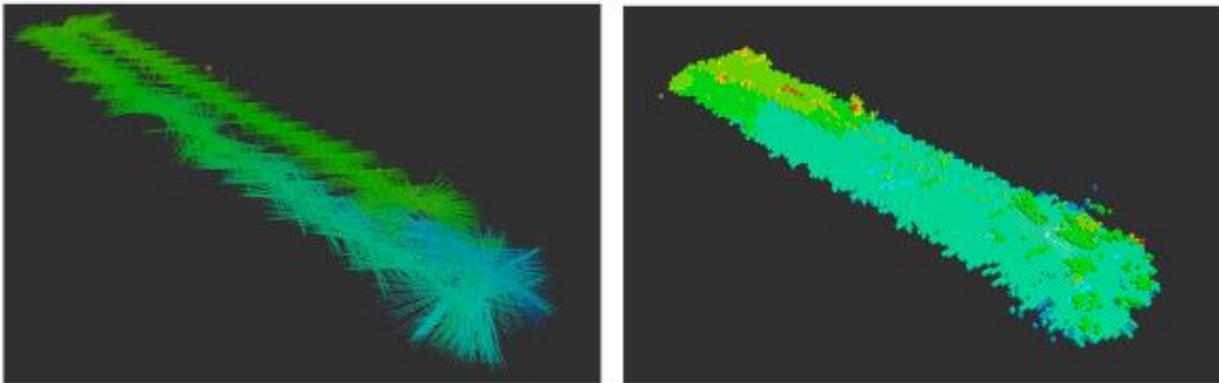


Abbildung 6.8.9: Unterschiedliche Auflösungen der OctoMap aus den selben Daten eines Flugversuches; Auslösungen: links: 0.1 [m], rechts: 1.0 [m]



Abbildung 6.8.10: Flugpfad inklusive Wegpunkte als Basis für die Kartierung

Das eigentliche Verfahren zur Detektion von Hindernissen wird anhand der so schrittweise erweiterten Karte, welche in Form einer OctoMap vorliegt, durchgeführt. Bei der Verwendung

des LiDAR müssen zur Hindernisvermeidung mehrere Randbedingungen berücksichtigt werden, da sich durch die Art des Sensor und der durch diesen angestrebten Winkelauflösung beispielsweise eine Abhängigkeit zur maximalen Fluggeschwindigkeit des UAS ergibt. Für den Fall des verwendeten Laserscanners Hokoyo UTM 30-LX sind einige Parameter unveränderliche Konstanten.

Dazu gehören vornehmlich

- die Scanrate des LiDAR mit 40 Hz, sowie
- die vertikale Winkelauflösung einer Scanzeile von 0.25°.

Einige Parameter, die für die Erkennung von Hindernissen und der Kollisionsvermeidung von Bedeutung sind, müssen jedoch definiert werden. Dazu zählen beispielsweise die minimale Größe des zu detektierenden Hindernisses  $b$ , die Detektionsreichweite  $d$ , in welcher das Objekt erkannt werden soll, und der Abtastbereich  $\zeta$ , welcher sich um die Flugrichtung des UAS aufspannt. Alle drei Parameter beeinflussen maßgeblich die horizontale Auflösung aber auch die maximale Fortbewegungsgeschwindigkeit des UAS.

Im Rahmen des Projektes wurden einige Parameter festgelegt. Dazu gehören die minimale Größe des zu detektierenden Objektes, welche auf 30 Zentimeter festgelegt wurde, um auch den kleinsten Teilnehmer des simulierten Schwarmverbundes zu berücksichtigen. Weiterhin wurde die minimale Größe des Abtastbereiches auf 45 Grad begrenzt und eine Überstreichung des gesamten Abtastbereiches mit einer Frequenz von 2 Hertz vorgesehen.

Darüber hinaus wurde eine maximale Geschwindigkeit der UAS von 2.5 Metern pro Sekunde angenommen. Somit ergibt sich für den Detektionsabstand nach gegebener Formel folgende Distanz.

$$d = 3 * v = 3 * 2.5 = 7.5m$$

In folgender Abbildung sind die Beziehungen zwischen den einzelnen Parametern dargestellt.

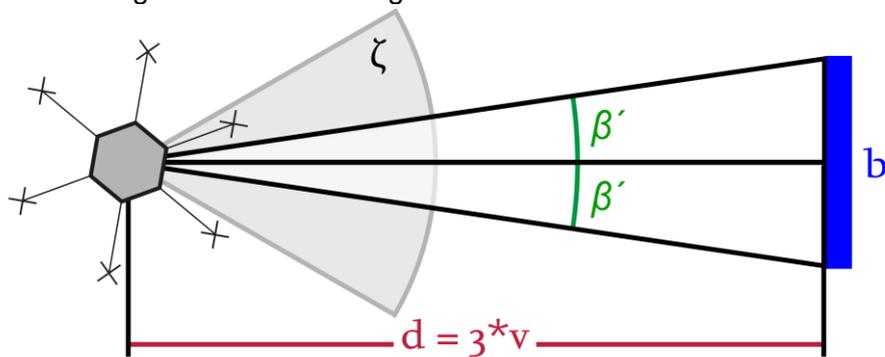


Abbildung 6.8.11: Darstellung der Beziehung der frei wählbaren Parameter

Die horizontale Winkelauflösung  $\beta$  berechnet sich wie folgt:

$$\beta = 2 * \beta' = \frac{2 * \arcsin(0.5 * b)}{3 * v} [rad]$$

Der Abtastbereich kann somit wie folgt berechnet werden:

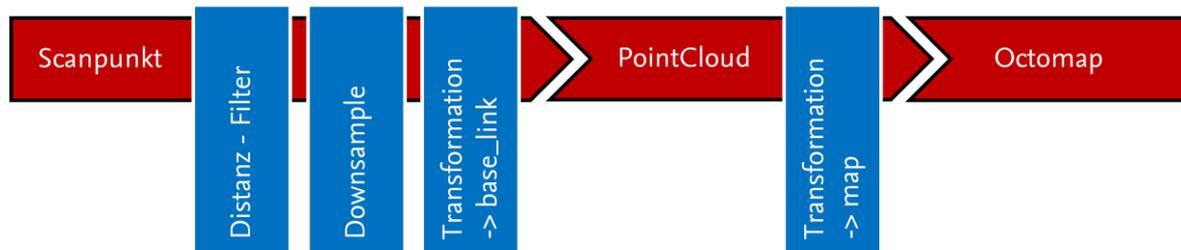
$$\zeta = \frac{2 * f_{LiDAR} * \beta'}{f_{Servo}} [rad]$$

Mit der Frequenz, mit welcher der Bereich durch den Servo abgetastet werden soll, kann nun die benötigte Drehgeschwindigkeit  $\omega_{Servo}$  bestimmt werden:

$$\omega_{Servo} = \zeta * f_{Servo} [rad/s]$$

Das Heading des Multirotorsystems, um welchen der Abtastbereich angelegt ist, wird dabei als bekannt angenommen.

Die aus den LiDAR Daten gewonnen Entfernungsmessungen durchlaufen, bevor sie zur Erzeugung einer OctoMap genutzt werden, eine Reihe von Prozessschritten um die Größe der Punktwolke, die Anzahl der enthaltenen Punkte, auf eine zweckmäßige Anzahl zu verringern. Der Ablauf der Datenprozessierung bis hin zur resultierenden OctoMap ist in der nachfolgenden Grafik dargestellt.



**Abbildung 6.8.12: Verarbeitungsschritte der LiDAR Daten**

Auf Basis der vorgegebenen Parameter wie beispielsweise der Größe des Objektes und der maximalen Detektionsdistanz zum Objekt kann eine Filterung der jeweiligen Scanzeile erfolgen. Dies geschieht mit dem Ziel die Menge der Daten zu reduzieren und somit die Erzeugung der OctoMap, welche als Basis zur Kollisionsvermeidung herangezogen wird, zu beschleunigen und eine möglichst geringe Zeitverzögerung zwischen der Datenerzeugung und der Auswertung, dem Ray-Casting, zu erzielen.

Um eine Karte zu erhalten, die auch Daten vorangegangener Scans enthält, werden alle Punkte vom körperfesten Koordinatensystem (base\_link) ins globale Koordinatensystem (map) transformiert.

### Ray-Casting

Die eigentliche Hinderniserkennung findet anhand der durch die LiDAR Daten erzeugten OctoMap statt. Dabei wird ausgehend von insgesamt 5 Positionen entlang den Abmessungen des Flugsystems eine Ray-Casting Operation durchgeführt. Die Startpositionen sind in der nachfolgenden Grafik abgebildet und werden auf Basis der aktuellen Position des Flugsystems berechnet. Eine Beschränkung der durchgeführten Ray-Casting Operationen ist nötig, da nur eine begrenzte Kapazität an Rechenleistung zur Verfügung steht und insbesondere Ray-Casting Operationen als rechenintensiv gelten. Mit der Begrenzung der Ray-Casting Operationen geht jedoch auch eine Anpassung der Auflösung der OctoMap einher. Denn diese muss so aufgelöst sein, dass alle Voxel in Flugrichtung, die der Querschnittsfläche des Flugsystems entsprechen auch durch die ausgesendeten Strahlen (Rays) berührt bzw. geprüft werden. Betrachtet man nun die in der Abbildung dargestellte Anordnung der Positionen, so wird deutlich, dass die Auflösung der OctoMap, die Kantenlänge eines Voxels, für die zur Hindernisvermeidung genutzten OctoMap, im Idealfall größer ist als der Durchmesser der Stirnfläche des unbemannten Flugsystems. Dadurch kann sichergestellt werden, dass alle relevanten Voxel mit einer geringen Anzahl von Ray-Casting Operationen untersucht werden können.

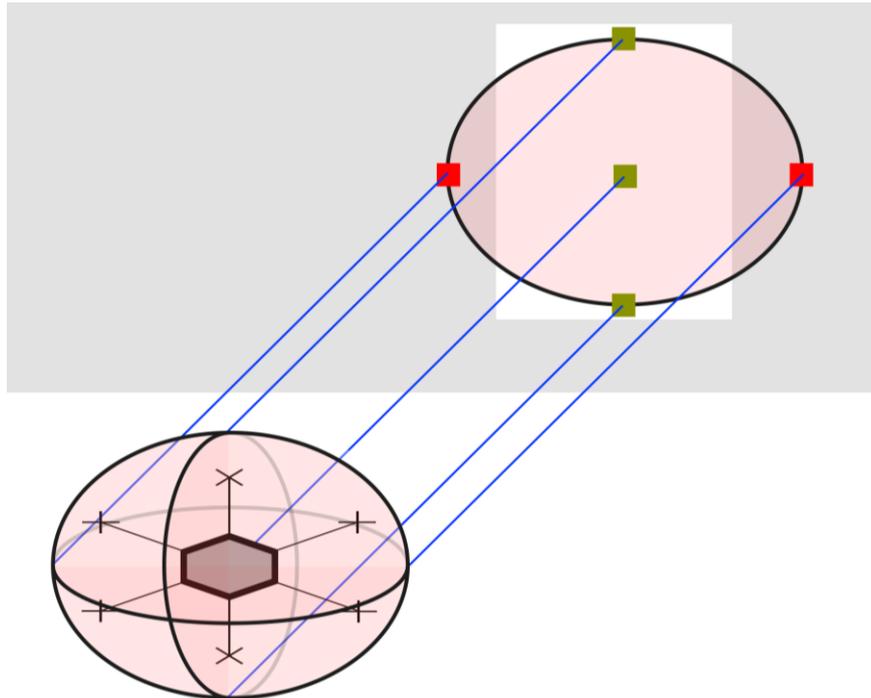


Abbildung 6.8.13: Ausgangspositionen der Ray-Casts

### Vektorfeld - Histogramm

Durch die Umsetzung des Ray-Casting auf Basis der OctoMap konnte eine Hinderniserkennung mit Kollisionsvermeidung umgesetzt werden. In Erweiterung dazu erfolgte die Umsetzung eines Ansatzes, welcher in der Lage ist das unbemannte Flugsystem kollisionsfrei auf Basis der vorhandenen Kartendaten, in Form der OctoMap, durch die Kommandierung von Geschwindigkeiten heranzuleiten. Dazu muss der Algorithmus auch in der Lage sein die kinematischen Grenzen des Systems einzubeziehen. Eine Möglichkeit für bodengebundenen Systeme liefert der in [2] vorgestellte 3DVFH+ Algorithmus, welcher selbst eine Weiterentwicklung der Vector Field Histogramm (VFH) Methode [3] ist und zu der Kategorie der auf einem mathematischen Modell basierenden Algorithmen gehört.

Für die Anwendung auf Flugsysteme wurde der 3DVFH+ Algorithmus erweitert, denn dieser widmete sich lediglich der Implementierung für ein bodengebundenen System.

Somit wurde der in [2] vorgestellte Ansatz erweitert um auch auf Flugsysteme angewendet zu werden.

