



Virtual Crater - Entwicklung einer virtuellen Simulations- und Demonstrationsumgebung zur planetarischen Exploration mit Fokus auf extraterrestrische Krater

DFKI Robotics Innovation Center

6. November 2012

Das diesem Bericht zugrunde liegende Vorhaben wurde von der Raumfahrt-Agentur des Deutschen Zentrums für Luft- und Raumfahrt e.V. mit Mitteln des Bundesministeriums für Wirtschaft und Technologie aufgrund eines Beschlusses des Deutschen Bundestages unter dem Förderkennzeichen 50RA0903 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.

Deutsches Forschungszentrum für künstliche Intelligenz
Robotics Innovation Center
Prof. Dr. Kirchner
Robert-Hooke-Str. 5
28359 Bremen, Germany



Inhaltsverzeichnis

1	Einleitung	4
2	Kurzdarstellung nach NKBF 98 8.2 I	5
2.1	Aufgabenstellung	5
2.2	Voraussetzungen, unter denen das Vorhaben durchgeführt wurde	6
2.3	Planung und Ablauf des Vorhabens	6
2.4	Wissenschaftlicher und technischer Stand zu Beginn des Vorhabens	9
2.4.1	Stand der Technik zu Beginn des Vorhabens	9
2.4.2	Fachliteratur und verwendete Dokumentationsdienste	12
2.5	Zusammenarbeit mit anderen Stellen	12
3	Eingehende Darstellung	13
3.1	Erzielte Ergebnisse	13
3.1.1	AP1000: Management	13
3.1.2	AP2000: Konzeption, Anforderungsanalyse	16
3.1.3	AP2100: Analyse der Anforderungen an die Benutzerschnittstellen	16
3.1.4	AP2200: Analyse der Anforderungen an Schnittstellen zwischen realer Hardware und Simulationssystem	18
3.1.5	AP2300: Anforderungsanalyse Simulationssystem	19
3.1.6	AP2400: Anforderungsanalyse Referenzexperimente	22
3.1.7	AP3000: Definitions- und Entwicklungsphase	30
3.1.8	AP3100: Definition und Spezifikation der Benutzerschnittstelle zwischen Simulationssystem und Simulationsanwender	30
3.1.9	AP3110: Visualisierung / Bedienung / Interaktion	30
3.1.10	AP3120: Programmiermodell für Erweiterungen	33
3.1.11	AP3200: Definition und Spezifikation der Schnittstellen zwischen realer Steuerung und Simulation	33
3.1.12	AP3300: Definition anwendungsspezifischer Erweiterungen des Simulationssystems	40
3.1.13	AP3310: Darstellung	40
3.1.14	AP3320: Dynamiksimulation	46
3.1.15	AP3330: Sensorsimulation	47
3.1.16	AP3340: Simulation der Antriebe	52
3.1.17	AP3350: Bodenmechanik	57
3.1.18	AP3360: CAD-Import	63
3.1.19	AP3400: Definition Referenzexperimente	63
3.1.20	AP3500: Entwurf von Verfahren zur Auswertung von Simulationsergebnissen hinsichtlich der Optimierung des Designs	72
3.1.21	AP4000: Implementierungsphase	75
3.1.22	AP4100: Benutzerschnittstelle Simulationssystem	75
3.1.23	AP4110: Visualisierung / Bedienung / Interaktion	75
3.1.24	AP4120: Entwicklungsumgebung für Erweiterungen	76
3.1.25	AP4200: Implementierung der Schnittstellen zwischen realer Steuerung und Simulationssystem	76



3.1.26 AP4300: Projektspezifische Erweiterungen am Simulationssystem	76
3.1.27 AP4310: Darstellung	76
3.1.28 AP4320: Dynamik	77
3.1.29 AP4330: Sensorik	77
3.1.30 AP4340: Antriebe	78
3.1.31 AP4350: Bodenmechanik	81
3.1.32 AP4360: DFKI-CAD-Modellimport	91
3.1.33 AP4400: Modellierung / Aufbau Referenz-experimente (real & virtuell)	91
3.1.34 AP4410: Aufbau der Referenzexperimente im realen Testbed	91
3.1.35 AP4420: Modellierung der Referenzexperimente im virtuellen Testbed	103
3.1.36 AP4500: Aufbau einer Simulations-/Projektionsumgebung im DFKI	107
3.1.37 AP4600: Einarbeitung in das und Schulung auf dem Simulationssystem	111
3.1.38 AP5000: Transfer, Systemtests, Verifikation	111
3.1.39 AP5100: Systemtest und Anpassungen der Benutzerschnittstellen des Simulationssystems	111
3.1.40 AP5200: Systemtest und Anpassungen der Schnittstellen zwischen Steuerungshardware und Simulationssystem	111
3.1.41 AP5300: Anpassung der Simulationsmodelle an reale Ergebnisse	113
3.1.42 AP5400: Durchführung der Referenzexperimente	138
3.1.43 AP5410: Durchführung im realen Testbed	138
3.1.44 AP5420: Durchführung im virtuellen Testbed	164
3.1.45 AP5430: Vergleich virtueller und realer Ergebnisse	171
3.1.46 AP6000: Einsatz und Extrapolation	183
3.1.47 AP6100: Definition einer Referenz-Mondmission	183
3.1.48 AP6200: Modellierung and Implementierung einer virtuellen Referenz- Mondmission	186
3.1.49 AP6300: Demonstration Referenz-Mondmission	190
3.1.50 AP6400: Auswertung der Ergebnisse der virtuellen Referenzmission	190
3.2 Die wichtigsten Positionen des zahlenmäßigen Nachweises	197
3.3 Nutzen und Verwertungsplan	197
3.4 Forschungsergebnisse Dritter	199
3.5 Veröffentlichungen	203



1 Einleitung

Das vorliegende Dokument stellt den Abschlussbericht des Projektes "Virtual Crater - Entwicklung einer virtuellen Simulations- und Demonstrationsumgebung zur planetarischen Exploration mit Fokus auf extraterrestrische Krater" dar. Virtual Crater wurde von der Raumfahrt-Agentur des Deutschen Zentrums für Luft- und Raumfahrt e.V. mit Mitteln des Bundesministeriums für Wirtschaft und Technologie aufgrund eines Beschlusses des Deutschen Bundestages unter dem Förderkennzeichen 50RA0903 gefördert.

Dieser Abschlussbericht folgt dabei dem in der Anlage 2 zum NKBF 98 gegebenen Muster, in Kapitel 2 wird eine kurze Darstellung des Projektes gemäß Punkt I gegeben. Kapitel 3 stellt die inhaltlichen Entwicklungen in einer eingehenden Darstellung gemäß Punkt II des Musters dar. Die Punkte III (Erfolgskontrollbericht) und IV (Berichtsblatt bzw. Document Control Sheet) werden durch gesondert abgegebene Dokumente erfüllt.



2 Kurzdarstellung nach NKBF 98 8.2 I

In diesem Kapitel wird zunächst die Aufgabenstellung des Projektes beschrieben (Abschnitt 2.1). Anschließend werden die Voraussetzungen des Vorhabens (Abschnitt 2.2) sowie Planung und Ablauf des Vorhabens (Abschnitt 2.3) behandelt. Es folgen Erläuterungen zum wissenschaftlichen Stand (Abschnitt 2.4) und zur Zusammenarbeit mit anderen Stellen (Abschnitt 2.5).

2.1 Aufgabenstellung

Das Ziel des Projekts "Virtual Crater" war die Entwicklung einer virtuellen Testumgebung, die es ermöglicht, Robotersysteme kostengünstig in einer realitätsnah simulierten, lunaren Kraterlandschaft zu programmieren, zu testen und zu optimieren. Eine Besonderheit dieses Projekts ist der Abgleich der virtuellen Testumgebung mit realen Referenzexperimenten zu Systemen und Systemkomponenten. In diesem Rahmen arbeiteten das Deutsche Forschungszentrum für Künstliche Intelligenz (DFKI GmbH) und die Dortmunder Initiative zur rechnerintegrierten Fertigung (RIF) e.V. als Partner zusammen.

Im Projekt Virtual Crater sollte eine umfassende Simulationsumgebung entwickelt werden, die es ermöglicht, Missionen zur Erforschung der Mondoberfläche zu programmieren und zu testen sowie neue Konzepte vorzuführen. Damit sich diese Testumgebung so realitätsnah wie möglich verhält, sollten verschiedene Parameter und Prozesse präzise identifiziert werden, um vielseitige Simulationskomponenten daraus abzuleiten. Um dies zu erreichen, sollten umfassende physikalische Experimente durchgeführt werden, um diese mit analogen simulierten Experimenten zu vergleichen. Diese Experimente werden Referenzexperimente genannt. Die Simulation wiederum soll - zusammen mit speziellen Optimierungswerkzeugen - verwendet werden, um Hardware zu optimieren, Explorations-Szenarien aufzubauen und entsprechende Missionen zu simulieren. Weiterhin war ein wichtiges Ziel, die Robotersteuerung ohne Anpassungen sowohl an das Simulationssystem als auch an das reale Testbed anbinden zu können, so dass der virtuelle Roboter wie der reale programmiert und insbesondere durch denselben Programmcode gesteuert wird. Um eine Verbesserung des Eindrucks der „Immersion“ in die virtuelle Welt zu erreichen, wird das Simulationssystem auf einem stereoskopischen Mehrschirm-Rückprojektionssystem (CAVE) sowie auf einem normalen Arbeitsplatzrechner als autarke Entwicklungsumgebung eingesetzt.

Wichtig waren weiterhin Echtzeitfähigkeiten, um die Konstruktion der Systeme intuitiv zu unterstützen. Ein bedeutsames Ziel waren neben echtzeitfähigen Sensor- und Aktuatormodellen auch echtzeitfähige und dennoch aussagekräftige Boden-Kontaktmodelle, um das Verhalten eines Laufroboters in extraterrestrischen Kraterumgebungen auch auf weichen Untergründen wie z.B. dem Regolith auf dem Mond simulieren zu können.



2.2 Voraussetzungen, unter denen das Vorhaben durchgeführt wurde

Das Vorhaben wurde durch das Deutsche Forschungszentrum für Künstliche Intelligenz (DFKI) und dort im Robotics Innovation Center (RIC) am DFKI Standort Bremen in Zusammenarbeit mit der Dortmunder Initiative zur rechnerintegrierten Fertigung (RIF) e.V. durchgeführt.

Unter der Leitung von Prof. Dr. Frank Kirchner verfügt das RIC über langjährige Erfahrung in der Entwicklung vollständiger Steuerungselektronik und Mechanikkomponenten, sodass für jedes Projekt optimierte Elektronik und Mechanik entsprechend der identifizierten Anforderungen erstellt werden kann. Ebenso verfügt das RIC über weitreichende Erfahrungen in der high- und low-level Kontrolle von Robotersystemen. Lernverfahren, sowie simulationsbasierte Evolutionsstrategien und genetische Algorithmen wurden am RIC ebenfalls z.B. im Projekt SpaceClimber eingesetzt, um die mechanische Struktur und auch das Laufverhalten des Roboters zu entwickeln. Für die Entwicklung von Robotersystemen verfügt das RIC auch über langjährige Erfahrungen auf dem Gebiet der Simulation, sowohl im nicht echtzeitfähigen hoch präzisen Bereich als auch mit echtzeitfähigen interaktiven Simulationen.

Somit verfügte das RIC über das nötige Know-How, um die bereits zu Projektbeginn am RIC vorhandenen Simulationstechnologien für VEROSIM® weiterzuentwickeln, so dass realitätsnahe Modelle für physikalische und mechanische Komponenten wie Aktuatoren und die Bein-Boden-Interaktion des Roboters sowie durch die Kooperation mit MMI und RIF eine immersive Kontrolle von Simulation und simuliertem Roboter zur Verfügung stehen.

2.3 Planung und Ablauf des Vorhabens

Das Projekt Virtual Crater ist ursprünglich mit einer Laufzeit von 36 Monaten geplant worden. Der Projektbeginn war der 01.05.2009.

Wie im Virtual Crater Antrag (Kapitel 6.2.1) festgelegt, fungierte das MMI (Institut für Mensch-Maschine-Interaktion RWTH AC) als ein Unterauftragsnehmer vom RIF im Rahmen eines F+E Auftrages. Aus diesem Grund kooperierte das DFKI in inhaltlichen Aufgaben mit dem MMI und in administrativen Angelegenheiten mit der RIF.

Am 01.10.2009 fand ein Review-Meeting mit den Teilnehmern DFKI, MMI/RIF, und DLR statt. In diesem Meeting haben DFKI und MMI/RIF zusammen in einer Präsentation dem DLR die Arbeitsergebnisse und die weitere Arbeitsplanung vorgestellt und mit dem DLR diskutiert.

Nach diesem Review-Meeting hat das DLR dem DFKI und MMI/RIF mitgeteilt, dass die Quartalsberichte dem jeweiligen fachlichen Bearbeiter der Agentur übermittelt werden sollen. DFKI und MMI/RIF haben vereinbart, ab dem vierten Quartal einen gemeinsamen Quartalsbericht zu schreiben.

Im vierten Quartal wurde eine zweitägige Einführung in VEROSIM® am MMI in Aachen durchgeführt. Teilgenommen haben an dieser Einführung, die vom MMI gegeben wurde, zwei Mitarbeiter vom RIC Bremen (Michael Rohn und Yong-Ho Yoo). Inhalt der



Einführung waren das grundsätzliche Modellieren in VEROSIM[®] und das Schreiben von Plug-Ins für VEROSIM[®]. Über diese Plug-Ins kann das DFKI die Simulation erweitern und an die vorhandenen Tools wie Matlab oder Adams anbinden. Im Rahmen dieser Einführung hat das DFKI drei Lizenzen für VEROSIM[®] in der Form von USB-Hardware-Dongles erhalten, so dass auch im DFKI an der Umsetzung in VEROSIM[®] gearbeitet werden konnte.

Am 24. März 2011 fand ein Meilenstein-Treffen mit dem DLR und den Partnern RIF und MMI in Bremen statt, auf dem der Projektstand vorgestellt und demonstriert wurde.

Bei der Auswertung von Ergebnissen aus den Referenzexperimenten und deren Berücksichtigung in den Simulationsmodellen ist es durch die Komplexität des physikalischen Verhaltens und die dadurch erhöhte Bandbreite von erforderlichen Experimenten zu Verzögerungen im Projekt gekommen. Damit die Verzögerung aufgeholt und das Projekt mit erweiterten qualitativ höherwertigen Ergebnissen erfolgreich abgeschlossen werden konnte, wurde eine kostenneutrale Verlängerung des Projekts bis zum 31.08.2012 beantragt und bewilligt. Die Dortmunder Initiative zur rechnerintegrierten Fertigung (RIF) e.V. stimmte auch für diese Verlängerung und hat diese ebenfalls für sich beantragt und auch diese wurde bewilligt.

Infolgedessen wurden in 2012 auch verstärkt neue Referenzexperimente durchgeführt und für die Aktuatorsimulation wurde auch ein neues unterstützendes präzises Simulationsmodell entwickelt. Ferner wurde der Projektplan für die verbleibenden Arbeitspakete entsprechend angepasst.

Abb. 1 zeigt die endgültige zeitliche Planung der Arbeitspakete.

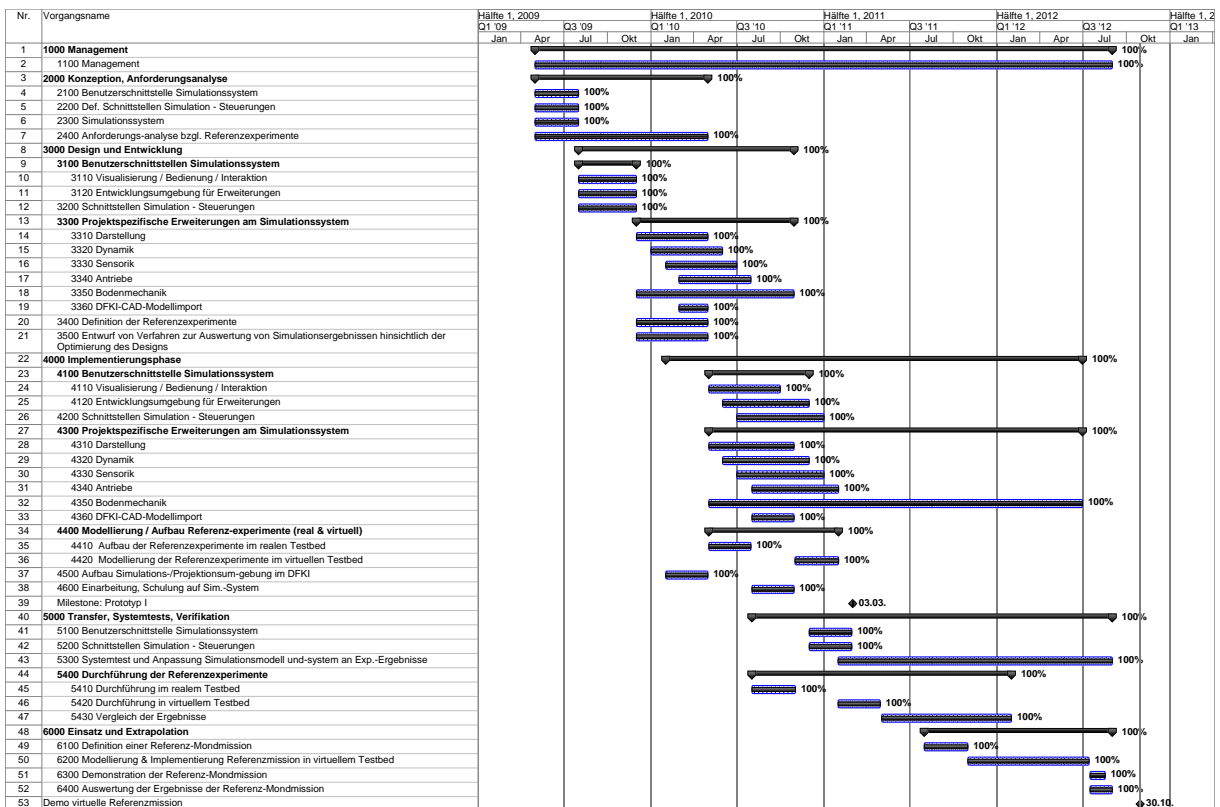


Abbildung 1: Gantt Ablaufplan - endgültige Version



2.4 Wissenschaftlicher und technischer Stand zu Beginn des Vorhabens

2.4.1 Stand der Technik zu Beginn des Vorhabens

2.4.1.1 Artverwandte, ganzheitliche Simulationssysteme Im Bereich der Simulation mobiler Systeme existierten zu Beginn des Projekts bereits unterschiedliche auf einzelne Anwendungsbereiche spezialisierte Simulationswerkzeuge. Nachfolgend soll hierzu ein kurzer Überblick über den damaligen Stand gegeben werden.

ROAMS Ähnliche Ziele wie die des Projektvorhabens Virtual Crater verfolgt die Entwicklung des Simulationssystem „ROAMS“ („Rover Analysis, Modeling and Simulation“) [48], die am Jet Propulsion Laboratory der NASA vorangetrieben wird. Das System dient ausschließlich zur Simulation mobiler, radgebundener Roboter auf fremden Planeten. Hierzu können alle relevanten Komponenten einer solchen Konfiguration simuliert werden: Dynamik, Aktoren, Energieversorgung, Sensoren, Steuerung und Umgebung. Um möglichst authentische Modelle und annähernd optimale Parametersätze für Dynamikverhalten und Sensoren zu erhalten, wurden Experimente in der Realität mit entsprechenden Simulationen verglichen [24]. Es konnte gezeigt werden, auf diesem Wege vielversprechende Optimierungen der Simulationsmodelle möglich sind. In [41] wird ein Modell für die Rad-Gelände-Interaktion vorgestellt, das bodenmechanische Effekte berücksichtigen soll. Es wird angenommen, dass sich alle Effekte durch Kräfte darstellen lassen, die entlang der oder orthogonal zur Oberflächennormale und in einem einzelnen Kontaktpunkt zwischen Rad und Boden wirken. Dieses Kontaktmodell wird ebenfalls durch reale Experimente validiert und die zugehörigen Parametersätze optimiert, um eine maximale Annäherung des Modellverhaltens an die Realität zu erreichen. ROAMS nutzt unterschiedliche freie und nicht freie Softwarebibliotheken [41]. Exemplarisch seien hier nur einige genannt: Es basiert auf dem „DARTS & Dshell“ Framework [10] für die Raumfahrzeugsimulation. Für die Kollisionserkennung kommt SWIFT++ zum Einsatz, für das 3D-Rendering OpenInventor und für Raytracing-Aufgaben POVRAY.

Cyberbotics Webots™ Webots™ ist ein Beispiel für eine kommerzielle Plattform zur Simulation mobiler Roboter. Hierbei handelt es sich um eine Art vollständiger Entwicklungsumgebung für Steuerungsprogramme für mobile Roboter. Der gesamte Entwicklungsprozess bestehend aus Modellierung, Programmierung, Simulation und Transfer von Steuerungssoftware auf eine der unterstützten Hardwareplattformen wird abgedeckt. Für die Modellierung können Geometriedaten in Form von VRML-Dateien importiert oder aber ein interner Editor genutzt werden. Steuerungen werden in C++ oder Java gegen gegebene Interfaces programmiert und auf derselben PC-Plattform wie der Simulator ausgeführt. Alternativ ist eine Anbindung von Matlab-basierten Steuerungen über TCP/IP vorgesehen. Der Haupt-Anwendungsbereich von Webots ist nicht die hochgenaue Simulation von speziellen Roboterkonfigurationen. Vielmehr schafft die Anwendung eine einfache aber sehr vollständige Arbeitsumgebung, um erste Erfahrungen mit eigenen Robotersteuerungen sammeln zu können. Tiefgehende Eingriffe



in das Verhalten von Aktoren, Sensoren oder gar der physikalischen Simulation sowie in die Visualisierung waren zum damaligen Zeitpunkt nicht möglich. Entsprechend seiner Ausrichtung werden hardwareseitig ausschließlich populäre Plattformen wie LEGO Mindstorms™ und Sony Aibo™ unterstützt.

Das Player Framework Player ist ein Open Source Projekt, das unter der GNU General Public License veröffentlicht wird. Player selbst ist kein Simulationssystem, sondern eine Middleware, die ursprünglich eine Abstraktionsschicht zwischen Robotersteuerung und unterschiedlichen Roboter-Hardware-Komponenten realisiert. Player selbst fungiert als ein TCP/IP-Server. Es definiert eine Reihe abstrakter Interfaces, die die Schnittstellen zu üblichen Komponenten eines mobilen Robotersystems widerspiegeln.

Als 3D-Simulations- und Visualisierungssystem kommt zusammen mit Player i.A. „Gazebo“ zum Einsatz, ebenfalls ein Open Source Projekt, das seinerseits auf weiteren Bibliotheken aufsetzt. So wird zur Visualisierung die OGRE3D (<http://www.ogre3d.org>) Engine genutzt und als Dynamiksimulationssystem die Open Dynamics Engine (<http://www.ode.org>) verwendet. Player liefert im Hinblick auf die Abstraktion von Hardware oder Simulation einen interessanten Ansatz, hat gleichzeitig durch die Vorgabe bestimmter Infrastrukturen und eines zum damaligen Zeitpunkt begrenzten Realisierungsstatus jedoch auch große Einschränkungen.