Die bei der Erweiterung zum Tragen kommenden Beziehungen wurden aus dem ursprünglichen rein horizontalen Ansatz in [2] abgeleitet. Dabei wurde das kinematische Modell eines Flächenfliegers angenommen, welcher in seinen Flugbereichsgrenzen eine vertikale Positionsänderung nur in Kombination mit einer horizontalen Positionsänderung vollführen kann. So erfolgt im Gegensatz zum Anwendungsfall für ein bodengebundenen Fahrzeug nicht nur die Berechnung eines Azimutwinkels  $\alpha_z$  sondern auch die Berechnung eines Elevationswinkels  $\alpha_e$ .

Dabei werden zur Berechnung der Winkel die Koordinaten des Vehicle Center Points (VCP) und des jeweiligen Voxels  $i$  herangezogen. Der Satz der zu betrachtenden Voxel wird durch die um das Vehikel aufgespannten Bounding Box begrenzt. Die Winkel berechnen sich demnach wie folgt:

$$\alpha_z = \arctan\left(\frac{y_i - y_0}{x_i - x_0}\right)$$

$$\alpha_e = \arctan\left(\frac{z_i - z_0}{p}\right) \text{ mit } p = \sqrt{(x_i - x_0)^2 + (y_i - y_0)^2}$$

Zur Transformation in die für das Histogramm nötigen Winkelsektoren mit  $z$  Längengraden  $\beta_z$  und  $e$  Breitengraden  $\beta_e$  wurde folgende Beziehung definiert:

$$z = 2e$$

Somit ist die Winkelauflösung in der Horizontalen doppelt so groß wie in der Vertikalen und für die Umrechnungskonstante  $\alpha$  ergibt sich folgendes Verhältnis:

$$\alpha = \frac{\pi}{e} = \frac{2\pi}{z}$$

Aus der Berechnung des Azimutwinkels ergeben sich Werte im Wertebereich von  $\alpha_z \in [\pi; -\pi]$ . Um in den Grenzen von  $[0; 2\pi]$  für den korrespondierenden Histogrammbereich  $\beta_z \in [0; z - 1]$  bei gleich bleibenden Nullpunkt zu bleiben, wird folgende Vorschrift angewandt:

$$tfAngle(\alpha_z) = \begin{cases} -\alpha_z & \text{wenn } \alpha_z \leq 0 \\ 2\pi - \alpha_z & \text{wenn } \alpha_z > 0 \end{cases}$$

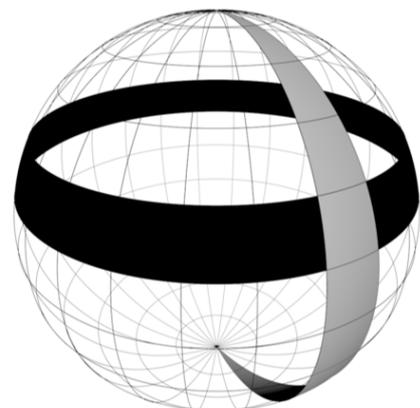
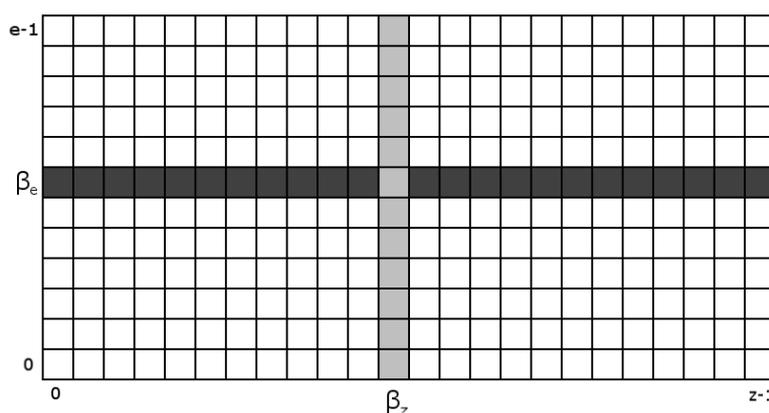
Mit der Umrechnungskonstante ergibt sich somit für den Längengrad  $\beta_z$  folgende Umrechnung:

$$\beta_z = \frac{tfAngle(\alpha_z)}{\alpha}$$

Der Wertebereich für den Elevationswinkel und somit des Breitengrades muss ebenfalls berücksichtigt werden. Für diesen wird der Wertebereich zunächst von  $\alpha_e \in [-\frac{\pi}{2}; \frac{\pi}{2}]$  zu  $[0; \pi]$  verschoben. Für den Breitengrad  $\beta_e \in [0; e - 1]$  genügt demnach die Addition von  $\pi/2$ .

$$\beta_e = \frac{\alpha_e + \frac{\pi}{2}}{\alpha}$$

Das aus den oben aufgeführten Gleichungen erstellbare zweidimensionale Histogramm ist in der nachfolgenden Abbildung 6.6.14 dargestellt. Dieses berücksichtigt noch nicht die vollständige Kinematik des Systems, sondern lediglich physische Hindernisse innerhalb der Bounding Box.



**Abbildung 6.8.14: 2D polares Histogramm abgeleitet aus den Polarkoordinaten einer Kugeloberfläche**

Zusammengefasst werden zur Erzeugung des Histogramms und der Hindernisvermeidung folgende Schritte periodisch durchgeführt:

1. Erkundung der OctoMap

2. Berechnung des 2D Histogramms
3. Berechnung des effizientesten Pfades
4. Kommandierung von Geschwindigkeiten

Bei der Erkundung der OctoMap wird lediglich ein kleiner Ausschnitt der Karte herangezogen. Dieser wird als Bounding Box bezeichnet. Dabei kann die Größe der Bounding Box frei gewählt werden. Im Idealfall berücksichtigt die Bounding Box in ihrer Ausdehnung schon den Planungsintervall und den Geschwindigkeitsvektor des Systems in einem gewissen Maße. Sind diese bekannt, so kann die Festlegung der Abmessungen der Bounding Box dynamisch erfolgen, andernfalls müssen Grenzwerte der Geschwindigkeiten zur Bildung der Bounding Box herangezogen werden. Letzterer Ansatz wurde innerhalb der Projektes umgesetzt, mit dem Resultat, dass die Bounding Box in jedem Schritt identische Abmessungen besitzt. Dabei wird die Bounding Box immer um das Vehikel aufgespannt.

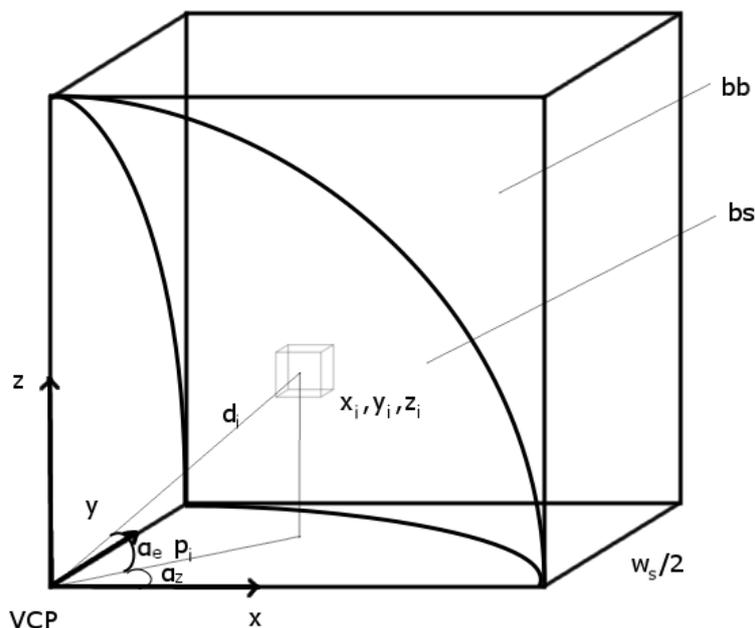
Iterativ werden alle Voxel innerhalb dieser quadratischen Bounding Box analysiert. Jedoch werden nicht alle Voxel in der Bounding Box zur Erstellung des Histogramms herangezogen. Es werden nur die Voxel herangezogen welche der folgenden Bedingung genügen.

$$d_i \leq \frac{w_s}{2}$$

Dabei entspricht  $d_i$  der euklidischen Distanz zwischen dem Vehicle Center Point  $VCP(x_0, y_0, z_0)$  und dem aktuell zu analysierendem Voxel an den Koordinaten  $x_i, y_i, z_i$ .

$$d_i = \sqrt{(x_0 - x_i)^2 + (y_0 - y_i)^2 + (z_0 - z_i)^2}$$

Somit wird aus der Bounding Box eine Bounding Sphere wie in der nachfolgenden Abbildung dargestellt.



**Abbildung 6.8.15: 1/8 Ausschnitt einer Bounding Box (bb) bzw. Bounding Sphere (bs)**

Nur die Voxel welche innerhalb der Bounding Sphere liegen werden letztendlich zur Erzeugung des Histogramms verwendet.

Die Implementierung wurde sowohl anhand eines Indoor – als auch Outdoorszenario getestet. Dabei wurde der Start- und Zielpunkt jeweils so gewählt, dass ein oder mehrere Hindernisse auf dem direkten Weg zum Zielpunkt liegen. In den nachfolgenden Abbildungen ist neben der OctoMap auch das Histogramm als Terminalausgabe dargestellt. Neben den binären werten

des Histogramms enthält die Terminalausgabe auch die Position des Vehikels und die berechnete anzusteuende Position innerhalb des Histogramms zur Umgehung des Hindernisses. Für die Versuche wurde auf veröffentlichtes und frei zugängliches Kartenmaterial der Albert-Ludwigs-Universität Freiburg zurückgegriffen.

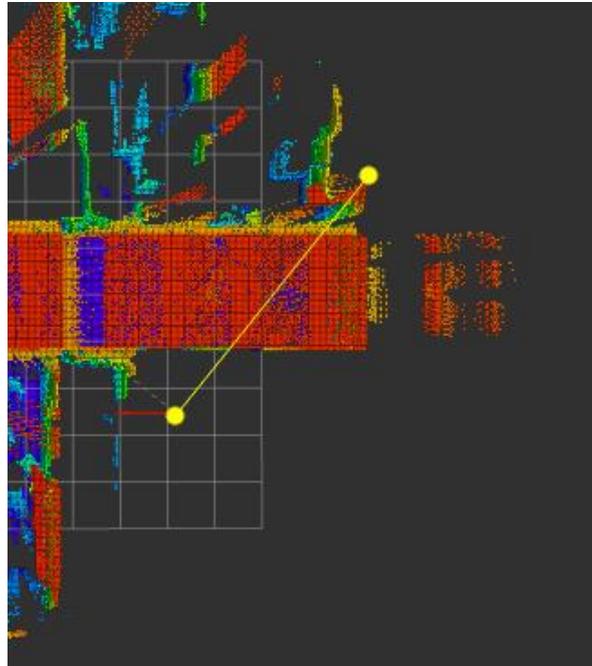


Abbildung 6.8.16: Start- und Zielpunkt des Outdoorszenarios (FR-079 corridor [5])

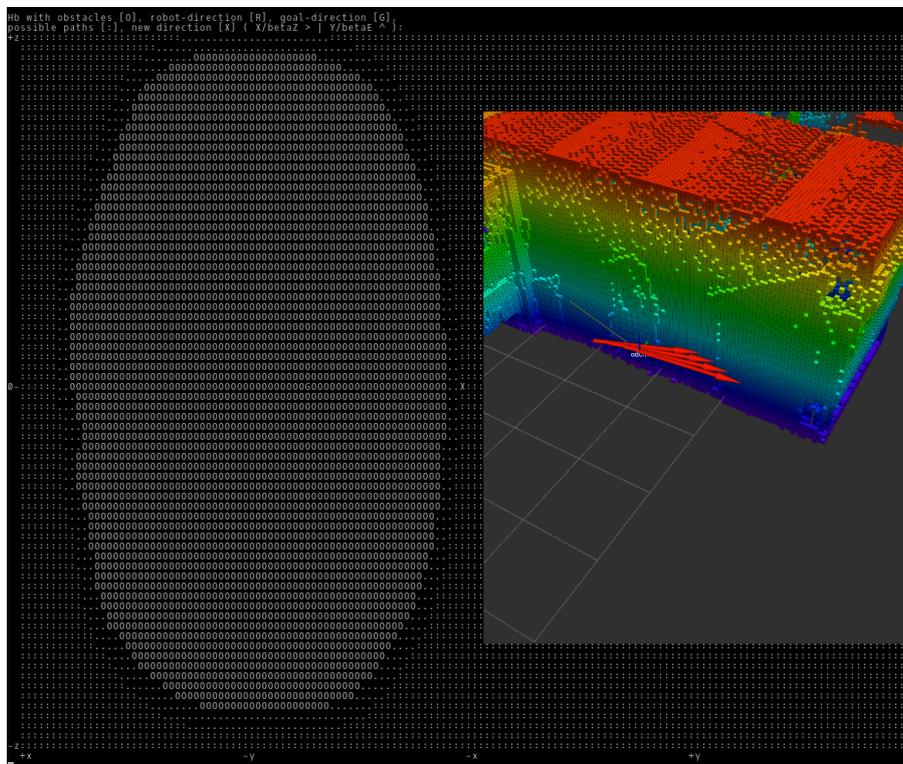


Abbildung 6.8.17: Outdoorszenario mit Histogramm (links) und OctoMap (rechts) inkl. Richtungsvektor des Vehikels