2.4.1.2 Dynamiksimulation Die Geschichte der Starrkörperdynamiksimulation ist umfangreich. Unterschiedliche Ansätze mit spezifischen Vor- und Nachteilen konkurrieren dabei darum, den besten Kompromiss aus physikalischer Korrektheit, numerischer Stabilität und Echtzeitfähigkeit zu bieten. Schaut man sich die tatsächlich realisierten Verfahren an, die einen gewissen Verbreitungsgrad erreicht haben, bleiben im Wesentlichen zwei Verfahren übrig: Die erste Verfahrensklasse, basierend auf Lagrange Multiplikatoren und Zwangsbedingungen, formuliert auf Ebene der Geschwindigkeiten und die zweite Klasse, die impulsbasierten Verfahren. Letztere basieren auf Arbeiten von James Hahn [19] und Brian Mirtich [32]. Ebenfalls zu den impulsbasierten Verfahren wird „Stewarts Methode“ gezählt [43]. Sie war Grundlage für viele weitere Arbeiten, u.a. von Anitescu [2]. Zuletzt stellte Jan Bender [8] eine praxistaugliche Realisierung (<http://www.impulse-based.de>) des eines impulsbasierten Verfahrens mit einigen Optimierungen vor. Daneben nutzt auch die Open Source Dynamiksimulationsbibliothek „Bullet“ (<http://bulletphysics.org>) ein impulsbasiertes Verfahren. Die Methode mit Lagrange Multiplikatoren wurde bereits 1990 in [46] zunächst nur unter bilateralen Zwangsbedingungen, also in gelenkgekoppelten Systemen ohne Kollisionen eingesetzt. Bei dieser Formulierung erfordert die Bestimmung der Gelenk-Zwangskräfte lediglich das Lösen eines linearen Gleichungssystems. David Baraff stellte in [3] erstmals die Erweiterung des Verfahrens auch für unilaterale Zwangsbedingungen und damit für (unelastische) Kontakte und Kontaktreibung vor. Dazu wird typischerweise ein gemischtes, lineares Komplementaritätsproblem (LCP) formuliert und nach dem Algorithmus von Cottle und Dantzig [13] gelöst. Die Open Dynamics Engine, ebenfalls eine Open Source Dynamiksimulationsbibliothek, nutzt diese Methode. Das Simulationssystem der RIF enthält eine eigene Dynamiksimulationskomponente, die ebenfalls



auf diesen theoretischen Grundlagen basiert. Das System hat sich im Einsatz in unterschiedlichen Anwendungen als robust, schnell, genau und sehr praxistauglich erwiesen. Die vollständige Eigenimplementierung ermöglicht gegenüber dem Rückgriff auf externe Bibliotheken detaillierte Einblicke in die Simulationsabläufe sowie die Möglichkeit, an jeder Stelle des Verfahrens eingreifen oder Erweiterungen vornehmen zu können.

2.4.1.3 Bodenmechaniksimulation Die Bodenmechaniksimulation war im Rahmen von Virtual Crater im Hinblick auf die Interaktionseffekte zwischen Laufroboter und Boden von Interesse. Bisher durchgeführte Untersuchungen in diesem Bereich bezogen sich wiederum vor allem auf radgebundene mobile Roboter. Die Mehrheit der Untersuchungen verfolgte dabei einen makroskopischen Ansatz, d.h. es wurden vereinfachte mathematische Modelle der bei der Interaktion wirkenden Kräfte entworfen und durch Experimente validiert und parametrisiert. Ein mikroskopischer Ansatz, bei dem das dynamische Verhalten z.B. einzelner Sandkörner und ihrer Interaktionen untersucht wird, verursacht demgegenüber einen erheblichen Rechenaufwand und schien zum damaligen Zeitpunkt nicht mit dem Anspruch der Echtzeitfähigkeit vereinbar zu sein. Im Folgenden werden exemplarisch einige Arbeiten skizziert. Eine weitverbreitete Theorie zur Beschreibung der Bodendynamik ist die „Bekker Theorie“ [7]. Diese Theorie beschreibt mathematische Performancemodelle für die Mobilität und Lokomotion von radbetriebenen Robotern. In diesen Modellen werden unter anderem Bodenkohäsion, Gravitation und Reibungswinkel als Eingabeparameter verwendet. Ausgabeparameter der Modelle sind „Drawbar Pull“, „Soil Thrust“ und „Motion Resistance“. Zwei planetarische Rover-Exploration-Simulatoren, RMPET (Rover Mobility Performance Evaluation Tool) und RCET (Rover Chassis Evaluation Tools) [31], wurden auf Basis der Bekker Theorie entwickelt. Das Tool RMPET bietet eine theoretische Performanceanalyse für Kontakteffekte zwischen Robotern und Untergründen bestehend aus unterschiedlichem Regolith. Das Tool RCET wurde zur Unterstützung des Entwurfs planetarischer Rover-Chassis entwickelt. RCET beinhaltet zwei Testbeds („Single-Wheel Testbed“ und „System-Level Testbed“) in denen Parameteruntersuchungen für Bodenkontakte durchgeführt werden können. In [5] wird ein Versuchsaufbau mit einem einzelnen Rad auf sandigem Untergrund beschrieben. Ziel der Versuche ist die Bestimmung eines genauen Modells für die Simulation eines Mars-Rovers. Es wird kein eigenes Bodenmechanikmodell entwickelt. Stattdessen werden lediglich Parameter für das kommerziell erhältliche Paket AESCO Soft Soil Tire Model hergeleitet. In [4] wird schließlich beschrieben, wie die Ergebnisse dieser Untersuchungen in ein Simulationssystem auf Basis von Matlab / Simulink einfließen. Kazuya Yoshida und Hiroshi Hamano entwickeln in [49] ein eigenes Modell für die Interaktion zwischen Rad und (weichem) Boden. Ihre Ausgangsgröße für die Modellentwicklung ist der Schlupf zwischen Rad und Boden. Auch in dieser Arbeit werden die Modellergebnisse mit einem realen Versuchsaufbau verglichen, allerdings werden Versuche mit einem vollständigen, 6-rädrigen Rover durchgeführt. Dies ist nur ein kleiner Ausschnitt aus bisherigen Forschungsarbeiten zu diesem Themenfeld. Zusammenfassend kann aber gesagt werden, dass damalige Arbeiten sehr stark auf den Bereich der radgebundenen Roboter bezogen waren, so dass die Ergebnisse nicht ohne weitere, anwendungsspezifische Untersuchungen auf



die Simulation von Laufrobotern übertragbar sind.

2.4.2 Fachliteratur und verwendete Dokumentationsdienste

Für die Entwicklungen im Rahmen von Virtual Crater am DFKI-RIC wurden die in Abschnitt 3 referenzierten Quellen herangezogen.

Das vollständige Literaturverzeichnis ist am Ende dieses Dokuments ab Seite 205 zu finden.

2.5 Zusammenarbeit mit anderen Stellen

Wie im Virtual Crater Antrag (Kapitel 6.2.1) festgelegt, fungierte das MMI (Institut für Mensch-Maschine-Interaktion RWTH AC) im Projekt Virtual Crater als ein Unterauftragsnehmer der RIF (Dortmunder Initiative zur rechnerintegrierten Fertigung (RIF) e.V.) im Rahmen eines F+E Auftrages. Aus diesem Grund kooperierte das DFKI in inhaltlichen Aufgaben mit dem MMI und in administrativen Angelegenheiten mit der RIF.

Das DFKI hat im Projekt Virtual Crater daher eng mit den Partnern RIF und MMI zusammengearbeitet. Diese Zusammenarbeit hat sehr gut funktioniert, so dass unter Anderem die am DFKI entwickelten Algorithmen z.B. zur Simulation und Visualisierung von Bodenmechanik in das Basis-Simulationssystem VEROSIM[®] des Partners eingebunden werden konnten.

Der Partner RIF hat im Verlauf des Projekts eine stereoskopische Mehrschirmprojektionsumgebung in den Räumen des DFKI aufgebaut, die in Verbindung mit VEROSIM[®] und Interaktionsmöglichkeiten wie einem Datenhandschuh eine immersive Nutzung und interaktive Kontrolle der Simulation ermöglicht, unter Einbindung der im Rahmen von Virtual Crater entwickelten Simulationsverfahren.



3 Eingehende Darstellung

3.1 Erzielte Ergebnisse

3.1.1 AP1000: Management

Im Projektzeitraum hat das DFKI die Management-Bereiche Coordination & Risk-Management, Controlling und Contracts für die Arbeitspakete mit DFKI-Anteilen durchgeführt.

Neben den eigenen Kosten wurde auch der Entwicklungsauftrag für die Simulations-/Projektionsumgebung stetig überwacht, der 2011 abgeschlossen wurde.

Das Projekt Virtual Crater begann mit der Förderung des DLR am 01.05.2009. Die Zusammenarbeit der einzelnen Projektpartner ist reibungslos abgelaufen.

Die verantwortlichen Ansprechpersonen für die Projektführung waren:

Dr. Jan Paul (DFKI)

Dr. Yong-Ho Yoo (DFKI)

Dipl.-Inform. Thomas Jung (MMI)

Dipl.-Inform. Michael Saal (RIF)

Wie im Virtual Crater Antrag (Kapitel 6.2.1) festgelegt, fungierte das MMI (Institut für Mensch-Maschine-Interaktion RWTH AC) als ein Unterauftragsnehmer der RIF (Dortmunder Initiative zur rechnerintegrierten Fertigung (RIF) e.V.) im Rahmen eines F+E Auftrages. Aus diesem Grund kooperierte das DFKI in inhaltlichen Aufgaben mit dem MMI und in administrativen Angelegenheiten mit der RIF.

Um den projektbezogenen Datenaustausch zwischen den Partnern zu ermöglichen, hat das DFKI einen gemeinsamen File Server (SVN: <https://svn.hb.dfki.de/Virtual-Crater>) zur Verfügung gestellt.

Am 01.10.2009 fand ein Review-Meeting mit den Teilnehmern DFKI, MMI/RIF, und DLR statt. In diesem Meeting haben DFKI und MMI/RIF zusammen in einer Präsentation dem DLR die Arbeitsergebnisse und die weitere Arbeitsplanung vorgestellt und mit dem DLR diskutiert.

Nach diesem Review-Meeting hat das DLR dem DFKI und MMI/RIF mitgeteilt, dass die Quartalsberichte dem jeweiligen fachlichen Bearbeiter der Agentur übermittelt werden sollen. DFKI und MMI/RIF haben vereinbart, ab dem vierten Quartal einen gemeinsamen Quartalsbericht zu schreiben.

Im vierten Quartal wurde eine zweitägige Einführung in VEROSIM® am MMI in Aachen durchgeführt. Teilgenommen haben an dieser Einführung, die vom MMI gegeben wurde, zwei Mitarbeiter vom RIC Bremen (Michael Rohn und Yong-Ho Yoo). Inhalt der Einführung waren das grundsätzliche Modellieren in VEROSIM® und das Schreiben von Plug-Ins für VEROSIM®. Über diese Plug-Ins kann das DFKI die Simulation er-



weitern und an die vorhandenen Tools wie Matlab oder Adams anbinden. Im Rahmen dieser Einführung hat das DFKI drei Lizenzen für VEROSIM® in der Form von USB-Hardware-Dongles erhalten, so dass auch im DFKI an der Umsetzung in VEROSIM® gearbeitet werden konnte.

Die 2009 durchgeführten ersten Arbeiten auf dem Gebiet der Bodenmechanik wurden auf "ISTVS 2009" Konferenz publiziert und können den folgenden Papern entnommen werden:

- Malte Roemmermann, Daniel Kuehn, Florian Cordes, Yong-Ho Yoo, Frank Kirchner: Concept Evaluation of Modeling Terrain Mechanics by a Neural Network, In 11th European Regional Conference of the International Society for Terrain-Vehicle Systems, (ISTVS-09), 05.10.-10.10.2009, Bremen
- Yong-Ho Yoo, Larbi Abdenebaoui, Michael Rohn, Malte Roemmermann, Frank Kirchner: Modelling and Simulation of Mobile Six-Legged Robot with Leg-Soil Contact In Proceedings of the European regional Conference of the International Society for Terrain-Vehicle Systems, (ISTVS-09), 05.10.-08.10.2009, Bremen.
- Larbi Abdenebaoui, Malte Roemmermann, Yong-Ho Yoo, Frank Kirchner: Optimization of terrain-vehicle simulation using evolutionary algorithms, In Proceedings of the European regional Conference of the International Society for Terrain-Vehicle Systems, (ISTVS-09), 05.10.-08.10.2009, Bremen.

Ein Paper-Abstract über das Projektkonzept und die bisherigen Ergebnisse wurde von DFKI und MMI gemeinsam an i-SAIRAS 2010 eingereicht.