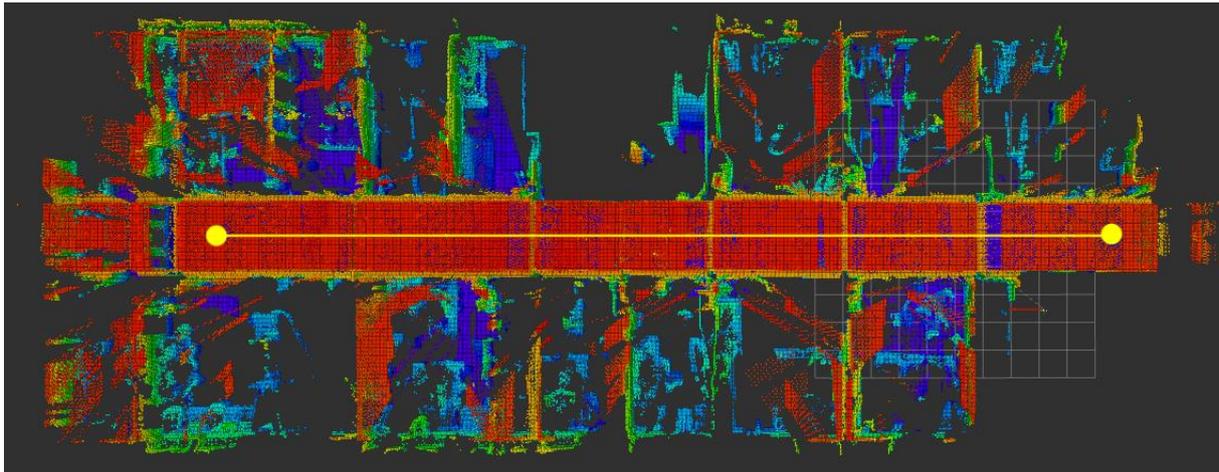


Abbildung 6.8.18: Start- und Zielpunkt des Indoorszenarios (FR-079 corridor [5])

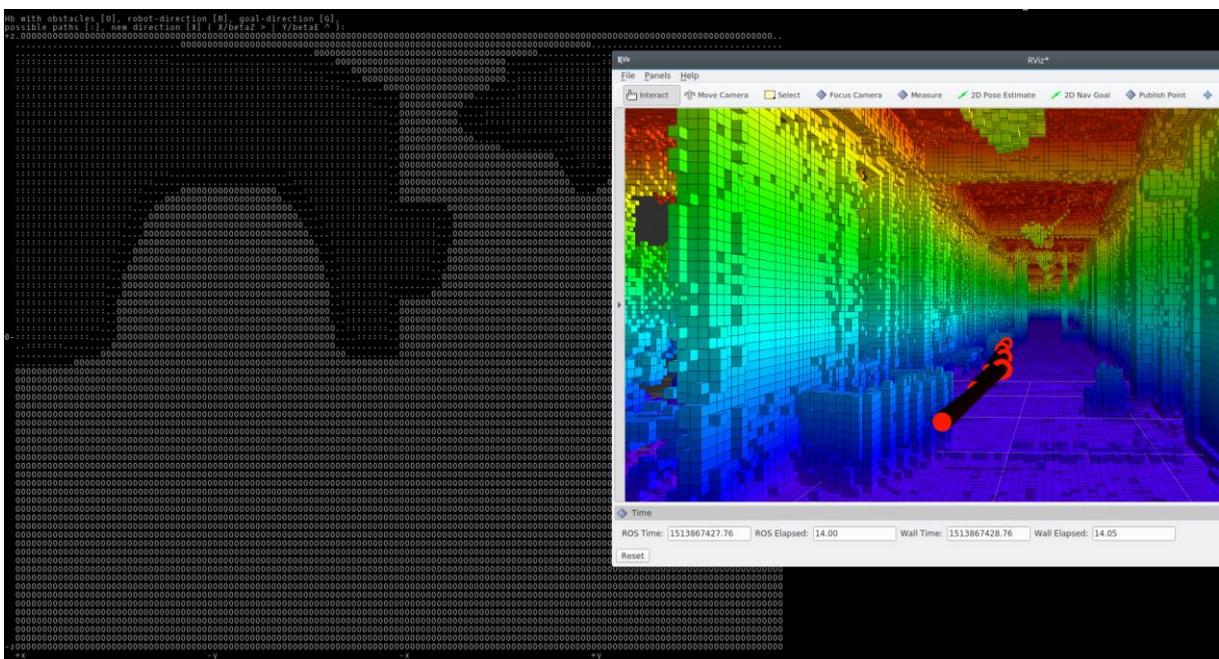


Abbildung 6.8.19: Binäres Histogramm des Indoorszenarios (links) und OctoMap (rechts) inkl. Richtungsvektor des Vehikels

## 6.8.4 Diagnosesystem, Supervision System und Missionsintelligenz

### 6.8.4.1 Allgemeines

In diesem Abschnitt soll die Funktionalität, den Aufbau und die Bedienung der CoSMiC Bodenstation beschrieben werden. Dazu werden einführend die Begriffe Sicherheitspilot, Operator und Agent definiert. Nachfolgend soll dann zunächst auf die Herausforderungen und die Motivation hinter den getätigten Entwicklungen erläutert werden, bevor auf technische Details und der Aufbau der CoSMiC Bodenstation eingegangen wird.

**Sicherheitspilot** = Person, die ein System mit z.B. Fernsteuerung direkt bedienen kann und im Normalfall lediglich eine überwachende Funktion ausführt. Soll in einem Fehlerfall die Kontrolle übernehmen können, muss dafür aber auch einen Fehlerfall frühzeitig und eindeutig erkennen können.

**Operator** = Person an der Bodenstation zur Überwachung des Missionsfortschrittes und zur Kommandierung von übergeordneten Missionszielen.

**Agent** = ist im Allgemeinen ein System, welches eingegebene Informationen verarbeitet und eine entsprechende Ausgabe produziert.

#### a) **Standardaufgaben eines Sicherheitspiloten und Operators**

Die Analyse von Aufgaben eines Sicherheitspiloten und Operators diente im Projekt dazu sinnvolle zu entwickelnde Funktionen für das Supervision System zu definieren. Dazu wurden im Rahmen der Anforderungsanalyse Sicherheitspiloten und Operatoren befragt und eine Liste an Standardaufgaben zusammengetragen:

- Telemetrie Verbindung herstellen und Qualität kontinuierlich prüfen
- Initialisierungen abwarten, ggf. Fehler lesen und interpretieren
- Sporadisch: Parameter überprüfen
- Umgebungsbedingungen kontinuierlich prüfen (Wetter, Wind, Start/Landeplatz)
- Startbedingungen prüfen / Zustände des Agenten überprüfen (z.B. Heading, Lage)
- Mission planen und hochladen
- Fehlermeldungen lesen und interpretieren
  - Ggf. Handlung durchführen, Abbruch, Landung, RTL
- Mission (geplante Wegpunkte, Höhen, Aktionen) prüfen
- Ablauf der Mission überwachen – fliegt der Agent entlang der geplanten Route?
- Zustände des Agenten überwachen, Energie, Lagewinkel, Geschwindigkeiten

#### b) **Herausforderungen**

Im Rahmen der Anforderungsanalyse wurde Herausforderungen für Sicherheitspiloten und Operatoren abgeleitet, die bei Feldversuchen mit unbemannten Agenten, welche mit einem hohen Automatisierungsgrad betrieben werden, entstehen:

##### **Sicherheitspiloten:**

- Kennt nicht alle Zustände des Agenten:
  - Modi (z.B. Messung oder Fahrt)
  - Fehlerzustände (z.B. Sensorfehler)
  - Ortungszustände (z.B. Genauigkeiten)
  - Kommunikation und Schwarmverhalten
- Kennt nicht die Absichten des automatisierten Agenten:
  - Modi (z.B. Messung im Gange, Landung etc.)
  - Nächster Wegpunkt, beabsichtigter Flugweg

### Operator:

- Verfügbare GUIs können oftmals nicht mehrere Agenten gleichzeitig zur Überwachung anzeigen und/oder haben keine Übersicht für den vertikalen Verlauf der Mission
- Operator kann nur sehr schwer alle Systemzustände gleichzeitig und/oder die Abhängigkeiten zwischen einzelnen Zuständen überwachen
- Operator kann nur sehr schwer anhand von Systemzuständen eine optimale Reaktion ableiten.
- Zu viele Informationen für Operator
- Ableitung von Maßnahmen nicht optimal
- Kennt ggf. nicht alle Konditionen
- Fehler durch Operator möglich
- Durch Überforderung, zu geringe Aufmerksamkeit usw.
- Problemlösung durch Operator zu langsam
- Planungszeit und Latenz
- Die Standardaufgaben eines Operators sollen automatisiert durchgeführt werden von einem Agenten

#### 6.8.4.2 Assistenzsystem

Das Ziel der in CoSMiC entwickelten Bodenstation war es sowohl die Sicherheitspiloten als auch die Operatoren mit bestimmten Funktionen zu unterstützen, um somit die sichere Erprobung des Schwarms auf der Erde zu gewährleisten und gleichzeitig eine Grundlage zu schaffen für die Entwicklung einer Bodenstation für eine echte Mission. Das System ist grundsätzlich in der Lage alle Systemzustände, die über einen Sensor erfasst werden können zu überwachen und bei Überschreitung von festgelegten oder angelernten Grenzwerten den Benutzer zu informieren. Um dieses Ziel zu erreichen müssen zunächst die Zusammenhänge und die Grenzwerte der Zustandsgrößen für jeden Agenten recherchiert und interpretiert werden. Dann werden diese dem Diagnosesystem zur Verfügung gestellt.

Die nächste Abbildung zeigt das Vorgehen bei der Verwendung eines wissensverarbeitenden Systems. Grundlage sind Wissensquellen die zum einen Gespräche mit Sicherheitspiloten und zum anderen technische Daten der Fahrzeuge darstellen. Das Wissen muss im nächsten Schritt richtig abgebildet werden indem es richtig erfasst und strukturiert wird, danach folgt die Formalisierung in einer Wissensdatenbank. Wird nun ein bestimmtes Problem verarbeitet kann mit Hilfe der Datenbank eine Lösung erarbeitet und diese dem Benutzer dargestellt werden.

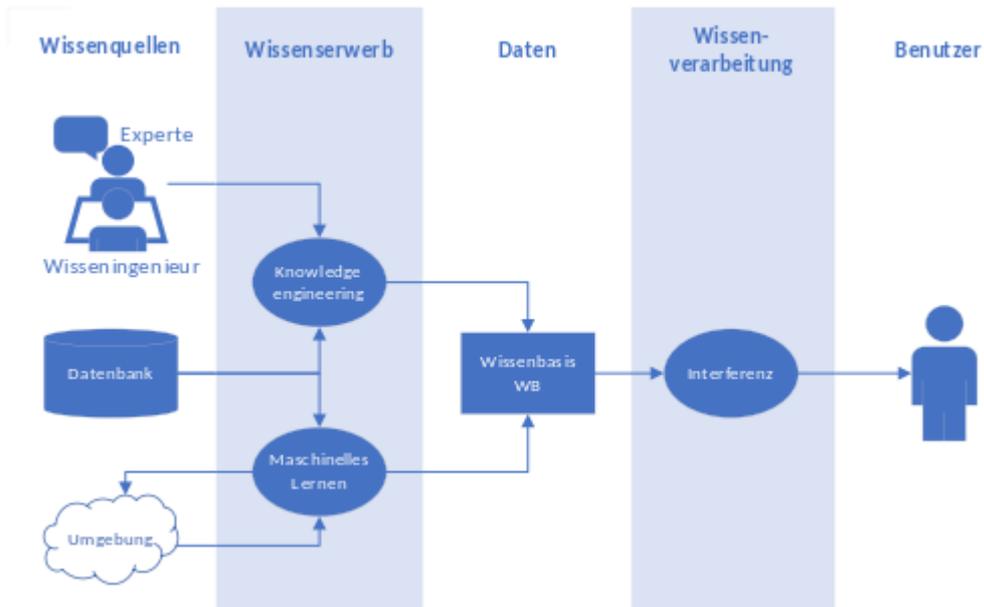


Abbildung 6.8.20: Grundlage wissensverarbeitendes System

### c) Automatisierungsgrade

Die folgende Übersicht zeigt die verschiedenen Automatisierungsgrade die in der Robotik definiert sind. In VaMEx CoSMiC wurden die Automatisierungsgrade IV bis VI betrachtet. Vor allem bei den ersten Erprobungen von experimentellen Flugsystemen wurden auf den Automatisierungsgrad IV zurückgegriffen, bei dem das System Vorschläge macht, die der Benutzer allerdings erst bestätigen muss, bevor die Handlung ausgeführt wird. Bei der Abschlussdemonstration wurde letztlich gezeigt, dass es möglich ist den Schwarm im Automatisierungsgrad VI zu betreiben.

Stufe	Beschreibung	Begriff
I	Ruhende Automation	
II	Automation schlägt alle möglichen Handlungen vor	
III	Automation schlägt ausgewählte Handlung vor	
IV	Automation schlägt ausgewählte Handlung vor, die vom Bediener bestätigt werden muss	management-by-consent (MbC)
V	Automation handelt bis es explizit verboten wird	management-by-exception (MbE)
VI	Automation handelt ohne Möglichkeit zu intervenieren	fully automatic

Automatisierungsgrad

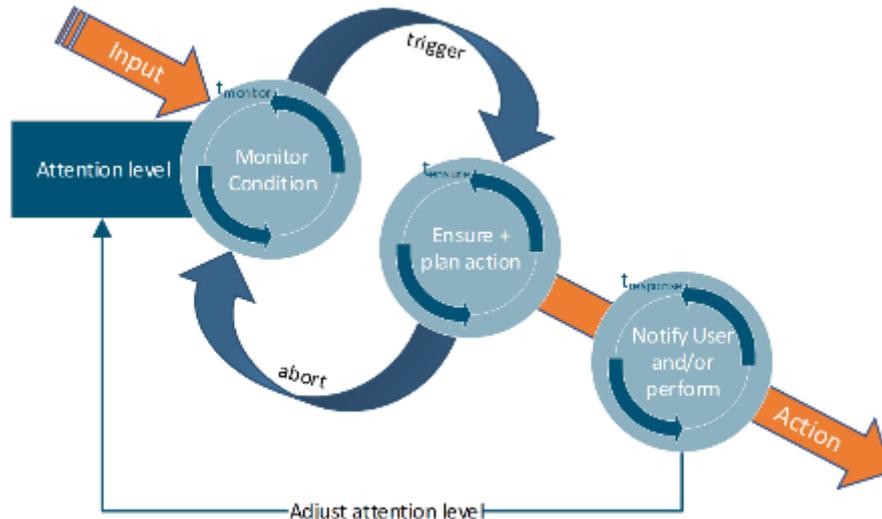
Beteiligung durch Mensch

Abbildung 6.8.21: Automatisierungsgrade in der Robotik

### d) Datenstruktur und Prozess zur Generierung von Vorschlägen und Handlungen

Die Daten von Sensoren und der Zustandsüberwachung (Input) wird vom Diagnosesystem geliefert und im nächsten Schritt in einem für jeden Zustand definierbaren Zeitintervall überwacht (Monitor Condition). Wird eine Abweichung eines Zustandes festgestellt, wird im nächsten Schritt sichergestellt, dass die Abweichung kein Resultat eines kurzzeitigen Effektes war. Sollte dies der Fall gewesen sein, wird der weitere Prozess abgebrochen und zurück zur Überwachung gesprungen. Sollte die Abweichung beständig sein, wird abhängig vom eingestellten Automatisierungsgrad entweder der Nutzer informiert oder eine Handlung

durchgeführt. Danach wird ggf. die Priorität des überwachten Zustandes angepasst. In der nachfolgenden Abbildung ist dieser Prozess dargestellt. Mit Hilfe dieses Systems kann flexibel für jede zu überwachende Zustandsgröße die entsprechenden Parameter, Texthinweise, Zeitintervalle und abgeleiteten Handlungen hinterlegt werden.



**Abbildung 6.8.22: Ablaufdiagramm für die Generierung von Hinweisen und Handlungen**

Die Information des Benutzers erfolgt durch Hinweisfenster, die in der Nähe der Informationsfenster der Agenten angezeigt werden. Diese Hinweisfenster enthalten einen Informationstext und Optionen zur Auswahl für den Benutzer, die je nach eingestelltem Automatisierungsgrad ausgewählt werden oder die nach einer bestimmten Zeit vom System automatisch ausgewählt werden.

### e) Implementierung

Das Ziel für die Entwicklung des automatischen Diagnosesystems und der Einbezug aktueller Systeminformationen in Entscheidungen der Missionsintelligenz war die sichere Durchführbarkeit einer automatisch ablaufenden Mission mit unbemannten Systemen. Nachfolgend wird das Konzept für die Überwachung des automatischen Starts und die Überwachung flugmechanischer Grenzen beschrieben.

#### **Automatischer Start**

Für einen automatischen Start werden folgende Startbedingungen überwacht:

- GNSS 3D Fix (wird in VaMEx CoSMiC überschrieben vom Experimentalnavigationssystem)
- Akkuspannung größer als 95% der Ladeschlussspannung der im Fahrzeug verwendeten LiPo Akkus
- Missionsziel vorhanden (bzw. Wegpunkt)

#### **Überwachung flugmechanischer Grenzen**

##### a) Geschwindigkeiten

Die Motorleistungen werden überwacht und werden bei Überschreitung von Grenzwerten eine Warnung generiert. Die Steiggeschwindigkeit und/oder Horizontalgeschwindigkeit kann somit reduziert werden, falls Wind, Gewicht, Degradation, sinkende Luftdichte etc. dazu führen, dass die Motorleistung sich erhöht

und zu einer Überbelastung der Motoren führen würde.

#### b) Startgewichtsüberwachung

Motorleistung nach dem Start überwachen und auf Plausibilität prüfen. Wird eine bestimmte gemittelte Motorleistung während der Startphase überschritten, wird automatisch das Fluggerät umgehend in einen Landemodus versetzt. Somit wird des Antriebssystems vor einer Überlastung geschützt. Gründe dafür können sein:

- zu viel Wind,
- zu hohe Startmasse,
- zu geringe Luftdichte,
- Akkuspannung zu gering,
- Antriebssystem anderweitig degradiert.

Wenn während der Startphase einer der Motoren ein Limit überschreitet wird ebenfalls eine Landung kommandiert. Gründe dafür können sein:

- zu viel Wind,
- Propeller defekt oder falsch montiert,
- Antriebseinheit anderweitig degradiert.

#### 6.8.4.3 Aufbau der Bedienoberfläche

Die Entwicklung des Supervision Systems hatte zum Ziel eine einfache Darstellungsmöglichkeiten für die verschiedenen Parameter und Schätzgrößen innerhalb des Schwarm-Systems zu schaffen. Diese Darstellungsmöglichkeiten dienen zum einen während der Durchführung von Teil-Experimenten zum Debugging, zum anderen als vereinfachte Visualisierung für finale Experimente und Demonstrationen. Diese Informationen werden auf einer speziell entwickelten Bodenstation aufbereitet und angezeigt.

Ein weiterer Punkt ist die Darstellung und Überwachung der Absichten der automatisierten Systeme. Da die unbemannten Systeme in diesem Projekt mit einem hohen Automatisierungsgrad betrieben werden, sind die Absichten der einzelnen Schwarmteilnehmer nicht eindeutig voraussagbar. Die Sicherheitspiloten der Vehikel müssen jedoch wissen was das einzelne Vehikel plant, um ggf. eingreifen zu können.

Hierfür ist eine permanente Überwachung der missions- und fahrzeugrelevanten Systemzustände notwendig, welche durch ein geeignetes Diagnosesystem sichergestellt werden soll. Da nicht für alle Systemzustände und Fahrzeuge geeignete Sensoren zur Verfügung standen, konnte im Rahmen dieses Projektes die Überwachung nur für einzelne Subsysteme der Fahrzeuge entwickelt und implementiert werden.

#### f) Safety Area (Sicherheitsbereich)

Mit dieser Funktion kann ein Geo-Fence aufgespannt werden, in dem der Roboterschwarm agieren soll. Die Bodenstation überwacht die Einhaltung dieser Grenzen und generiert eine Warnung falls diese Grenze von einem System überschritten wird. Reagiert der Benutzer nicht auf die Warnung, wird das System automatisch zurück innerhalb der Grenzen geführt.

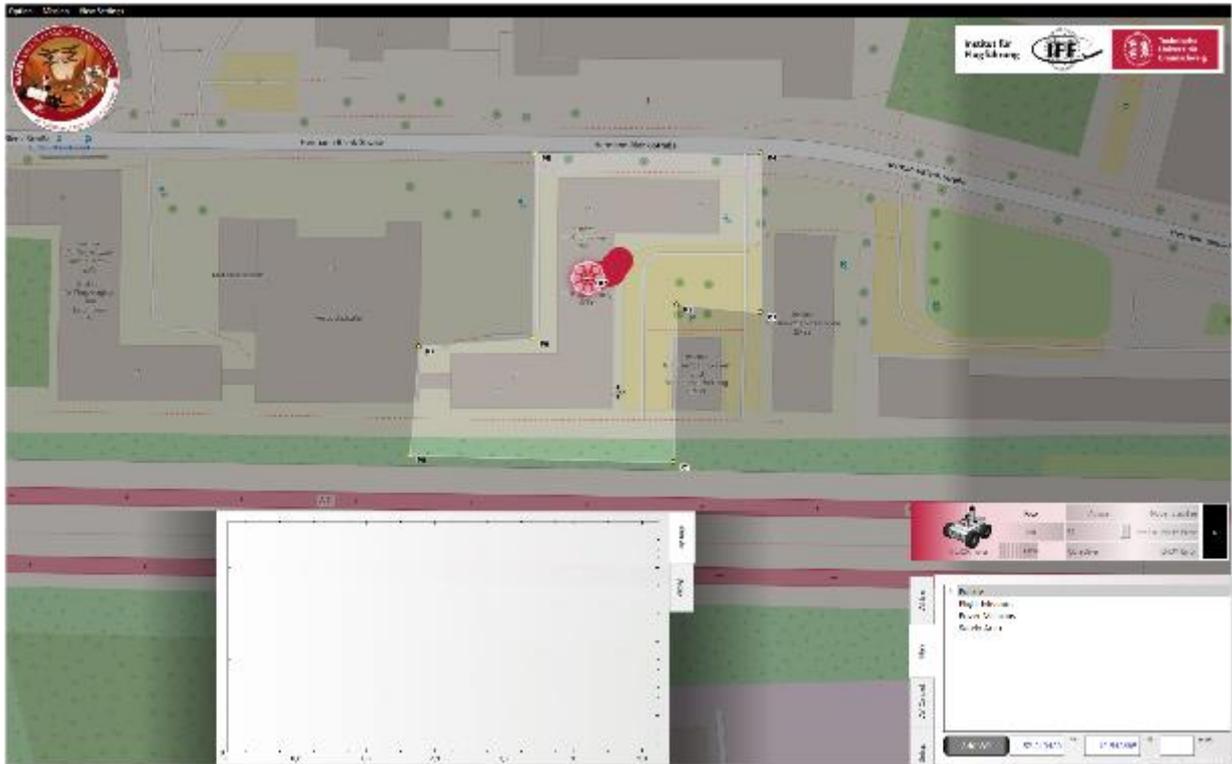


Abbildung 6.8.23: CoSMiC Bodenstation: Safety Area

### g) Missionsziele und Zustände

Die aktuellen Missionsziele, Wegpunkt und Flugpfade, sowie die aktuelle Position werden von jedem Rover mit einem festen Intervall ausgelesen und dem Benutzer angezeigt. Somit weiß der Operator/Sicherheitspilot was der Rover tun wird und kann gezielter überwachen.