- Yong-Ho Yoo (DFKI), Thomas Jung (MMI), Malte Roemmermann (DFKI), Malte Rast (MMI), Frank Kirchner (DFKI), and Juergen Rossmann(MMI): Developing a Virtual Environment for Extraterrestrial Legged Robot with Focus on Lunar Crater Exploration, i-SAIRAS 2010 (The 10th International Symposium on Artificial Intelligence, Robotics and Automation in Space), August 29 - September 1, 2010, Sapporo, Japan

Bei der Auswertung von Ergebnissen aus den Referenzexperimenten und deren Berücksichtigung in den Simulationsmodellen ist es durch die Komplexität des physikalischen Verhaltens und die dadurch erhöhte Bandbreite von erforderlichen Experimenten zu Verzögerungen im Projekt gekommen. Damit die Verzögerung aufgeholt und das Projekt mit erweiterten qualitativ höherwertigen Ergebnissen erfolgreich abgeschlossen werden konnte, wurde eine kostenneutrale Verlängerung des Projekts bis zum 31.08.2012 beantragt und bewilligt. Die Dortmunder Initiative zur rechnerintegrierten Fertigung (RIF) e.V. stimmte auch für diese Verlängerung und hat diese ebenfalls für sich beantragt und auch diese wurde bewilligt.

Infolgedessen wurden in 2012 auch verstärkt neue Referenzexperimente durchgeführt und für die Aktuatorsimulation wurde auch ein neues unterstützendes präzises Simulationsmodell entwickelt. Ferner wurde der Projektplan für die verbleibenden Arbeitspakete entsprechend angepasst.



Innerhalb des zweiten Quartals 2012 wurde folgendes Paper eingereicht und akzeptiert:

- Yong-Ho Yoo, Lorenz Quack Towards, a Real-Time Capable Realistic Soil Contact Model Using a Particle System, In The 2nd Joint International Conference on Multibody System Dynamics , (IMSD-12), 29.5.-01.6.2012, Stuttgart, o.A., Jun/2012.



3.1.2 AP2000: Konzeption, Anforderungsanalyse

Ein Großteil der Anforderungsanalyse fand auf Seiten des DFKIs statt, da sie hauptsächlich auf die Simulationsanforderungen für Missionen und Szenarien im SpaceClimber- und LUNARES-Projekt basiert. Das aus der Anforderungsanalyse resultierende Dokument (im DFKI-Fileserver: `\\files\research\projects\all\Virtual-Crater\documentation\workpackages\Technische-Arbeitspaket-Berichte_extern\AP2000\main.pdf`) wurde anschließend von RIF/MMI überprüft und bezüglich der Entwicklung und Implementierung an der Simulationssoftware VEROSIM[®] und der Anbindung von Projektionsumgebungen bewertet. Dieses Dokument wurde dann dem DLR zur Verfügung gestellt.

Um für die einzelnen Unter-Arbeitspakete zu AP2000 eine gemeinsame Basis zu erhalten, wurde zunächst für alle Unter-Arbeitspakete zusammen eine Analyse bezüglich Mond- und Marsmissionen durchgeführt. Der Schwerpunkt lag hierbei auf Missionen, die einen Bodeneinsatz beinhalteten oder nennenswerte robotische Systeme einsetzen. Zusätzlich wurden die relevanten Ergebnisse aus dem LUNARES Projekt des DFKIs zusammengefasst. Dazu gehören die Ergebnisse einer Fußdesign-Studie, einer Recherche bezüglich lunarer Krater und einer Abschätzung über den Einfluss unterschiedlicher Umweltbedingungen auf den Ablauf einer Weltraummission. Die Eigenschaften und Ausstattungen der Roboter und die wichtigen Umweltbedingungen bei Weltraummissionen wurden tabellarisch zusammengefasst. Diese Vorarbeiten sollten die Abschätzung vereinfachen, welche Anforderungen an das Simulationssystem bestehen und welche Referenzexperimente im Rahmen dieses Projektes sinnvoll oder sogar notwendig sind.

3.1.3 AP2100: Analyse der Anforderungen an die Benutzerschnittstellen

Die Arbeiten für dieses Arbeitspaket wurden planmäßig in dem Quartal Q2-2009 abgeschlossen. Im Rahmen der Anforderungsanalyse wurden die Anforderungen an die Benutzerschnittstelle erarbeitet. Hierbei hat sich eine Aufteilung in vier wesentliche Aspekte als sinnvoll erwiesen.

1. **Modellbibliothek für grundlegende Simulationsmodelle** Das Simulationssystem sollte eine Art Modellbibliothek anbieten, in der bereits grundsätzliche Simulationsmodelle zum schnellen Einsatz enthalten sind.
2. **Entwicklungsschnittstelle für Erweiterungen** Das Simulationssystem soll die Möglichkeit bieten, die Simulationsverfahren um einzelne angepasste Simulationsmodelle ergänzen bzw. anpassen zu können.
3. **Stereo-Visualisierung:** Für maximale Immersion der Anwender soll das Simulationssystem die Möglichkeit zur echten Stereo-3D-Projektion bieten.
4. **Interaktion:** Bei der Durchführung von Realzeitsimulationen sollen Anwender in die Lage versetzt werden, interaktiv in die laufende Simulation einzugreifen - z.B. durch „Stören“ des Verlaufs durch Veränderung von Umweltbedingungen.



Im Folgenden werden die einzelnen Anforderungen kurz genannt. Für weitere Informationen steht die Anforderungsanalyse als Dokument zur Verfügung.

- **Bibliothek für grundlegende Simulationsmodelle:** Zur schnellen Entwicklung von Prototypen sollte das Softwaresystem eine Art Modellbibliothek anbieten, die solche Modelle bereitstellt und den schnellen Einsatz in ganzheitlichen Simulationsmodellen erlaubt.
- **Entwicklungsschnittstelle für Erweiterungen:** Das Simulationssystem soll eine Entwicklungsumgebung bereitstellen, die die Entwicklung anwendungsspezifischer Simulationserweiterungen für die realzeitfähige Simulation erlaubt.
- **Stereo-Visualisierung:** Das Simulationssystem sollte in der Lage sein, beliebige stereoskopische Visualisierungssysteme, zum Beispiel stereoskopische Monitore, Head-Mounted-Displays sowie Projektionssysteme anzusteuern.
- **Interaktion:**
 - Der am DFKI bereits vorhandene Datenhandschuh soll an die Simulationsumgebung angebunden werden können.
 - Das System sollte offen sein für die Anbindung beliebiger 3D-Eingabegeräte - hierzu ist ein generisches Konzept erforderlich.
 - Die physikalische Simulation sollte Benutzereingaben über 3D-Eingabegeräte berücksichtigen können (Bsp.: "Verschieben eines Steins").

3.1.4 AP2200: Analyse der Anforderungen an Schnittstellen zwischen realer Hardware und Simulationssystem

Die Arbeiten wurden planmäßig in dem Quartal Q3-2009 abgeschlossen. Einerseits wurde die Konsistenz dieses Arbeitspaketes mit den Anforderungen der anderen Arbeitspakete gewährleistet. Andererseits wurden die Anforderungen unter Absprache mit den Projektpartnern angepasst. Für eine systematische Herangehensweise wurden in diesem Arbeitspaket das Steuerungssystem und der zu simulierende Roboter auf verschiedene Ebenen untergliedert. Dabei bezeichnet das Steuerungssystem alle Komponenten, die ein Kontrollsignal generieren. Für die Unterteilung wurde das Steuerungsframework aufgabenbezogen in drei Ebenen gegliedert. Angefangen mit der untersten Ebene, die für die direkte Kommunikation mit der Hardware des Roboters zuständig ist. In die zweite Ebene werden Komponenten eingestuft, die für die Grundsteuerung des Systems notwendig sind. Dazu gehören Lokomotionsverhalten, sowie das Verarbeiten von Sensorinformationen auf reaktiver Ebene. Zuletzt beinhaltet die dritte und höchste Ebene die Steuerungskomponenten, die für Navigation, Planung und weitere "high-level" Verhalten zuständig sind. Aus der beschriebenen aufgabenbezogenen Unterteilung folgt generell auch eine Untergliederung der Komponenten bezogen auf die Datenmenge und die Kommunikationsfrequenz.

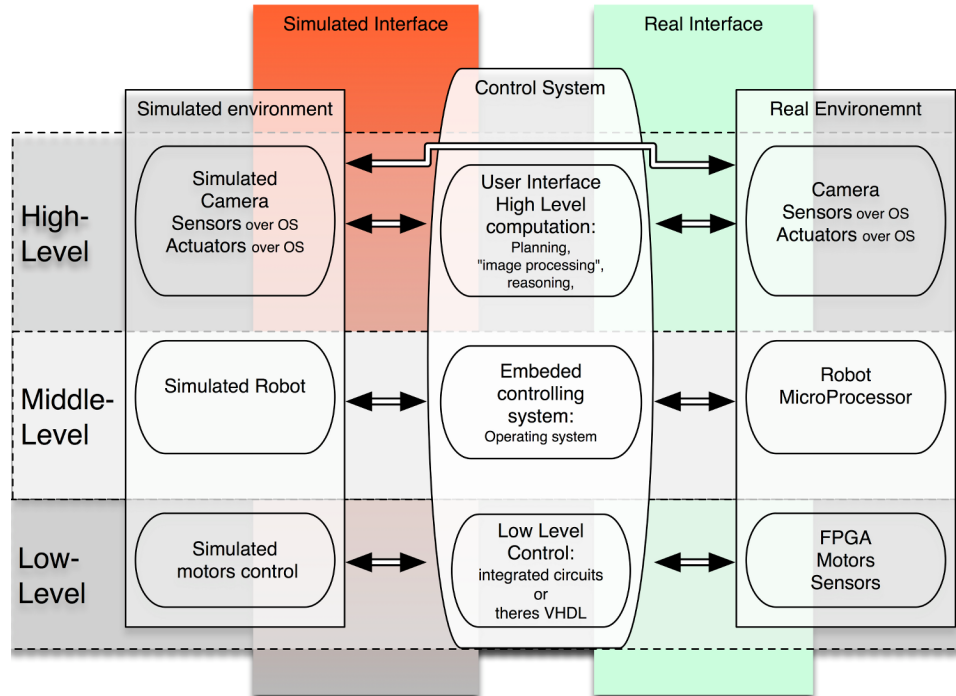


Abbildung 2: Schematische Darstellung der Schnittstellen zwischen dem Kontrollframework und der realen Umgebung rechts bzw. der virtuellen Umgebung links.

So wird auf der untersten Ebene eine sehr hohe Frequenz bei der Übertragung weniger Werte benötigt. Entsprechend wird auf der mittleren Ebene immer noch eine hohe Frequenz mit mittelgroßen Datenmengen benötigt und auf der höchsten Ebene ist meist eine niedrige Datenfrequenz mit einer hohen Datenmenge notwendig. Grund-



sätzlich wird für das Steuerungssystem in jeder Ebene zwischen zwei Schnittstellen unterschieden. Eine für die reale Umgebung und die Andere für die entsprechende simulierte Umgebung, siehe Abbildung 2. Eine genaue Beschreibung und Analyse der drei Ebenen kann der Anforderungsanalyse entnommen werden.

3.1.5 AP2300: Anforderungsanalyse Simulationssystem

Die Arbeiten wurden planmäßig in dem Quartal Q3-2009 abgeschlossen. Die in diesem Arbeitspaket festgelegten Hauptanforderungen sind:

Aktuatoren

Analyse	Zunächst wurde eine Analyse der bestehenden Antriebstechnologien durchgeführt. Aus den vorhandenen Technologien wurden diejenigen, die für Weltraumeinsätze geeignet sind, ausgewählt und in einer genaueren Analyse betrachtet. Abgeleitet aus der Analyse wurden die Anforderungen an die Antriebe erarbeitet.
Motormodell	Eine Übertragungsfunktion von gewünschter Gelenkposition und gewünschter Geschwindigkeit auf eine resultierende Kraft an dem Gelenk muss vorhanden sein.
Parameter	Das Motormodell soll über Parameter der realen Motorspezifikationen konfiguriert werden können.
Spiel	Das Spiel in den Gelenken (Getriebe) muss einstellbar sein um die Fehlposition des Motors korrekt simulieren zu können.
Co-Simulation	Es soll die Möglichkeit bestehen, das Motormodell aus anderen Simulationen zur Laufzeit einzubinden (z.B. sollen präzise Motormodelle aus Matlab/Simulink verwendet werden können). Beim Verwenden der Co-Simulation ist die Echtzeitfähigkeit nicht gefordert. Die Co-Simulation soll lediglich das Entwickeln der abstrakten Motormodelle vereinfachen.