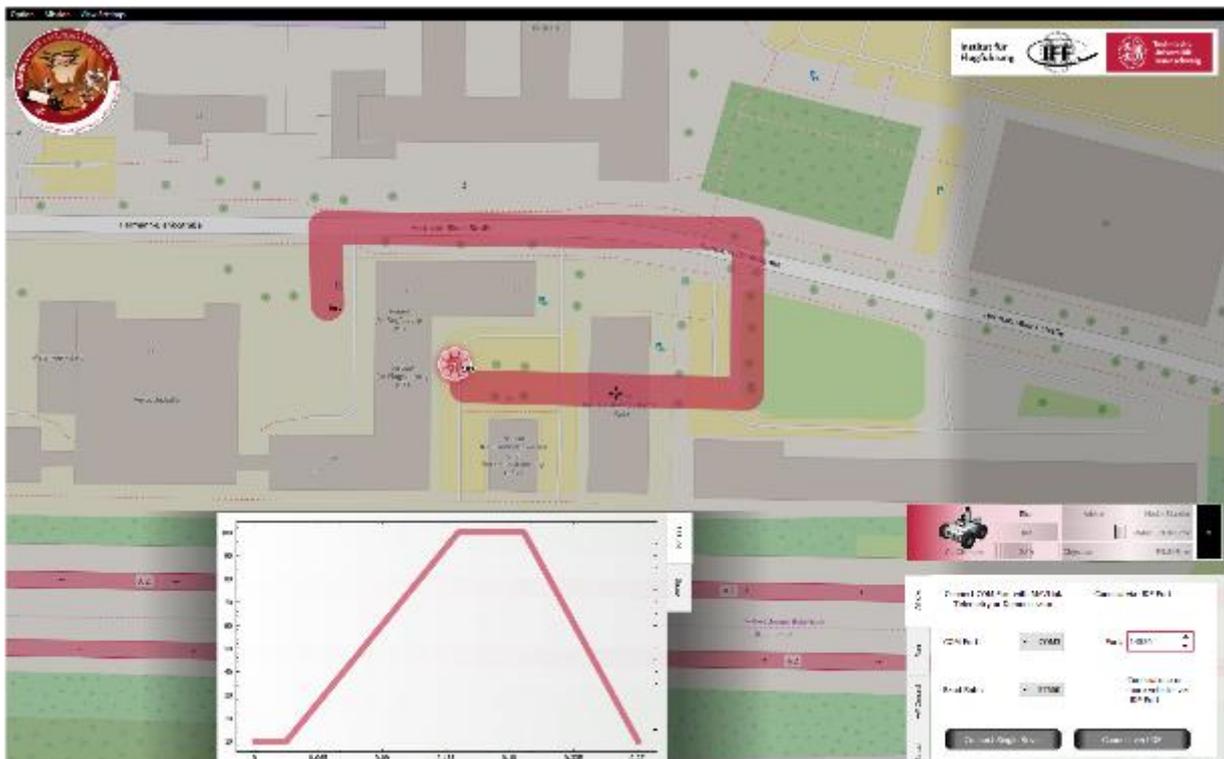


Abbildung 6.8.24: CoSMiC Bodenstation: Missionsziele und Zustände

## h) Schwarmüberwachung

Die vorgestellten Funktionen können wir einen Schwarm von Robotern angezeigt werden. Hinweise des Diagnosesystems werden in Form von Hinweisfenstern angezeigt. Je nach Automatisierungsgrad dienen diese zur Auswahl für den Operator oder lediglich zur Information des Operators.

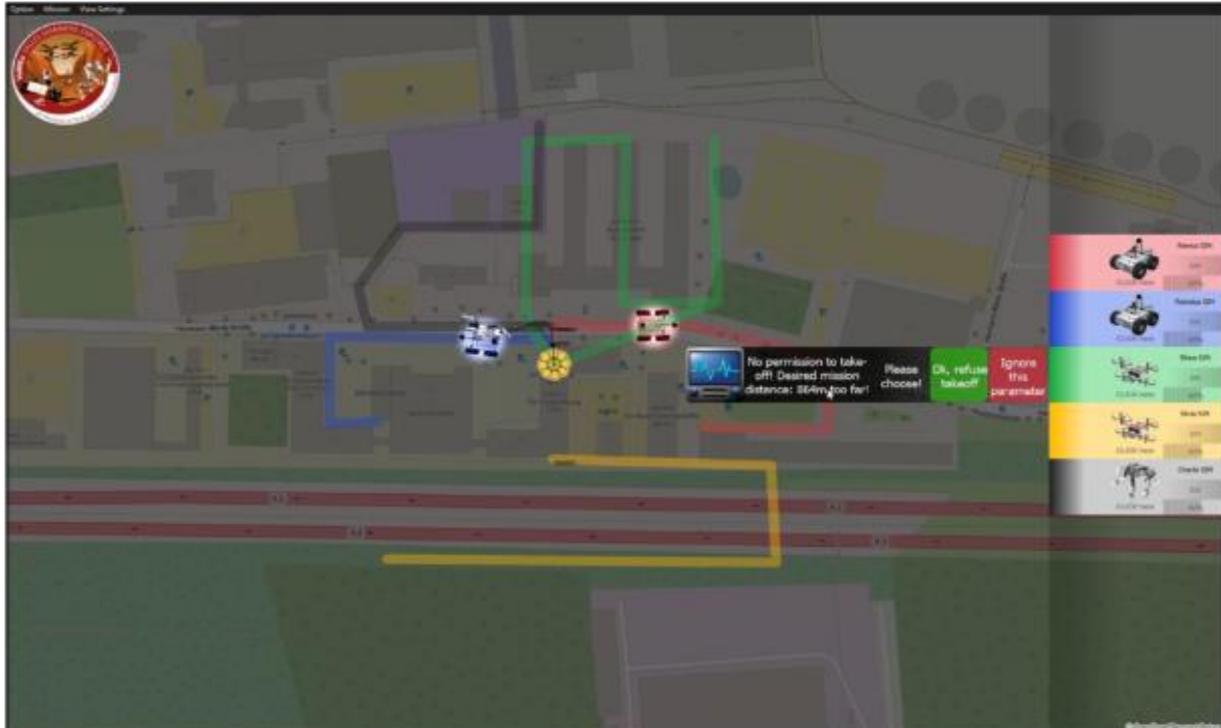


Abbildung 6.8.25: CoSMiC Bodenstation: Schwarmüberwachung

### 6.8.4.4 Direktor

Zusätzlich wurde eine Missionsintelligenz entwickelt, welche zwischen dem Operator und dem Schwarm agiert und so den Operator entlastet und zeitkritische Entscheidungen vorwegnehmen kann. Somit kann mit sehr geringen Latenzzeiten auf plötzlich eintretende Ereignisse reagiert werden.

Der Direktor hat folgende Aufgaben:

- Initiale Missionsplanung:
  - Aufteilung des Explorationsgebietes in Segmente
  - Größe der Segmente abhängig von Roverleistungen
- Eingreifen wenn nötig:
  - Ausfall eines Rovers → Übergabe der bisherigen Aufgaben an einen anderen Rover
  - Ausfall eines Rover → Inspektion durch einen anderen Rover z.B.

Die Folgende Abbildung zeigt die initiale Missionsplanung mit Unterstützung des Direktors. Der Operator muss lediglich die äußeren Grenzen des Explorationsgebietes zeichnen. Der Direktor teilt dieses Gebiet nach Leistung der Agenten selbstständig auf und generiert automatisch Wegpunkte zur Exploration dieses Gebietes. Diese werden in Missionen für jeden Agenten zusammengefasst und an die jeweiligen Agenten geschickt.



Abbildung 6.8.26: Automatische Missionsgenerierung des Direktors

## 6.9 Versuchsdurchführung

### 6.9.1 Schwarmnavigation und Schwarmexploration



Abbildung 6.9.1: Ein Teil des VaMEx-CoSMiC-Teams (TUM-KN und DLR-KN) im Versuchsgelände nach ersten gemeinsamen Tests.

#### 6.9.1.1 Schwarmlokalisierung

Zur Demonstration der Schwarmlokalisierung und des Vorteils der Kooperation im Netzwerk wurde der Experimentalaufbau basierend auf SDRs, den Rovern, und dem Partikelfilter in eine Experimentumgebung (Kiesgrube) gebracht. Der Partikelfilter wurde um eine grafische Ausgabe erweitert, siehe Abbildung 6.9.2, um sowohl die Hypothesen des Partikelfilters, als auch die geschätzten Positionen und tatsächlichen Position der Rover anzuzeigen zu können. Mit Hilfe des GNSS-RTK-Moduls auf jedem Rover konnte auch der Fehler der Positionsschätzung in Relation zu den GNSS-RTK-Schätzungen gezeigt werden.



Abbildung 6.9.2: Grafische Darstellung der Schwarmlokalisierung bei einer Demonstration.

Die Rover wurden manuell gesteuert. Abbildung 6.9.3 zeigt die Experimentumgebung für die Schwarmlokalisierung zu Beginn des Experiments. Zur Fixierung des Koordinatensystems wurden drei Ankerknoten platziert. Am rechten Rand des Erkundungsgebiets befindet sich eine kleine Hugelkette als Hindernis. Dies fuhrt dazu, dass die Funksignale von Anker 1 und Anker 3 abgeschattet werden. Eine Lokalisierung nur basierend auf den drei Ankerknoten wurde in diesen abgeschatteten Bereichen nicht funktionieren. Hier zeigt sich der Vorteil des Schwarms und der Kooperation im Netzwerk. Die Rover konnen sich gegenseitig unterstutzen, sodass ein oder mehrere Rover auch in diese ursprunglich abgeschatteten Bereiche gefuhrt und lokalisiert werden konnen. Abbildung 6.9.4 zeigt den Vorteil der Kooperation. Zwei Rover befinden sich in einem abgeschatteten Bereich und konnen Anker 1 nicht empfangen. Mit Hilfe der Distanzschatzungen zu den Nachbarrovern konnen diese dennoch genau lokalisiert werden.



**Abbildung 6.9.3: Beginn des Experiments zur Schwarmlokalisierung. Graue Punkte zeigen die Hypothesen des Partikelfilters, rote Rauten die geschatzte Position und grune Dreiecke die tatsachliche Position.**



**Abbildung 6.9.4: Zwei Rover befinden sich nun im abgeschatteten Bereich und konnen Anker 1 nicht mehr empfangen. Die Kooperation mit den umgebenden Rovern erlaubt dennoch eine Lokalisierung. Graue Punkte zeigen die Hypothesen des Partikelfilters, rote Rauten die geschatzte Position und grune Dreiecke die tatsachliche Position.**

### 6.9.1.2 Gasexploration

Die in ROS realisierte Gasexploration als HIL-Simulation wurde in einer Experimentumgebung (Kiesgrube) demonstriert. Für die lokale Kommunikation wurde die gemeinsame Funkbasisstation und für die Ortsinformation das GNSS-RTK verwendet. Abbildung 6.9.5 zeigt die zwei virtuell simulierten Gasquellen als Überlagerung mit einem Foto der Kiesgrube. Zusätzlich zur Ausströmung des Gases wurde ein Wind von rechts nach links im Bild wehend simuliert.



**Abbildung 6.9.5: Experiment zur Gasexploration im Schwarm. Die Ausströmungen der zwei Gasquellen sind als roter Rauch überlagert und ein Wind weht von rechts nach links.**

Alle Rover starten in einem kleinen Bereich am Rand des Explorationsgebietes und schwärmen für Gaskonzentrationsmessungen aus. zeigt die tatsächliche messbare Gaskonzentration. Der Wind führt zu einer erhöhten Konzentration links der Quellen und somit zu einer nicht symmetrischen Konzentrationsverteilung um die zwei Quellen.



**Abbildung 6.9.6: Beginn der Gasexploration: die tatsächliche Gaskonzentration ist als Überlagerung dargestellt.**

Abbildung 6.9.7 zeigt den zeitlichen Verlauf der Exploration an Hand von einzelnen Momentaufnahmen.

Die Karte zur geschätzten Quellenverteilung ist zu Beginn noch leer. Die Unsicherheit ist zu Beginn sehr hoch und hat die höchsten Werte auf der windabgewandten Seite. D.h. die Agenten werden zu Beginn auf Grund der Explorationsstrategie vermehrt Messungen in

diesem Bereich durchführen.

Nach wenigen Minuten der Exploration hat der Schwarm bereits die erste Quelle identifiziert, siehe Abbildung 6.9.7 und wenig später danach die zweite Quelle.

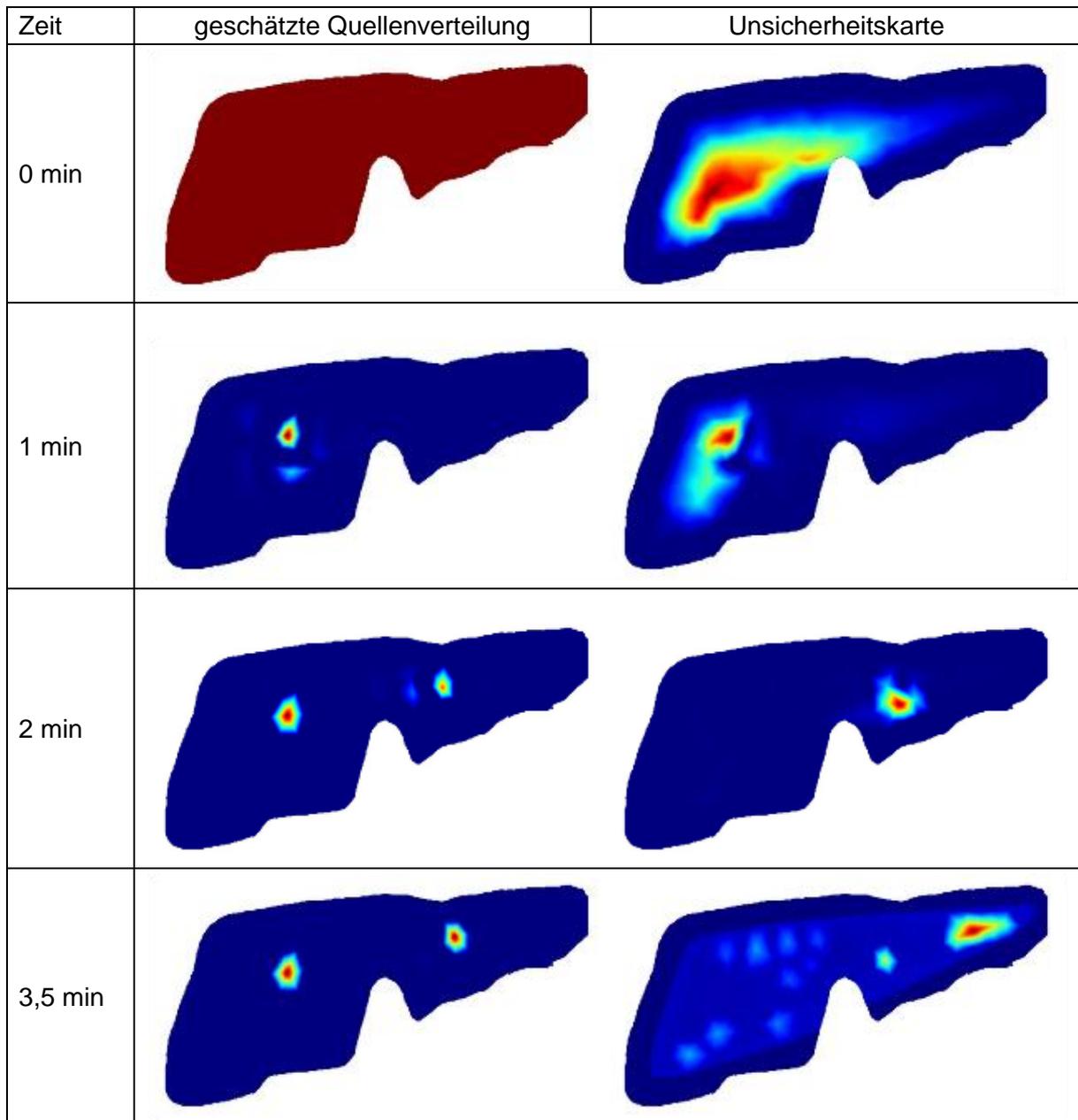


Abbildung 6.9.7: Verlauf der Gasexploration: Die einzelne Momentaufnahmen zeigen die geschätzte Gasquellenverteilung (links) und die berechnete Unsicherheitskarte (rechts).

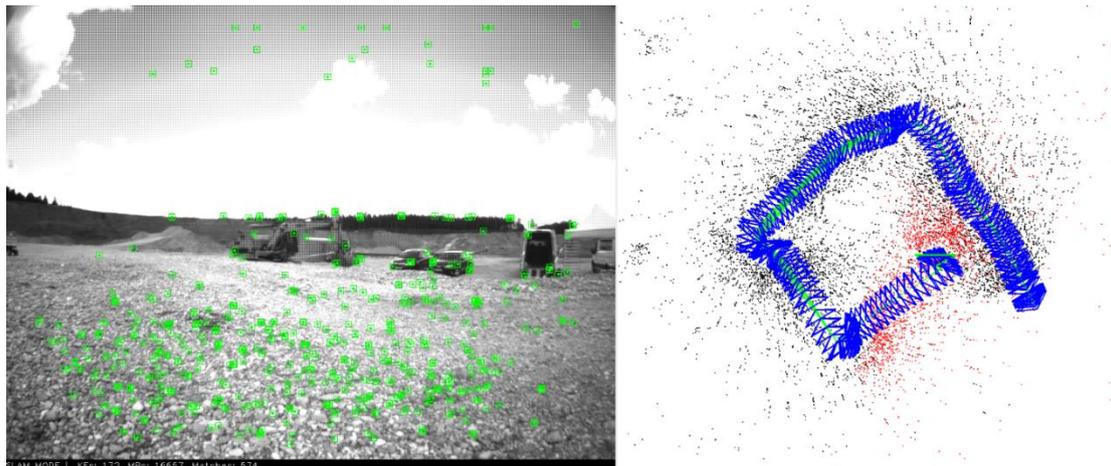
### 6.9.1.3 Exploration mit visuellem SLAM und Sensorfusion

TUM-NAV implementierte den entwickelten Sensor-Fusionsalgorithmus mit dem hochmodernen visuellen SLAM-Tool (ORB-SLAM) und zeigte die Leistungsfähigkeit mit dem integrierten System des Sensorpakets von TUM-NAV und dem Rover des DLR-IKNs in einer Mars-ähnlichen Umgebung, d.h. einer Kiesgrube. Abbildung 6.9.8 zeigt den Versuchsaufbau (dynamische und statische Rover, Basisstation und Bodenstation) und ein vor den Experimenten geplantes Missionsszenario.



**Abbildung 6.9.8: Experimenteller Messaufbau und geplantes Missionszenario**

Die Posen des dynamischen Rovers sowie die Kartenpunkte wurden am Bordcomputer geschätzt und mit dem aktuell beobachteten Bild an den Bodenstationscomputer übertragen. Der Benutzer konnte die geschätzte Trajektorie mit den Kartenpunkten und der aktuellen Szene wie in Abbildung 6.9.9 dargestellt überwachen.



**Abbildung 6.9.9: Eine Momentaufnahme des Echtzeit-Monitoring-Systems der Bodenstation**

## **7 Wichtigste Positionen des zahlenmäßigen Nachweises**

### **7.1 TU Braunschweig**

Aus den bewilligten Mitteln wurden an der TU Braunschweig im Wesentlichen wissenschaftliche Mitarbeiter beschäftigt. Diese waren zur Bearbeitung der Themen erforderlich, da diese neben einer akademischen Ausbildung, Kenntnisse im Bereich RPAS, LiDAR Sensoren, Inertialnavigation, Sensorintegration sowie Kopplungsstrategien und -technologien haben mussten. Weiteres Personal wurde in Form von studentischen Hilfskräften im Rahmen von Programmier-, Mess- und Integrationsaufgaben sowie zur Durchführung von Feldversuchen über die komplette Laufzeit in dem Vorhaben eingesetzt. Darüber hinaus wurden Investitionen getätigt. Diese umfassten als größte Posten die Beschaffung eines Laserscanners vom Typ „Hokuyo UTM 30-LX“ als Ersatz für einen defekten Laserscanner, sowie Onboard Embedded Recheneinheiten für die unbemannten Robotersysteme

### **7.2 TU München**

Aus den bewilligten Mitteln wurden an der TU München im Wesentlichen wissenschaftliche Mitarbeiter beschäftigt. Diese waren zur Bearbeitung der Themen erforderlich, da diese neben einer akademischen Ausbildung, Kenntnisse im Bereich Multirotordynamik, nichtlinearer Regelung, adaptiver Regelung, Bild-verarbeitung, sowie Navigation aufweisen mussten. Weiteres Personal wurde in Form von studentischen Hilfskräften im Rahmen von Programmier-, Mess- und Integrationsaufgaben sowie zur Durchführung von Feldversuchen über die komplette Laufzeit im Vorhaben eingesetzt. Darüber hinaus wurden Investitionen getätigt. Diese umfassten als größte Posten die Beschaffung zweier Hexacopter UAV (Typ "Firefly" von Ascending Technologies GmbH), die Bodenfahrzeuge „Dr. Robot Jaguar“ als Versuchsträger sowie die für die Entwicklung und Untersuchung der beschriebenen Technologien benötigte Sensorik (z.B. Raytrix 3D Kamera). Außerdem wurden Inertialmesssensoren vom Typ „ADIS16488“ beschafft.

### **7.3 DLR Oberpfaffenhofen**

Aus den Mitteln des Unterauftrages wurden bei DLR-KN im Wesentlichen wissenschaftliche Mitarbeiter beschäftigt. Des Weiteren wurde Personal in Form von studentischen Abschlussarbeiten (Master/Diplomarbeiten) eingesetzt.

## **8 Notwendigkeit und Angemessenheit der geleisteten Arbeit**

### **8.1 TU Braunschweig**

Als Universitätsinstitut ist das IFF (TUBS) auf Förderungen im Rahmen von öffentlichen Förderprogrammen angewiesen, da eine Finanzierung von Forschungsvorhaben aus Eigenmitteln nicht möglich ist. Die Arbeiten wurden sorgsam geplant und im Sinne eines wirtschaftlichen Umgangs mit Fördermitteln bearbeitet.

Mit den in VaMEx-CoSMiC entwickelten Technologien wurde ein Beitrag erarbeitet, der den Einsatz von Schwärmen unbemannter Trägersysteme, z.B. für extraterrestrische Explorationsmissionen, ermöglicht. Dabei lassen sich Teile der entwickelten Technologien und Verfahren auch auf terrestrische Anwendungen, wie z.B. der Erkundung im Katastrophenfall mit einem Schwarm von unbemannten mobilen Trägersystemen, übertragen.

### **8.2 TU München**

Als Universitätsinstitute sind der Lehrstuhl für Flugsystemdynamik und der Lehrstuhl für Kommunikation und Navigation der TUM auf Förderungen im Rahmen von öffentlichen Förderprogrammen angewiesen, da eine Finanzierung von Forschungsvorhaben aus Eigenmitteln nicht möglich ist. Die Arbeiten wurden sorgsam geplant und im Sinne eines wirtschaftlichen Umgangs mit Fördermitteln bearbeitet.

Mit den in VaMEx entwickelten Technologien wurde ein Beitrag erarbeitet, der den Einsatz von Schwärmen unbemannter Trägersysteme, z.B. für extraterrestrische Explorationsmissionen, ermöglicht. Dabei lassen sich Teile der entwickelten Technologien und Verfahren auch auf terrestrische Anwendungen, wie z.B. der Erkundung im Katastrophenfall mit einem Schwarm von unbemannten mobilen Trägersystemen, übertragen.

### **8.3 DLR Oberpfaffenhofen**

Die Thematik der funkbasierten Schwarmnavigation und der Schwarmexploration, zusammen mit dezentralen Schätzalgorithmen wurden durch das Vorgängerprojekt VaMEx in DLR-KN verankert. Das Projekt VaMEx-CoSMiC und die Finanzierung u.a. durch den Unterauftrag erlaubten es DLR-KN die entsprechende Expertise als grundlegende Basis für weitere Arbeiten weiter zu entwickeln, sowie die Technologien in gemeinsamen Experimenten den Partnern zur Verfügung zu stellen.

Mit den in VaMEx-CoSMiC entwickelten Technologien wurde ein Beitrag erarbeitet, der den Einsatz von Schwärmen unbemannter Trägersysteme, z.B. für extraterrestrische Explorationsmissionen, ermöglicht. Dabei lassen sich Teile der entwickelten Technologien und Verfahren auch auf terrestrische Anwendungen, wie z.B. der Erkundung im Katastrophenfall mit einem Schwarm von unbemannten mobilen Trägersystemen, übertragen.

## **9 Voraussichtlicher Nutzen, Verwertbarkeit**

### **9.1 TU Braunschweig**

An der TU Braunschweig wurde das Vorhaben VaMEx-CoSMiC schon zur Laufzeit intensiv dazu genutzt über studentische Arbeiten, den Einsatz studentischer Hilfskräfte sowie der Einbindung der in VaMEx-CoSMiC erzielten Erkenntnisse in aktuelle Vorlesungen,

wissenschaftlichen Nachwuchs auszubilden und an aktuelle Forschungsthemen heranzuführen.

Darüber hinaus können und werden die Erkenntnisse im Rahmen anderer Forschungsaktivitäten und Forschungsvorhaben genutzt. Beispiele dafür sind die Übertragung von Erkenntnissen und Ergebnissen im Vorhaben „ANKommEn 2“ und in Projekten der VaMEx-Initiative, z.B. VIPE, LAOLa und VTB.

### **9.1.1 Erfindungen/Schutzrechtsanmeldungen**

Im Rahmen des Vorhabens sind bislang keine Erfindungs- oder Schutzrechtsanmeldungen erfolgt.

### **9.1.2 Wirtschaftliche Erfolgsaussichten**

Während der Projektlaufzeit wurden unter anderem eine vertragslose Kooperation mit der IAV GmbH eingegangen um ein Flugsystem für eine Anwendung im Innenbereich von Bürogebäuden zur Postzustellung zu entwickeln. Die Problemstellung der infrastrukturunabhängigen Ortung innerhalb eines Gebäudes wurde dazu in einer Masterarbeit untersucht. Weiterhin wurde in einer vertragslosen Kooperation mit der Asse GmbH der Versuch der automatisierten Erkundung und Kartierung von Stollen vorgenommen. Durch die starke Staubentwicklung, die durch die Abwinde des Fluggerätes hervorgerufen wurde und Messungen auf Basis eines Laserscanners nahezu unmöglich machte, wurde von weiteren Versuchen abgesehen und die Kooperation nicht weiter vertieft.