Sensoren

Fehlermodelle	Ein oder mehrere Fehlermodelle müssen mit der Ausgabe aller Sensoren verknüpft werden können.
Kamera	Kameras werden zur Navigation, Hinderniserkennung und Bodenerkennung eingesetzt und müssen daher auch simuliert werden.
Gelenkposition	Die Gelenkposition ist notwendig, damit der simulierte Roboter seinen Zustand kennt und entsprechend reagieren kann.
Gelenkgeschwindigkeit	Die Gelenkgeschwindigkeit wird ebenfalls benötigt für ein internes Modell des Roboters.
Gelenkkräfte	Siehe Gelenkposition und Gelenkgeschwindigkeit.
Energie	Siehe Gelenkposition und Gelenkgeschwindigkeit.
Beschleunigungssensoren	Um Odometrie und äußere Einflüsse messen zu können.
Neigungssensoren	Sind für die Lagebestimmung und Haltungskontrolle notwendig.
Druck- / Kontaktsensoren	Sind notwendig für Lokomotionsverhalten.
Abstandssensoren	Zum Beispiel Infrarot, Ultraschall oder Laserscanner. Die Sensoren werden zur Hinderniserkennung und zum Erstellen eines Weltmodells (Karte) verwendet.

Dynamik

Festkörperdynamik	Muss das Simulieren von festen Körpern mit Masse- und Trägheitseigenschaften erlauben.
Kollisionserkennung	Eine Kollisionserkennung muss ebenfalls in der Simulation enthalten sein, damit die Roboter mit ihrer Umgebung interagieren können.
Kontaktdynamik	Die Kontaktdynamik muss mindestens über Coulomb-Reibung verfügen. Verschiedene Kontaktmodelle sollen zusätzlich über "shared libraries" entwickelt und in die Simulation geladen werden können.
Gelenke	Gelenke stellen Verbindungen zwischen zwei Körpern her. Dabei können sie Verschiebungen und Verdrehungen der Körper zueinander auf festgelegte Achsen reduzieren (z. B. für Scharniergelenke oder Linearverbindungen).
Gelenkdynamiken	Gelenke müssen über eigene Dynamiken verfügen können (z. B. über Feder- und Dämpfungseigenschaften).



Bodendynamik

Bekker-Theorie	Die Bekker-Theorie ist zur Modellierung von Rad-Boden-Kontakten (bzw. Ketten-Boden-Kontakten) in der Industrie und in der Forschung etabliert. Da sich Bodendynamik für das Projekt als sehr wichtiger Faktor herausgestellt hat, soll eine Kontaktbehandlung auf Basis der Bekker-Theorie eingesetzt werden. Dafür müssen allerdings die bestehenden Implementierungen für rad- und kettenbetriebene Fahrzeuge an die Eigenschaften von Laufrobotern angepasst werden.
Bodenmodellierung	Für das Projektziel, der Referenzmondmission, wurde eine Analyse bezüglich der Mondoberfläche bzw. der Kratereigenschaften durchgeführt. Für diese Analyse konnte zu großen Teilen auf das Lunares Projekt zurückgegriffen werden. Anhand der ermittelten Daten sollen die Umgebungen modelliert und die Untergrundeigenschaften gewählt werden.

Darstellung

Eine Analyse wurde zu den benötigten Anforderungen an die Darstellung der Simulationssoftware durchgeführt. Die dabei ermittelten Anforderungen wurden der Anforderungsdefinition beigefügt. Grundsätzlich sollte das schon vorhandene Basissystem "VEROSIM®" die meisten Anforderungen bereits erfüllen.

DFKI-CAD-Modellimport

Da die meisten Roboter im DFKI in Form von CAD-Modellen vorhanden sind, wurden die verschiedenen Formate und die verwendeten CAD-Programme erfasst. Innerhalb des "Virtual Crater" Projektes soll ein weitestgehend automatisierter Weg zum Portieren der CAD-Modelle in die Simulationssoftware entwickelt werden. Dazu wurden Anforderungen an einen entsprechenden CAD Modellimport gestellt, welche ebenfalls der Anforderungsdefinition hinzugefügt wurden.



3.1.6 AP2400: Anforderungsanalyse Referenzexperimente

Als Vorbereitung auf AP3400 (Definition der Referenzexperimente) und AP6100 (Definition einer Referenz-Mondmission) wurden 2009 Informationen zu real durchgeführten extraterrestrischen Bodenmissionen zusammengefasst, in denen erwähnenswerte robotische Systeme eingesetzt wurden. Der Schwerpunkt lag hierbei auf der Ausrüstung und den Leistungen des jeweils eingesetzten robotischen Systems und auf den während der Mission aufgetretenen Problemen. Folgende Missionen wurden betrachtet (Tabelle 1):

Mission	System	Typ
Luna 17	Lunokhod 1	Rover
Luna 21	Lunokhod 2	Rover
Apollo 15	Lunar Roving Vehicle (LRV)	Mondfahrzeug
Viking 1		Orbiter und Lander
Viking 2		Orbiter und Lander
Mars Pathfinder	Rover Sojourner	Lander und Rover
Mars Exploration Rover Mission	Spirit und Opportunity	Rover

Tabelle 1: Betrachtete Weltraummissionen

Zusätzlich wurden für Virtual-Crater relevante Informationen aus dem ebenfalls am DFKI durchgeführten Projekt LUNARES zusammengefasst. Dazu gehören die technischen Daten des verwendeten Roboters Scorpion, die Ergebnisse der durchgeführten Fußdesign-Studie, eine Abschätzung, in welchem Maße Umweltbedingungen extraterrestrische Missionen beeinflussen können, und eine Beschreibung der im Space-Testbed vorgenommenen Demomission. Abschließend wurde eine Tabelle angefertigt, in der die Eigenschaften und die Ausrüstung der Systeme zusammengefasst wurden.

Referenz-Experiment-Setups mit einer effektiven Tool-Chain

VEROSIM[®] ist eine Echtzeit-Simulationsumgebung, die sich generell besser für reaktive Input-/Output-Funktionalitäten und visuelle Plausibilität eignet, während Nicht-Echtzeit-Simulationssysteme wie Adams, Matlab oder 20-sim besser für die Analyse präziser Verhaltensweisen geeignet sind. Um die Systemparameter zu optimieren und bestimmte Lücken zwischen realen und virtuellen Umgebungen zu verringern, benötigen wir eine effektive Tool-Chain zwischen präziser Simulation und Echtzeit-Simulation, mit der die Konstruktionen der Systeme, die Bodenkontaktmodelle und ihre Parameter, die mit präzisen Simulationen analysiert wurden, in die Echtzeitsimulation übertragen und in diese integriert werden. Um den Realismus der Simulation in VEROSIM[®] zu verbessern, benötigen wir zur Unterstützung präzise Simulationssysteme wie Adams, 20-sim und Matlab/Simulink. Es ist beispielsweise möglich, dass ein mathematisches Modell, das aus dem echten Antrieb abgeleitet wurde, vollständig in einem präzisen auf Adam/Matlab basierenden Modell getestet wurde und dass dieses

Modell dann zu einem weniger komplexen abstrakten Modell vereinfacht werden kann, um dieses in der Echtzeitsimulation VEROSIM[®] einzusetzen. Das wichtigste Ziel der Referenzexperimente ist es, die Systemparameter des Laufroboters (SpaceClimber) und die Umgebungsparameter für die Simulation der Bodendynamik zu finden und zu optimieren. Das folgende Schema ist für die Analyse und die abgleichende Anpassung der Referenzexperimente sowohl in der realen als auch der virtuellen Testumgebung erforderlich:

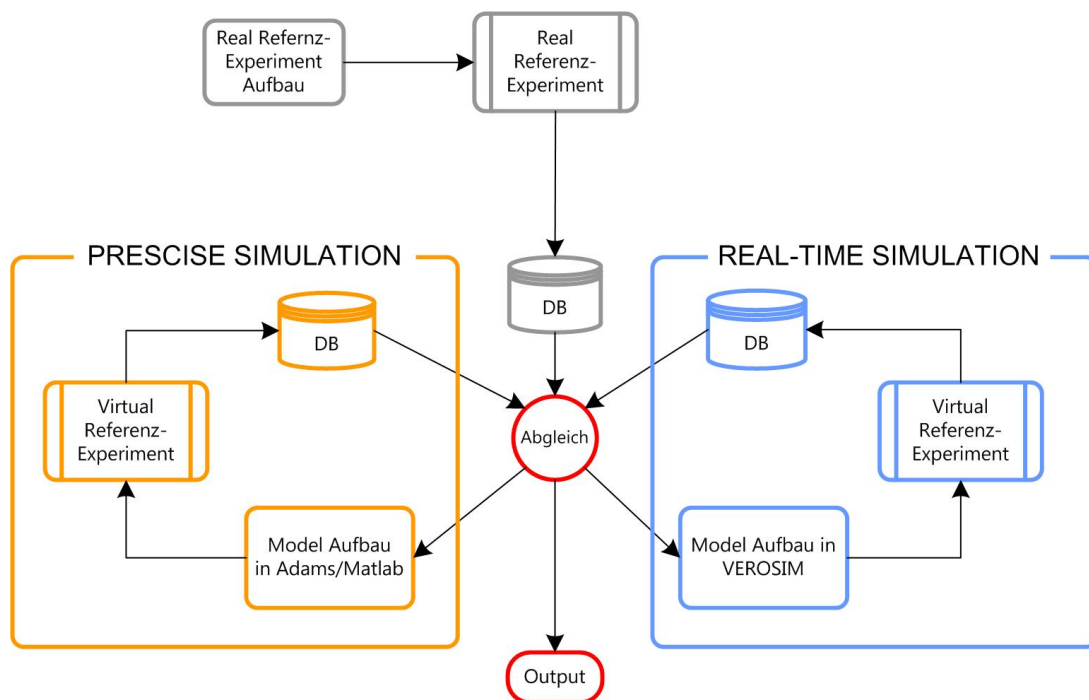


Abbildung 3: Reference Experiment Setup

Die Hauptzielsetzung dieser Testumgebungen ist die realistische Simulation von Laufrobotern auf der Mondoberfläche. Daher liegen die wichtigsten Aspekte der Simulation, die so angepasst werden müssen, dass sie der Realität entsprechen, in der Simulation der Antriebe, der Sensoren und der Bodendynamik bei der Interaktion zwischen den Beinen und der Mondoberfläche. Für jeden dieser Aspekte werden Referenzexperimente definiert. Die Interaktion zwischen Beinen und Boden steht in Zusammenhang mit der bekannten Forschung zur Rad-Boden-Interaktion für extraterrestrische Rover-Missionen.

Anforderung

- Referenz-Experimente sollten zusammen mit einer effektiven Tool-Chain aufgebaut werden, die dann verwendet wird, um Systemparameter zu optimieren, indem die gegebenen lunaren Szenarien präzise modelliert und simuliert werden.
- Das wichtigste Ziel dieser Testumgebung ist die realistische Simulation von



Laufrobotern auf der Mondoberfläche. Daher sollten die Verhaltensweisen des Laufroboters (SpaceClimber) durch Referenzexperimente in verschiedenen Detailstufen systematisch analysiert werden.

- Es sollte in Referenzexperiment für die Bein-Boden-Interaktion aufgebaut werden, mit einem Substrat, das Regolith simuliert. Auf diesem Substrat sollten Beinbewegungen durchgeführt und analysiert werden.

Automatische Optimierung

Bei der automatischen Optimierung der Simulation handelt es sich um eine Parameteroptimierung. Dabei werden für jedes Referenzexperiment Parameter mit zugehörigen Wertebereichen definiert, die das Verhalten der Simulation maßgeblich beeinflussen.

Die real gemessenen Daten jedes Referenzexperiments werden in einer einheitlichen Form archiviert.

Die Schnittstelle zum Bewerten eines Parametersatzes ist für jedes Referenzexperiment gleichermaßen definiert. Dabei muss jedes Experiment (in Form eines Programmes oder einer dynamischen Programm-Bibliothek) eine Funktion zur Verfügung stellen, die n Werte als Eingabeparameter verarbeitet und einen Rückgabewert als Genauigkeitsmaß liefert.