Über die bisherigen Kooperationen hinaus soll innerhalb eines Jahres nach Projektende durch eine Akquise von Projektpartnern eine Initiierung von Folgeaktivitäten zur gezielten Weiterentwicklung durchgeführt werden.

### **9.1.3 Wissenschaftlich/technische Erfolgsaussichten**

#### **Primärveröffentlichungen der TU Braunschweig**

- (1) M. Gäbel, T. Krüger, S. Nowak, J. Meifarth and U. Bestmann; "Multisensor Concept for Autonomous Navigation of Unmanned Systems in GNSS-denied Environments", ION GNSS+ 2017, Portland (Oregon), 2017

#### **geplante Veröffentlichungen der TU Braunschweig**

Schlussbericht "VaMEx CoSMiC - Valles Marineris Explorer Cooperative Swarm Navigation, Mission and Control" Teilprojekt TU Braunschweig, FKZ 50NA1520 Veröffentlichung in der TIB Hannover für März 2019 vorgesehen

#### **Abschlussarbeiten an der TU Braunschweig:**

##### **a) Bachelorarbeiten:**

- (1) „Konzept zur Gestaltung einer aufgabenorientierten Bodenkontrollstation für unbemannte Flugsysteme“
- (2) „Evaluierung eines Kommunikationssystems für Exploration mit unbemannten Robotern“
- (3) „Umsetzung einer Sensordatensimulation als virtuelle Testumgebung für eine Robotermission“
- (4) „Konzeptionierung und Untersuchung einer dynamischen Umfeldwahrnehmung für unbemannte Systeme unter Verwendung eines LiDARs“
- (5) „Sensoremulation zur Stützung eines UAS-Autopiloten in Experimentalsystemen“
- (6) „Umsetzung einer dreidimensionalen Echtzeit-Hindernisvermeidung für die Anwendung auf OctoMaps“
- (7) „Eine Explorationsstrategie für eine unbemannte Erkundung eines zuvor unbekanntes, dreidimensionalen Raumes“

#### **b) Studienarbeiten:**

- (1) „Konstruktion einer Vorrichtung zur 3D-Umfeldwahrnehmung mit einem LIDAR auf unbemannten Robotersystemen“
- (2) „Integration und Erprobung eines Autopiloten für einen unbemannten Fahrroboter“
- (3) „Transparentes Routen von MAVLink Nachrichten zur flexiblen Kontrolle eines Roboterschwarms“
- (4) „Zentralisierte Konflikterkennungsstrategie zwischen unbemannten Flugsystemen und bemannten Luftfahrzeugen im unteren Luftraum“
- (5) „Clearing Solar Panels of Dust using a Hexacopter“
- (6) „Automatische Hinderniserkennung für ein unbemanntes Flugsystem“
- (7) „Flugmechanische Modellierung von Multicopter-Systemen in MATLAB/Simulink“

#### **c) Masterarbeiten:**

- (1) „Simulation einer low-cost Sensorik für die autonome Navigation eines Outdoor-Lieferroboters“
- (2) „Innovative Reinigungskonzepte für fahrerlose Fahrzeuge basierend auf RPAS“
- (3) „Konzipierung und Implementierung eines Hindernisvermeidungssystems für unbemannte Flugsysteme“
- (4) „Evaluation of a Combined Visual-Sonic-Inertial Sensor System“
- (5) „Flight Control Design and Simulation of a Tandem Tilt Wing RPAS“
- (6) „Konzipierung und Implementierung eines optischen Ortungssystems zur Navigation von unbemannten autonomen Systemen“

#### **d) Projektarbeiten:**

- (1) „3D Navigation in GNSS-denied Environments for Unmanned Systems“

#### **Weitere Veröffentlichungen und Veranstaltungen:**

Zudem wurde das Projekt VaMEx-CoSMiC auf folgenden Veranstaltungen vertreten und präsentiert:

- ILA2016 in Berlin

### **9.1.4 Wissenschaftliche und wirtschaftliche Anschlussfähigkeit**

Die nächste Phase der technischen Entwicklungen wurde durch das Folgevorhaben „VaMEx-IRExA“ eingeleitet. Im Rahmen dieses Vorhabens sollen die Weiterentwicklungen hin zu einem hochautomatisierten und größtenteils autark operierenden Schwarms vorangetrieben werden. Im Fokus steht dabei die Entwicklung und Implementierung von Software zur intelligenten und robusten Schwarmsteuerung mit Mitteln des maschinellen Lernens und der künstlichen Intelligenz.

## **9.2 TU München**

An der TU München wurde das Vorhaben VaMEx-CoSMiC schon zur Laufzeit intensiv dazu genutzt über studentische Arbeiten, den Einsatz studentischer Hilfskräfte sowie der Einbindung der in VaMEx-CoSMiC erzielten Erkenntnisse in aktuelle Vorlesungen, wissenschaftlichen Nachwuchs auszubilden und an aktuelle Forschungsthemen heranzuführen. Beispiele hierfür sind die Vorlesungen „Nichtlinear Flugregelung“ und „Model Reference Adaptive Control“ (FSD) und „Visual Navigation“ (KN).

Darüber hinaus können und werden die Erkenntnisse im Rahmen anderer Forschungsaktivitäten und Forschungsvorhaben genutzt. Ein Beispiel dafür ist die generelle Übertragung von Erkenntnissen, Ergebnissen und Tools im Bereich ausfallsicherer

Multirotorsysteme auf die Forschungsvorhaben innerhalb der Multirotorgruppe (zB. DFG Projekt AdapNMPC und das Nachfolgeprojekt VaMEx - Cosmic). Erkenntnisse zur Ausfallsicherheit von Multirotorsystemen dienten auch als Input für Institutionen, die Richtlinien zur UAV Nutzung in Deutschland entwickeln. Des Weiteren werden die Ergebnisse aus VaMEx für den Aufbau einer Forschungsgruppe zur visuellen Navigation genutzt. Weiterhin fließen die generierten Erkenntnisse in das Habilitationsvorhaben von Dr. Gabriele Giorgi ein.

## 9.2.1 Lehrstuhl für Kommunikation und Navigation

### 9.2.1.1 Wissenschaftlich/technische Erfolgsaussichten

#### Primärveröffentlichungen:

- (1) Zhu, C., Giorgi, G. and Günther, C., 2018. 2D Relative Pose and Scale Estimation with Monocular Cameras and Ranging. *Navigation: Journal of The Institute of Navigation*, 65(1), pp.25-33.
- (2) Zhu, C., Giorgi, G., Lee, Y.H. and Günther, C., 2018, April. Enhancing accuracy in visual SLAM by tightly coupling sparse ranging measurements between two rovers. In *Position, Location and Navigation Symposium (PLANS), 2018 IEEE/ION* (pp. 440-446). IEEE.
- (3) Lee, Y.H., Zhu, C., Giorgi, G. and Guenther, C., 2018, April. Stereo vision-based simultaneous localization and mapping with ranging aid. In *Position, Location and Navigation Symposium (PLANS), 2018 IEEE/ION* (pp. 404-409). IEEE.
- (4) Zhu, C., Giorgi, G. and Günther, C., 2017. Planar Pose Estimation using a Camera and Single-Station Ranging Measurements. In *Proceedings ION GNSS+ 2017*, Portland, Oregon.
- (5) Zhu, C., Giorgi, G. and Günther, C., 2016. Scale and 2D Relative Pose Estimation of two Rovers using Monocular Cameras and Range Measurements. In *Proceedings of the 29th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+ 2016)*, Portland, Oregon. Institute of Navigation (pp. 794-800).

#### Wissenschaftliche Besuche:

- (1) Chen Zhu, Fusion of Cameras and Sparse Ranging Measurements in Multi-agent SLAM, Carnegie Mellon University, Pittsburgh, USA, Sep. 2017

#### Bachelorarbeiten:

- (1) Christian Werner, Implementation of a Vision-based SLAM with Inertial Aid, 2015

#### Masterarbeiten:

- (1) Chiraz Nafouki, Visual SLAM with Ranging Aid, 2016

#### Forschungspraxis:

- (1) Adrian Belli, Indoor ground truth system development with UWB ranging sensors, Apr. - Aug. 2018
- (2) Boxin Zhang, Scale Estimation of Two Cooperative Rovers Using Monocular Cameras and Sparse Range Measurements - A Real-time Implementation, -Aug. 2018

#### **Werkstudenten:**

- (1) Aayush Singla, Multi-sensor data synchronization and testbed development, Mär.-Aug., 2018
- (2) Guillermo Alejandro Barraza Montiel, Structure design and manufacture for the system integration with DLR-KN, Mär.-Aug., 2018
- (3) Ghaith Allah Chebil, RTK data reception and a testbed development, Okt.-Dez., 2017
- (4) Xueyang Kang, Data synchronization of stereo images and UWB ranging measurements, 2017
- (5) Zhiwei Han, Implementation of Visual SLAM with a Raspberry Pi, 2017

#### **Vorlesung:**

- (1) Visual Navigation, Dr. Gabriele Giorgi, WiSe

#### **Events:**

- (1) ILA-Berlin: gemeinsamer VaMEx-Stand der Initiative, Juni 2016

## **9.2.2 Lehrstuhl für Flugsystemdynamik**

### **9.2.2.1 Erfindungen/Schutzrechtsanmeldungen**

Im Rahmen des Vorhabens sind bislang keine Erfindungs- oder Schutzrechtsanmeldungen erfolgt.

### **9.2.2.2 Wirtschaftliche Erfolgsaussichten**

Die Erkenntnisse des Vorhabens werden bereits und sollen in Zukunft in verschiedenen Industrieprojekten eingesetzt werden. Hier ist beispielsweise das Industrieprojekt "5TOL" des Lehrstuhls für Flugsystemdynamik zu nennen, bei welchem ein Transitionsfluggerät mit der Partnerfirma Autel Europe GmbH entwickelt wird. Zusätzlich fließen die Erkenntnisse auch die Entwicklung von bemannten Systemen (Projekt AFX V600 mit AutoFlightX) mit ein und bringen hier einen erheblichen Sicherheitsgewinn.

### **9.2.2.3 Wissenschaftlich/technische Erfolgsaussichten**

Die Ergebnisse im Bereich des Flugvehikelentwurfs für die Marsatmosphäre und der Simulation in Marsumgebung sollen in einem Nachfolge-Projekt zur Entwicklung eines Flugvehikels gemeinsam mit dem Lehrstuhl für Hubschraubertechnologie angewendet werden. Der Antrag für dieses Projekt wird in Kürze gestellt.

#### **Primärveröffentlichungen:**

- (1) G. P. Falconí, F. Holzapfel, "Position tracking of a hexacopter using a geometric backstepping control law - Experimental results", in *Aerospace Electronics and Remote Sensing Technology (ICARES), 2014 IEEE International*, pp.20-25.
- (2) G. P. Falconí, F. Holzapfel, "Position Tracking of a Multicopter using a Geometric Backstepping Control Law", in *CEAS EuroGNC, 2013*
- (3) G. P. Falconí, V. Marvakov and F. Holzapfel, "Fault Tolerant Control for a Hexarotor System Using Incremental Backstepping", in *IEEE Multi-Conference on Systems and Control (CCA), 2016*, pp. 237-242.
- (4) G. P. Falconí, J. Angelov and F. Holzapfel, "Hexacopter Outdoor Flight Test Results

Using Adaptive Control Allocation Subject to an Unknown Complete Loss of One Propeller", in *3rd Conference on Control and Fault-Tolerant Systems (SysTol)*, 2016, pp. 367-374

- (5) G. P. Falconí, J. Angelov and F. Holzapfel, „Adaptive Fault-Tolerant Position Control of a Hexacopter Subject to an Unknown Motor Failure." *Int. J. Appl. Math. Comput. Sci* 28.2 (2018): 309-321.

### **Sekundärveröffentlichungen:**

- (1) Mühlegg, M., Niermeyer, P., Falconi, G. P., & Holzapfel, F. (2015, September). "L1 fault tolerant adaptive control of a hexacopter with control degradation". In *Control Applications (CCA), 2015 IEEE Conference on* (pp. 750-755). IEEE.
- (2) Mühlegg, M., Raffler, T., & Holzapfel, F. (2016, November). "Quadcopter flight test results of a consistency monitoring algorithm for adaptive controllers". In *Control, Automation, Robotics and Vision (ICARCV), 2016 14th International Conference on* (pp. 1-6). IEEE.

### **Abschlussarbeiten:**

#### **a) Studienarbeiten:**

- (1) „Inkrementelle Regelung für ein Multirotor-System“ (Valentin Marvakov)
- (2) „Methoden zur Allokation von Steuergrößen für ein Hexakopter System“ (Jorg Angelov)
- (3) „Adaptives Backstepping für Positionsfolgeregelung eines Hexakopters“ (Christoph Kramer)
- (4) „Entwicklung eines autonomen Luftschiffs für eine Marsmission“ (Henrique Zingg)
- (5) „Design of a MATLABSIMULINK based simulation model for a flying vehicle on Mars“ (Daniel Bodmer)

#### **b) Masterarbeiten:**

- (1) „Online Flight Path Optimization on a Hexarotor UAV Using IPOPT“ (Florian Schwaiger)
- (2) „Design of a UAV Concept for the Mars Valles Marineris Exploration“ (Philipp Reiß)
- (3) „Design and Augmentation of a Nonlinear Backstepping-Based Position Tracking Controller for a Rotary-Wing Mars Vehicle (Daniel Bodmer)
- (4) „Dynamik und Regelremulation für ein Marsvehikel“ (Sascha Rimet)

### **Weitere Veröffentlichungen und Veranstaltungen**

Zudem wurde das Projekt VaMEx-CoSMiC auf folgenden Veranstaltungen vertreten und präsentiert:

- Doktorandendialog zwischen TU Berlin, TU Hamburg-Harburg und TUM
- Synergieworkshops der VaMEx-Initiativen

## **9.3 DLR Oberpfaffenhofen**

### **9.3.1 Erfindungen/Schutzrechtsanmeldungen**

Zu Schutzrechtsanmeldungen kam es in diesem Vorhaben nicht.

### **9.3.2 Wirtschaftliche Erfolgsaussichten**

Die im Vorhaben entwickelten Algorithmen und Experimentalaufbauten stellen einen proof-

of-concept dar. Die entwickelten Verfahren haben einen starken Fokus im Raumfahrtbereich um eine Kommunikation, Lokalisierung, und Zeitsynchronisation von Robotern und Sensoren für die extraterrestrische Exploration zu ermöglichen. Eine Weiterentwicklung des Experimentalaufbaus als Raumfahrtkomponente könnte zu einem Einsatz sowohl in der kommerziellen Raumfahrt, als auch in der Raumfahrtforschung führen.

### 9.3.3 Wissenschaftlich/technische Erfolgsaussichten

Die wissenschaftliche und technische Verwertung der Ergebnisse erfolgt über Veröffentlichungen bei internationalen Konferenzen und Journalen, Messeauftritte und Workshops, sowie über die Förderung des wissenschaftlichen Nachwuchses durch Abschlussarbeiten, Praktika, und Vorlesungen.

#### Primärveröffentlichungen:

- (1) Staudinger, Emanuel und Shutin, Dmitriy und Manss, Christoph und Viseras, Alberto und Zhang, Siwei. *Swarm Technologies For Future Space Exploration Missions*. In: ISAIRAS &apos;18: 14th International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-sairas), 2018, Madrid, Spanien.

#### TPC-Member bei internationalen Konferenzen:

- IEEE Workshop on Positioning, Navigation and Communication (WPNC, 2015),
- IEEE International Conference on Wireless for Space and Extreme Environments (WiSEE, 2015)
- IEEE Globecom - 9th International Workshop on Wireless Networking and Control for Unmanned Autonomous Vehicles (Wi-UAV, 2018)

#### Wissenschaftliche Besuche:

- (1) Im Juni 2016 besuchte Herr. Zhang den Prof. Henk Wymeersch an der Chalmers University of Technology in Schweden für einen wissenschaftlichen Austausch für eine Dauer von zwei Monaten.
- (2) Im November 2017 besuchte Hr. Wiedemann das Centre for Applied Autonomous Sensors Systems (AASS) der Örebro Universität in Schweden.

#### Masterarbeiten:

- (1) Juan Marchal Gomez: Dropped Sensor Network
- (2) Wu Tsung-Huan: Distributed Localization Based on Reference Frame Consensus for Extra-Terrestrial Swarm Navigation

#### Bachelorarbeiten:

- (1) Lukas Magel: Position-Based Routing in Dynamic Mesh Networks for Applications with Micro-UAVs

#### Praktika/Werkstudenten:

Während des Vorhabens wurden sechs Studenten im Rahmen von Praktika und Werkstudententätigkeiten gefördert.

#### Vorlesung:

Die theoretischen Grundlagen zur Funklokalisierung und Schwarmnavigation werden als Vorlesung "Robot and Swarm Navigation" am Lehrstuhl für Kommunikation und Navigation der TU München gehalten, in welcher die Grundlagen zur Funklokalisierung und Schwarmnavigation den Studenten beigebracht wird.

#### Events:

Neben den Veröffentlichungen präsentiert DLR-KN die Arbeiten zur Schwarmnavigation und Schwarmexploration der Öffentlichkeit bei diversen Workshops, Messeauftritten und Delegationsbegleitungen. Das Zielpublikum reicht von allgemein Interessierten der Öffentlichkeit bis hin zu Entscheidungsträgern bei der NASA. Nachfolgend sind diese Events chronologisch sortiert aufgelistet:

Okt. 2015	Institutstag mit Besuch von Prof. Dittus und Dr. Reile
Dez. 2015	DLR-Delegation besucht NASA AMES und NASA Headquarters in Washington DC
Feb. 2016	DLR besucht die NASA JPL-Kollegen Dr. Zmuidzinas und Dr. Fred. Hadaegh
Juni 2016	ILA-Berlin: gemeinsamer VaMEx-Stand der Initiative
Juni 2016	50-Jahre DLR-KN Feier und Institutstag
Sep. 2016	DLR-KN Messestand auf der ION-GNSS in Portland, USA
Sep. 2016	DLR-Delegation besucht NASA JPL und NASA Headquarters
Okt. 2016	Vortrag auf dem ROBEX Meeting an der TU-Kaiserslautern
Feb. 2017	JAXA-Delegation besucht das DLR
Mai 2017	Vortrag am ESA Mars Exploration Day in Noordwijk
Juni 2017	Vortrag im IGDK Optimal Design Workshop
Dez. 2017	Besuch von der KI-Abteilung des DFKI
März 2018	HGF-Überprüfung mit Besuch externer Gutachter
Sept. 2018	Messestand auf dem IAC in Bremen

Im Frühjahr 2017 besuchte ein Filmteam DLR-KN für einen Beitrag bei ARD alpha "Schwarmverhalten: Die Intelligenz der Vielen" sowie Planet Wissen "Schwarmverhalten – die Intelligenz der Vielen". Diese Beiträge wurden Mitte/Ende 2017 ausgestrahlt und schlossen auch einen Studiobeitrag durch Prof. Fiebig (Leiter der Abteilung Kommunikationssysteme) mit ein.

Die im Rahmen von VaMEx-CoSMiC erarbeiteten Algorithmen und aufgezeichneten Rohdaten aus den Experimenten stehen auch nach Projektende zur Verfügung. Die Datensätze aus den Experimenten erzeugen einen signifikanten Mehrwert, da Algorithmen weit über das Projektende hinaus noch mit diesen Daten evaluiert werden können. Diese Ergebnisse fließen zudem in aktuelle, laufende Projektvorhaben mit ein. Die theoretischen Grundlagen sind zudem beliebig anwendbar und nutzbar für zukünftige Vorhaben. Die von DLR-KN entwickelten Verfahren sind nicht nur im Raumfahrtbereich sondern auch im terrestrischen Bereich einsetzbar, z.B. zur Lokalisierung/Navigation von automatisierten Fahrzeugen, und Personen: daher erfolgt ein Transfer in laufende Vorhaben und eine Nutzung für zukünftige Vorhaben.

### 9.3.4 Wissenschaftliche und wirtschaftliche Anschlussfähigkeit

Die entwickelten Verfahren sind generisch und können für verschiedene Anwendungsbereiche adaptiert werden. Daher erfolgt die Weiterentwicklung der Verfahren in diversen Forschungsprojekten mit unterschiedlicher Zielanwendung. Bereits während der Laufzeit des Vorhabens erfolgte ein Transfer zwischen VaMEx-CoSMiC und weiteren Projekten, welche nachfolgend aufgelistet sind.

- Explorer-Initiativen des DLR Raumfahrtmanagements: Die Synergieworkshops der Explorer-Initiativen wurden intensiv zum Austausch genutzt und werden auch zukünftig besucht.
- EU-H2020 Projekt HIGHTS - High precision positioning for cooperative-ITS (2015-2018)
- EU-H2020 Projekt HEIMDALL – Multi-Hazard Cooperative Management Tool for Data Exchange, Response Planning and Scenario Building (2017-2020)
- DFG-Projekt: ModeNAV: Nutzung von Multimodenantennen zur Schwarmlokalisierung und Lagebestimmung der Agenten (2017-2019)
- VaMEx-VTB (2018): Beitrag der Schwarmexploration und Schwarmnavigation zum virtuellen Testbed
- Helmholtz Zukunftsthema ARCHES (2018-2020): kooperierende Roboter in der Raumfahrt und Tiefsee.

Neben den laufenden Projekten wurde ein Förderantrag im Rahmen der Deutsch-Israelischen Zusammenarbeit eingereicht um zu evaluieren, in welchem Umfang die entwickelten Verfahren neben der Raumfahrt domäne, auch in der Tiefseedomäne eingesetzt werden. Der Antrag befindet sich aktuell in der zweiten Begutachtungsstufe und soll 2019 starten.

## **10 Bekanntgewordener Fortschritt anderer Stellen**

### **10.1 DLR Explorer-Initiative**

Dieses Projekt wurde im Rahmen der VaMEx-Initiative durchgeführt zu der auch die parallel durchgeführten Vorhaben VaMEx-VTB (FKZ 50NA1517-12), VaMEx-VIPE (FKZ 50NA1515-17) und VaMEx-LAOLa (FKZ 50NA1527-29) gehören.

Das Projekt VaMEx-VTB besteht aus zwei Teilen: das virtuelle Testbett, welches eine virtuelle Testumgebung für die in den VaMEx-Initiativen entwickelten Systemen bereitstellt, und eine Orbiter-Komponente, die für diese Systeme eine Absolut-Positionierung mittels eines oder mehrerer Orbiter bereitstellt.

Das LAOLa-Projekt beschäftigt sich mit der Möglichkeit eine Schar von Drohnen, Rover und Laufrobotern im lokalen System des Mars' zu Orten sowie die Landung aus dem Orbit mit Positionsdaten zu erfassen. Es handelt sich um eine Vorstudie, die sich mit den Grundlagen eines solchen Vorhabens befasst.

In dem abgeschlossenen Verbundvorhaben VIPE entwickelte der Verbund den vierbeinigen hominiden DFKI-Laufroboter „Charlie“, weiter, um diese verbleibende Lücke im Schwarm zu schließen. Zudem wurden im Projekt innovative Ansätze der vollautonomen Positionierung und Kartierung entwickelt, die Charlie eine besonders driftarme Positionsbestimmung auch unter komplexen Bedingungen ermöglichen.

Die VaMEx-Initiative ist ein Bestandteil der DLR Explorer-Initiativen, deren Ziel die Technologieentwicklung für zukünftige Raumfahrtmissionen darstellt.

### **10.2 TU Braunschweig**

Fortschritt anderer Stellen konnte auf den im Rahmen des Vorhabens besuchten Fachkonferenz ausgetauscht und diskutiert werden. Schwerpunkte dabei waren in erster Linie Entwicklungen im Bereich der automatisierten und kooperativen Schwarmnavigation und dreidimensionalem LiDAR SLAM, die, wenn möglich, bereits in den Arbeiten und Ergebnissen berücksichtigt wurden.

### **10.3 TU München**

Fortschritt anderer Stellen konnte auf den im Rahmen des Vorhabens besuchten Fachkonferenzen ausgetauscht und diskutiert werden. Schwerpunkte dabei waren in erster Linie Entwicklungen im Bereich Bilderkennung, Navigation und ausfallsicherer Regelung für unbemannte Flugsysteme, die, wenn möglich, bereits in den Arbeiten und Ergebnissen berücksichtigt wurden.

### **10.4 DLR Oberpfaffenhofen**

Die Allgemeine Thematik der kooperativen Schwarmpositionierung findet auch in anderen Bereichen Anwendung. Fortschritte an anderen Stellen wurden über entsprechende internationale Konferenzen und Workshops, sowie die stets fortwährende Literaturrecherche für die wissenschaftliche Arbeit mit berücksichtigt. Dabei zeigte sich u.a. dass ein Bedarf an robusten Kommunikations- und Navigationslösungen im New-Space auch für kommerzielle Anbieter von Interesse ist.

## 11 Zusammenfassung

Diese Arbeit wurde im Rahmen des Projektes VaMEx-CoSMiC (Valles Marineris Explorer-Cooperative Swarm Navigation, Mission and Control) angefertigt. Die im Rahmen des Vorhabens VaMEx-CoSMiC bearbeiteten Arbeitspakete wurden erfolgreich durchgeführt und abgeschlossen. Das Projekt wurde gefördert durch das Bundesministerium für Wirtschaft und Energie, verwaltet vom DLR Raumfahrtmanagement (FKZ 50 NA 1520, FKZ 50 NA 1521).

Gefördert durch:



aufgrund eines Beschlusses  
des Deutschen Bundestages