Sobald die Simulationsparameter und die Bewertungs-Schnittstelle für ein Referenzexperiment gegeben sind, kann ein Optimierungsverfahren aus dem Bereich der KI eingesetzt werden, um das Simulationsverhalten an die Realität anzupassen. Dafür ist der Algorithmus "CMA-ES" aus dem Bereich der Evolutionsstrategien vorgesehen. Dieser Algorithmus hat sich im DFKI bereits mehrfach zur Parameteroptimierung als nützlich erwiesen.

3.1.6.1 Probleme bei der Verbindung von Simulationsmodellen Werkzeuge zur Simulation von Physik wie Adams oder VEROSIM[®] basieren auf der Berechnung einzelner Zeitschritte und zeigen daher eine Verzögerung zwischen Ein- und Ausgabe. Durch diese Verzögerung kann das System instabil werden oder Eingabedaten verlieren. Um das Simulationsmodell stabil zu machen, ist es wichtig, die Probleme bei der Verbindung von Signalmodellen (Motoren und Controller) und kinematischen Modellen (Mechanik-Simulation) in VEROSIM[®] zu bedenken. Ein einfaches Gelenkmodell bestehe aus einem Gelenk-Controller und Gelenk-Kinematik. In einer Simulation, welche die Kraft-Ausgabewerte des Gelenk-Controllers auf die Gelenk-Kinematik überträgt, kann die Geschwindigkeit des Gelenks durch der Integration der ausgegebenen Kraft erhalten werden. Eine Simulation, die stattdessen eine Geschwindigkeit von Gelenk-Controller erhält und diese auf die Gelenk-Kinematik überträgt, kann die Kraft, die auf das Gelenk wirkt, durch Ableitung der Eingabe-Geschwindigkeit erhalten. Da die Frequenz der Gelenk-Kraft höher ist als die der Gelenk-Geschwindigkeit, kann eine Simulation,



die Kräfte als Eingabewerte erhält, instabiler sein als eine Simulation, die Geschwindigkeiten mit der gleichen Sampling-Rate als Eingabewerte erhält. Andererseits wird die Simulation, die Geschwindigkeiten verwendet, oft Ableitungsfehler bei der Kraftberechnung erhalten. Solche Probleme bei der Verbindung der Signalmodelle mit den kinematischen Modellen bei wechselnden Sampling-Raten sollten bei den virtuellen Referenzexperimenten berücksichtigt werden.

Referenzexperimente für Aktuatoren

Das vorgeschlagene System erlaubt die realistische Modellierung von Robotern, Sensoren und Aktuatoren und auch komplexen unstrukturierten dynamischen Umgebungen, die in einer fast realistischen Simulationsumgebung miteinander interagieren. Der Nutzen dieser Simulationen hängt stark davon ab, wie sehr sie validiert wurden und Validierungs-Daten vorliegen, um sie beim Modellieren von Komponenten zu nutzen und die Korrektheit der Ergebnisse zu beurteilen.

Damit Simulation und Realität besser übereinstimmen, wird das Verhalten eines echten Aktuators unter verschiedenen Belastungen und Betriebsbedingungen mit den Resultaten entsprechender Simulationen so präzise wie möglich verglichen.

Da die Anforderungen an die Aktuator-Referenzexperimente von dem zu verwendenden Aktuator abhängen, können keine spezifischen aber generelle Anforderungen an solche Experimente gestellt werden:

Sensoren

Simulationsumgebungen können genutzt werden, um Controller für robotische Systeme zu entwickeln. Hierbei werden die den Roboter betreffende Parameter durch einen Optimierungsverfahren wie CMA-ES [20] modifiziert und anschließend das Verhalten des Roboters in der Simulation beobachtet und bewertet. Auf diese Weise kann die Controller-Entwicklung beschleunigt und das reale System geschont werden.

Je genauer dabei die Realität in der Simulation nachgebildet wird, desto besser können die Ergebnisse aus der Simulation auf das reale System übertragen werden. Hierzu gehört eine sorgfältige Simulation der verwendeten Sensoren, deren Anzahl und Relevanz im robotischen System mit wachsenden Aufgabenbereichen und steigenden Autonomiegraden beständig zunimmt. Ein komplexes System agiert und reagiert auf Basis der Messdaten der internen und externen Sensoren. Sofern die Sensoren in der Simulation nicht ausreichend genau nachgebildet werden, wird sich der Roboter in der Realität nicht wie gewünscht verhalten,[11].

Bei der Sensorsimulation sollte ihr Rauschverhalten berücksichtigt werden. Zusätzlich wird das Verhalten vieler Sensoren von der Umgebung, in der sie eingesetzt werden, beeinflusst. Zu den beeinflussenden Faktoren gehören unter anderem Magnetismus, Luftdruck, Temperatur, Feuchtigkeit und Lichtverhältnisse. Die Sensorwerte der meisten Abstandssensoren werden zusätzlich von der Beschaffenheit und Farbe des Untergrundes beeinflusst. Im Rahmen der Referenzexperimente soll das genaue Rausch-



verhalten der relevanten Sensoren und die sie beeinflussenden Umgebungsfaktoren ermittelt werden.

Sensorrauschen

Die verwendeten Sensormodelle und Sensorrauschverhalten bestimmen maßgeblich die Verwendbarkeit der Simulationsergebnisse. In [25] wird gezeigt, dass eine erfolgreiche Controller-Entwicklung möglich ist, sofern die Sensormodelle sorgfältig aufgestellt werden und das Rauschen der Sensoren realistisch simuliert wird. Abbildung 5 zeigt das Verhalten eines mit Infrarot- und Umgebungslichtsensoren ausgestatteten Khepera-Roboters in der Realität und in der Simulation. Wie zu erkennen ist, führt nur der Controller zu einem vergleichbaren Verhalten, der unter Verwendung des realen Sensorrauschens trainiert wurde. Bei dem in diesem Paper beobachteten Rauschen handelte es sich um grobe Gauß-Verteilungen. In den Referenzexperimenten soll das genaue Rauschverhalten der Sensoren ermittelt und mit dem im Datenblatt angegebenen Rauschen verglichen werden. Es kann angenommen werden, dass das Rauschverhalten der meisten Sensoren anhand einer Gauß-Funktion beschrieben werden kann.

Umgebungsfaktoren

Für nahezu alle Sensoren existieren Umgebungsfaktoren, welche die Messergebnisse verfälschen. So werden IMUs generell durch Magnetismus, Ultraschallsensoren durch Luftdruck oder Drucksensoren durch die Umgebungstemperatur beeinflusst. In Tabelle 3.1.6.1 sind die theoretisch relevanten Umgebungsfaktoren zusammengefasst.

Eine realistische Simulation müsste dazu in der Lage sein, all diese Umgebungsfaktoren zu berücksichtigen. Dabei würden die Umgebungsfaktoren einmalig durch den Benutzer definiert und anschließend automatisch mit den Messwerten der Sensoren verrechnet werden. Auf diese Weise würde eine Simulationsumgebung erstellt werden, mit der nicht nur terrestrische, sondern auch extraterrestrische und Unterwasser-Umgebungen simuliert werden könnten.

Der Aufwand, der betrieben werden müsste, um alle Umgebungsfaktoren in der Simulation zu berücksichtigen und entsprechenden Referenzexperimente durchzuführen (besonders 1 bis 3), würde jedoch den Kosten-Nutzen-Faktor übersteigen und den Rahmen des Projektes übersteigen. Da der Schwerpunkt des Projektes außerdem in der Simulation der Bodendynamik liegt, sollte sich die Simulation der Sensoren auf die Berücksichtigung der Umgebungsfaktoren 5 und 6 (optional 7) beschränken. Dies betreffe hauptsächlich die Simulation der Abstandsensoren und der Kameras und könnte ohne größere Modifikationen in VEROSIM[®] umgesetzt werden. Dennoch sollte bei den Referenzexperimenten zumindest die Temperatur zum Zeitpunkt der Durchführung vermerkt werden.



Anforderungen

Anforderung 3.1. *In den Referenzexperimenten soll das Rauschverhalten der realen Sensoren ermittelt werden, die in VEROSIM[®] umgesetzt werden.*

Anforderung 3.2. *In den Referenzexperimenten soll das abweichende Verhalten der Sensoren, welches durch unterschiedliche Lichtverhältnisse und Oberflächenfarben hervorgerufen wird, definiert werden.*

Anforderung 3.3. *Die Referenzexperimente sollen es ermöglichen, eine Abbildung zu definieren, welche für jeden Sensor einen realistischen Ausgabewert liefert.*

Referenzexperimente für die ganzheitliche Simulation

Die bisher betrachteten Referenzexperimente beschäftigen sich mit den einzelnen Komponenten der Simulation. Schlussendlich sollen diese ein genaues Verhalten der ganzheitlichen Simulation ermöglichen. Ob das Zusammenspiel der einzelnen Komponenten zu dem gewünschten Ergebnis führt muss mit weiteren Referenzexperimenten untersucht werden. Hierfür muss das Gesamtverhalten von autonomen Robotern vermessen werden. Um dies zu ermöglichen soll ein Motion-Tracking Verfahren eingesetzt werden, welches die Positionen im Raum der einzelnen Komponenten des Roboters aufzeichnen kann. Zu den räumlichen Positionen sollen interne Sensorwerte synchron aufgezeichnet werden. Insbesondere sind dabei die Zustände der Aktuatoren interessant. Analog zu den realen Experimenten, müssen die Verhalten der Roboter in der Simulation ausgeführt und aufgezeichnet werden.

Die Genauigkeit die erreicht werden soll, muss eine Simulations gestützte Aussage bezüglich der Mobilität eines robotischen Systems in einer definierten Umgebung ermöglichen. Im speziellen sollen für definierte Bodeneigenschaften hervorgesagt werden können ob ein robotisches System eine Steigung beschreiten kann und welche Lokomotionsmöglichkeiten dem System zur Hindernisvermeidung in der Steigung zur Verfügung stehen (z.B. ob ein Laufroboter in einer Steigung noch definierte Kurven laufen kann).

Anforderung 3.4. *In einer Definierten Umgebung sollen die Sensorwerte zusammen mit der räumlichen Position des Roboters und seiner Extremitäten aufgezeichnet werden.*

Anforderung 3.5. *Die zu erreichende Simulationsgenauigkeit soll eine Aussage über die Mobilität eines Roboters in definierter Umgebung erlauben.*

Die Arbeiten an diesem AP wurden in Q2-2010 abgeschlossen.

Mechanisch

Die Möglichkeit, den Aktuator in allen Bewegungsrichtungen zu testen CW,CCW,...)
 Die Möglichkeit, den Aktuator in allen möglichen Geschwindigkeiten zu testen
 Die Möglichkeit, die Winkelgeschwindigkeit unter verschiedenen Bedingungen zu messen
 Die Möglichkeit, das Drehmoment unter bestimmten spezifizierten Bedingungen zu messen

Elektrisch

Die Möglichkeit, grundsätzliche Motorparameter zu messen
 Die Möglichkeit, Stromstärke und Spannung zu messen
 Die Möglichkeit, die Leistungsaufnahme zu berechnen oder zu messen

Weiteres

Automatisches Mitprotokollieren von Messdaten
 Die Möglichkeit, relevante Testdaten in das System zu importieren
 Die Möglichkeit, relevante Daten aus dem System in das Test-Untersystem zu übertragen
 Automatisches Berichten und Dokumentieren der Testprozedur und der Resultate

Tabelle 2: Anforderungen an Aktuator-Referenzexperimente

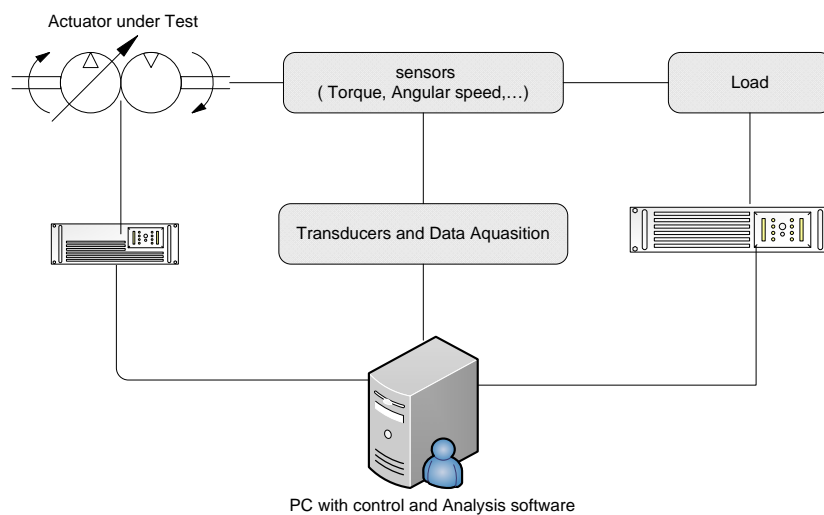


Abbildung 4: Basisaufbau der Aktuatorexperimente

gemessene Parameter	berechnete Parameter
Drehmoment	mechanische Leistung
Winkelgeschwindigkeit	elektrische Leistung
Stromstärke	Effizienz
Temperatur	

Tabelle 3: System-describing parameters of the joints

1	Temperatur
2	Magnetismus
3	Umgebenes Medium / Druck
4	Verschleiß
5	Lichtverhältnisse
6	Oberflächenfarbe
7	Oberflächenmaterial

Tabelle 4: Relevante Umgebungsfaktoren

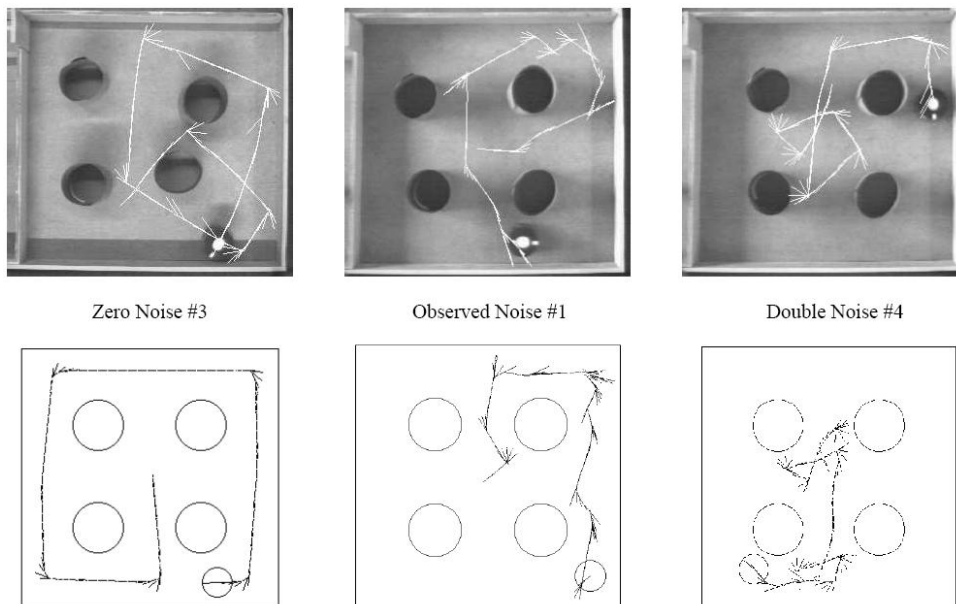


Abbildung 5: Verhaltensabweichung eines Khepera Roboters in der Realität und Simulation hervorgerufen durch unterschiedliches Rauschen beim Lernen des Controllers



3.1.7 AP3000: Definitions- und Entwicklungsphase

Basierend auf dem Anforderungsdokument (bzgl. AP2000) wurden die entsprechenden Definitionen und Spezifikationen der zu implementierenden Komponenten für den weiteren Projektfortschritt erstellt. Die Koordination und hauptsächliche Arbeit für AP3110 liegt beim RIF. Im DFKI wurde ein Szenario erarbeitet, welches die gewünschte Interaktion mit der Simulation VEROSIM[®] zeigen soll. Das Arbeitspaket AP3120 wurde hauptsächlich vom RIF durchgeführt. Im DFKI wurde dennoch begonnen, Modelle und Erweiterungen zu untersuchen und zu programmieren, um eine geeignete Lösung zu finden. Das Arbeitspaket AP3200 wurde laut Projektantrag von dem Projektpartner RIF übernommen. Allerdings wurde im DFKI ein Kommunikationsprotokoll auf Basis des Roboterbetriebssystems MONSTER [42] und ein Co-Simulation Framework auf Basis des MSC ADAMS und MATLAB definiert. In Arbeitspaket AP3310 wurden visuelle Ergänzungen für VEROSIM[®] definiert, welche das intuitive Verständnis bei der Durchführung einer Simulation verbessern. Diese Ergänzungen werden nachfolgend aufgezählt und kurz beschrieben. Zusätzlich wurden für jede Ergänzung mit 3ds Max Grafiken angefertigt, welche die Beschreibung der Ergänzung visuell unterstützen. Von diesen Grafiken wird aus Platzgründen nur eine kleine Auswahl aufgenommen. In Arbeitspaket AP3400 wurden die Referenzexperimente definiert, anhand derer die Validierung der Simulation durchgeführt wird. Die Beschreibung eines Referenzexperimentes beinhaltet die Umgebung, die zu testenden Komponenten und die benötigten Messtechniken.

3.1.8 AP3100: Definition und Spezifikation der Benutzerschnittstelle zwischen Simulationssystem und Simulationsanwender

3.1.9 AP3110: Visualisierung / Bedienung / Interaktion

Die Arbeiten wurden planmäßig in Quartal Q4-2009 abgeschlossen. Die Koordination und hauptsächliche Arbeit für dieses AP liegt beim RIF. Im DFKI wurde ein Szenario für AP3110 erarbeitet, welches die gewünschte Interaktion mit der Simulation VEROSIM[®] zeigen soll, und es wurde die Unterbringung für das 3D-Projektionssystem (CAVE), das im Rahmen des Projektes angeschafft wurde, geklärt. In diesem Szenario wird zunächst der Laufroboter mit Hilfe des Datenhandschuhs und des Trackingsystems, welches ein 6-DOF-Tracking ermöglicht, in der 3D-Umgebung platziert. Hierbei hilft das 6-DOF-Tracking erheblich, da so die Lage des Roboters einfach durch die Lage der getrackten Hand beeinflusst wird. So wird es möglich, den Roboter auf einem Hang abzusetzen, ohne dass initial eine ungünstige Pose eingenommen wird. Der zweite Teil des Szenarios besteht darin, dass durch eine Greif-Geste ein Stein aufgenommen werden kann. Dieser könnte dann dem Roboter als Hindernis in den Weg gelegt werden. Um dieses Szenario zu verdeutlichen, wurde die Abbildung 6 erstellt.

Das System sollte ursprünglich in dem Laborraum im Erdgeschoss der 2009 gerade im Bau befindlichen Space-Explorations-Halle untergebracht. Abbildung 7 zeigt, wie die CAVE in dem Labor aufgestellt werden sollte. Die rot eingezeichneten Strecken zeigen die Durchgänge für die Türen an. Die Rechtecke mit 1,52 Meter bzw. 2,40



Abbildung 6: Intuitive Manipulation mit einer virtuellen Hand.

Meter stellten die Grundfläche für jede Projektionsbox dar. Diese Abmessungen wurden durch die Breite der Projektionsfläche (1,52 Meter) und die nötige Entfernung der Beamer vorgegeben.

Der CAVE besteht aus fünf Projektionsboxen mit jeweils 100 Zoll Projektionsfläche. Diese Boxen sollten ursprünglich in einem Winkel von 45° bzw. 135° zueinander aufgestellt werden. So bekäme der Benutzer schon mit 5 Wänden einen 180° Rundumblick. Pro Projektionsfläche werden zwei Beamer mit passivem Polarisationsfilter eingesetzt. Durch das Tragen einer Brille mit entsprechenden Polarisationsfiltern nimmt der Betrachter auf jedem Auge ein Bild aus einer leicht anderen Perspektive wahr, was zu einem 3D-Eindruck führt. Auch wenn der 3D-Eindruck im Zentrum des CAVE am größten ist, können mehrere Personen gleichzeitig diese Erfahrung erleben.

Um die Bilder für das System zu generieren wird pro Beamer ein Rechner mit leistungsstarker CPU und Grafikkarte benötigt. Zu diesen zehn Rechnern kommen noch mindestens zwei weitere, die sich um das Verteilen der Simulation auf die einzelnen Projektionsflächen kümmern. Da diese Gerätschaften ein großes Maß an Wärme abgeben, ist eine Klimatisierung des Labors notwendig. Eine erste Abschätzung der entstehenden Wärme ist den Tabellen 5 und 6 zu entnehmen. 2009 wurden Angebote für die Klimatisierung eingeholt und die Umsetzung diskutiert.

Die CAVE wurde entgegen dieser ersten Planung schließlich in einem neu angemieteten Bürogebäude mit angepasstem Aufbau installiert.

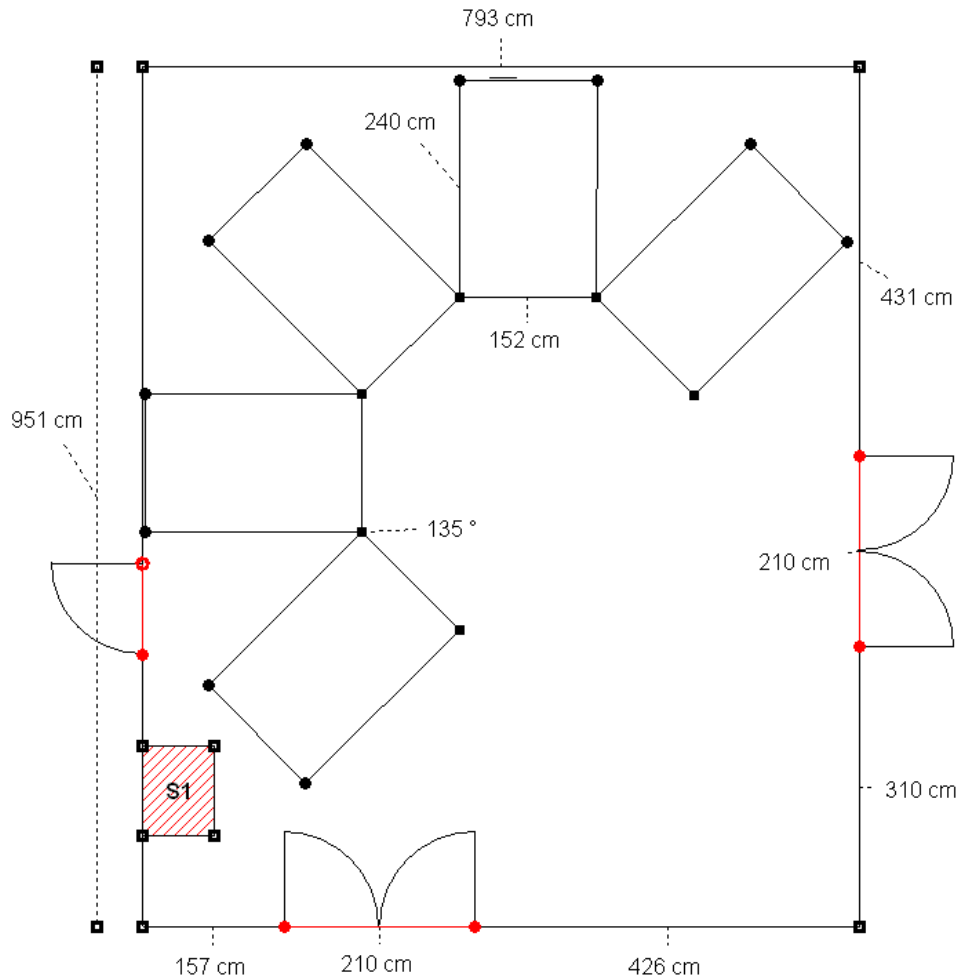


Abbildung 7: Skizze des Laborraums mit CAVE

Art	Watt	Anzahl	Summe Watt
Beamer	600	10	6000
PCs	500	12	6000
Serverschrank	3000	1	3000
Arbeitsplatzrechner	500	3	1500
			16500

Tabelle 5: Abschätzung der Entstehenden Wärme im Labor

Art	Watt	Anzahl	Summe Watt
Beamer	600	4	2400
PCs	500	5	2500
			4900

Tabelle 6: Abschätzung der zusätzlichen Wärme bei Erweiterung auf eine 7-fach Projektion



3.1.10 AP3120: Programmiermodell für Erweiterungen

Das Programmiermodell für Erweiterungen soll den Anwendern des Simulationssystems die Möglichkeit geben, eigene physikalische Modelle und Zusammenhänge in eine bestehende Simulationsumgebung integrieren zu können. Ein Beispiel hierfür sind spezialisierte Energieverbrauchsmodelle, die in Abhängigkeit von aktuell anliegenden Drehmomenten und Drehraten in den Gelenken eine Leistungsaufnahme und den Energieverbrauch während eines Experimentes bestimmen. Zur Integration eines solchen Programmiermodells stellt VEROSIM® diverse Schnittstellen bereit. So könnte die Anbindung z.B. auf der Input- /Output-Infrastruktur aufsetzen und hierdurch Verbindungen und funktionelle Zusammenhänge realisieren.

Dieses Arbeitspaket wurde hauptsächlich vom RIF durchgeführt und wurde planmäßig in dem Quartal Q4-2009 abgeschlossen. Das Arbeitspaket behandelt konkret die Aufgabe, den Mitarbeitern am DFKI eine Entwicklungsumgebung für VEROSIM® bereitzustellen, die ihnen für ihre Forschungs- und Entwicklungsarbeiten in Virtual Crater maximale Flexibilität und den notwendigen Leistungsumfang von VEROSIM® zur Verfügung stellt. Die Partner haben sich dazu entschieden, dazu die C++-Entwicklungsschnittstelle von VEROSIM® offenzulegen. Am 18. und 19. November waren dazu einige Mitarbeiter des DFKI am MMI in Aachen zu Besuch, wo sie eine entsprechende erste Schulung bekommen sowie die notwendige Software und einige Beispiel-Plugins (Sensor-Simulation, Motor-Modell) zur Verfügung gestellt bekommen haben. Damit steht am DFKI jetzt die Schnittstelle mit größtmöglicher Flexibilität zur Verfügung.

3.1.11 AP3200: Definition und Spezifikation der Schnittstellen zwischen realer Steuerung und Simulation

Das Steuerungssystem ist unter anderem für das Lokomotionsverhalten zuständig und arbeitet in der Regel mit einer Updatefrequenz zwischen 50 und 100 Hz. In den realen Robotern ist die Schnittstelle fest eingebettet und die Hardware ist direkt mit dem Controller-Board verbunden. Für die Steuerung der simulierten Roboter muss das Kontrollprogramm auf einem standard PC ausgeführt werden. Dadurch kann es dann über eine Socket-Verbindung an die Simulation angekoppelt werden. Diese Möglichkeit bietet das im DFKI verwendete Roboterbetriebssystem "MONSTER" bereits. Auf dieser Kommunikationsebene müssen die gewünschten Aktuatorwerte vom Betriebssystem an die Simulation übertragen werden. Die Rückrichtung beinhaltet Sensorwerte, die aus der Simulation als Feedback an das Betriebssystem versendet werden müssen. Um eine einfache Verknüpfung zwischen dem Betriebssystem und der Simulation für verschiedene Roboter gewährleisten zu können, sollte ein Protokoll festgelegt werden, welches abhängig von den verwendeten Aktuatoren und Sensoren den notwendigen Datenaustausch definiert. Dabei sollen die Datenmenge und die Datenformate genau der Datenkommunikation zwischen dem Betriebssystem und der realen Hardware entsprechen. In Abbildung 8 ist die Kommunikation auf der "Middle-Layer" Ebene dargestellt. Aus Perspektive der Kommunikation besteht die "Middle-Layer" Kommunikationsschleife aus drei Schritten:

1. Transfer von Sensorwerten
2. Berechnungsschleife des Betriebssystems (Verarbeiten der Sensoren und berechnen der Aktuatorwerte)
3. Transfer der gewünschten Aktuatorwerte

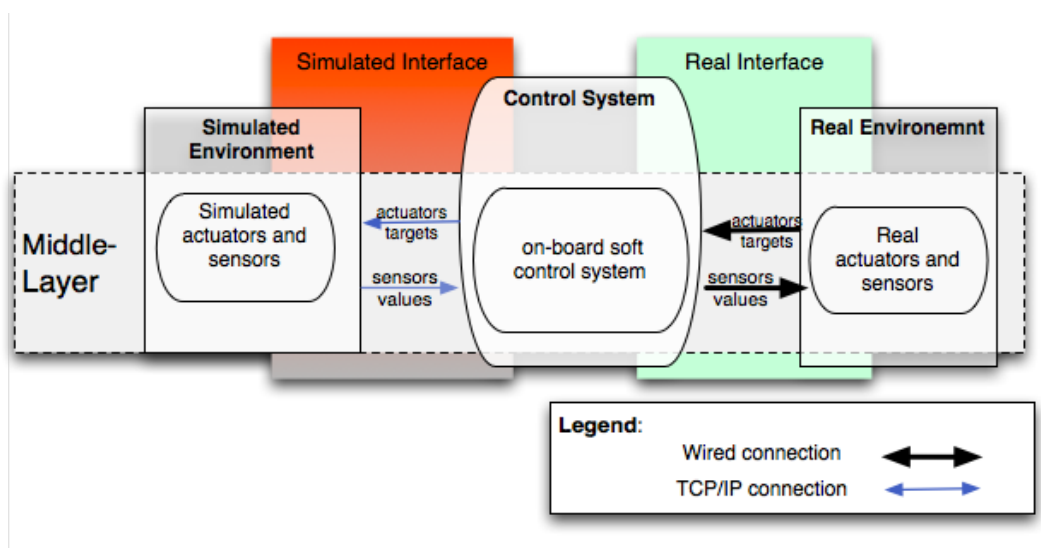


Abbildung 8: Eine Darstellung der Middle-layer Schnittstelle zwischen der realen Steuerung und der Simulation oder dem realen Roboter.

Die Koordination und hauptsächliche Arbeit für dieses AP lag beim RIF. Im DFKI wurde ein Kommunikationsprotokoll auf Basis des Roboterbetriebssystems MONSTER [42] und ein Co-Simulation Framework auf Basis des MSC ADAMS und MATLAB definiert. Die Kommunikation zwischen der Robotersteuerung und der Simulation sollte über eine Socket-Schnittstelle durchgeführt werden. Dies erlaubt, dass die Robotersteuerung ein eigenständiges Programm ist und sehr einfach entweder auf dem realen Roboter oder auf einem Arbeitsplatzrechner in Kombination mit der Simulation ausgeführt werden kann. Für die Kommunikation müssen die Aktuatorwerte und die Sensorwerte mit einer definierten Frequenz zwischen der Robotersteuerung und der Simulation ausgetauscht werden. Damit die Frequenz, gemessen an der virtuellen Zeit, unabhängig von der benötigten realen Rechenzeit für einen Simulationsschritt ist, soll eine synchrone Socketverbindung verwendet werden. Dies bedeutet, dass die Simulation und auch die Robotersteuerung blockiert werden, bis eine Aktualisierung der jeweiligen Aktuator- und Sensordaten erfolgt ist. Die Datenübertragung sollte auf ASCII Basis stattfinden, was das Debuggen und Loggen der Kommunikation ohne das Entwickeln weiterer Programme ermöglicht.

Für den Austausch der Daten werden zunächst die Sensordaten aus der Simulation an die Robotersteuerung versendet. Das zu versendende Format für einige definierte Sensoren ist in der Tabelle 7 aufgeführt. Dabei enthält die Spalte "c-Format" den Formatierungsstring der standard c-Funktion `printf()`. Analog zu den Sensoren werden dann alle Aktuatorwerte im ASCII Format an die Simulation übertragen. Die Tabelle



beinhaltet einige definierte Sensoren und kann für weitere Sensoren erweitert werden.

Sensor	Anzahl Bytes	c-Format
Positions-Sensor	33	%10.4f %10.4f %10.4f
Rotations-Sensor	21	%6.2f %6.2f %6.2f
Geschwindigkeits-Sensor	33	%10.4f %10.4f %10.4f
Kontakt-Sensor	2	%1d
Kontakt-Kraft-Sensor	10	%9.3f
Center-Of-Mass-Sensor	21	%6.2f %6.2f %6.2f
Gelenk-Positions-Sensor	11	%10.7f
Gelenk-Kraft-Sensor	7	%6.2f
Gelenk-Last-Sensor	7	%6.2f
Abstands-Sensor	7	%6.2f

Tabelle 7: Socket Kommunikationsprotokoll für einige definierte Sensordaten.

Co-simulation Framework Dieses Kapitel enthält die Definition eines Co-Simulations-Frameworks, das Simulationswerkzeuge verschiedener Anbieter, zusätzlich zu selbst entwickelten Werkzeugen, nahtlos miteinander verbindet. So müssen Teile oder Komponenten nicht länger unabhängig voneinander entwickelt werden. Weiterhin erleichtert Co-Simulation die Entwicklung einer Integrationsmethodik für das Design, das Testen und die Parameteroptimierung der verschiedenen Komponenten des Roboters und verbessert die Verifikation der virtuellen Modelle mit Hilfe realer Referenzexperimente und befähigt VEROSIM[®], detaillierte Analysen präziser Verhaltensweisen zu unterstützen.

Dieses Framework sollte dazu verwendet werden, das Roboter-Mikrokern-Betriebssystem M.O.N.S.T.E.R., das für die Entwicklung von Systemverhaltensweisen verwendet wird, MATLAB/Simulink für die Entwicklung von Motor-Controllern und Adams/View für die Entwicklung der Kinematik und Dynamik des Roboters miteinander zu verbinden. Aufbau und Arbeitsweise einer passenden Umsetzung des Frameworks werden im Folgenden vorgestellt (siehe Abbildung 9). Bei dem hier vorgestellten Konzept sollte die mechanische Struktur der realen Komponente (oder sogar des gesamten Roboters) in Adams/View modelliert werden, wo auch ihre Kinematik und Dynamik simuliert werden. Referenzexperimente zu den Motor-Controllern und das Testen und Optimieren verschiedener Steuerungs-Strategien können in MATLAB/Simulink durchgeführt werden. Das Gesamtverhalten des Roboters (z.B. ein Laufverhalten) und die gewünschten Bewegungen seiner Gelenke können mit unserem selbst entwickelten Roboter-Mikrokern-Betriebssystem M.O.N.S.T.E.R. erzeugt werden, das über ein angepasstes Socket-Kommunikationsprotokoll, das Teil des Frameworks ist, an die Simulatoren gekoppelt wird.

Die Auswirkungen verschiedener Techniken zur Steuerung der Motoren des Roboters sowie die Effekte der Optimierung einiger Modellparameter können ebenfalls analysiert werden. Das Co-Simulations-Framework ermöglicht eine „benutzerfreundlichere“ Art

der Roboterentwicklung, weil jedes Mitglied des Roboter-Design-Teams mit seinem präferierten Werkzeug arbeiten kann und das Framework dann diese Werkzeuge so verbindet, dass sie interaktiv zusammenarbeiten können.

Die Interfaces der Co-Simulation sowohl zu MATLAB als auch zu Simulink verbinden sie mit unseren eigenen Simulatoren, um deren Fähigkeiten zu erweitern. Das Interface zu Simulink erlaubt eine bidirektionale Co-Simulation funktionaler Blöcke aus der Simulink-Komponentenbibliothek mit anderen Komponenten, die durch VEROSIM® oder Adams/View simuliert werden. Das Interface mit MATLAB erlaubt aus MONSTER-Quellcode heraus die Ausführung von MATLAB-Befehlen, den Aufruf von „m-functions“, oder die Übertragung von Daten zum oder aus dem MATLAB-Arbeitsbereich. Die Kommunikation mit MATLAB wird über eine bestimmte Menge von Unterprogrammen realisiert sowie über Simulink-Blöcke, die sowohl für VEROSIM® als auch für andere externe *Benutzer* Programme vorbereitet sind. Auf jeder Ebene einer Design-Hierarchie können Befehle an MATLAB gesendet werden, um z.B. Design-Variablen an den MATLAB-Arbeitsbereich zu senden (oder sogar um einen Ausdruck zu senden, der ausgewertet werden soll, oder eine „m-function“ aufzurufen), nötige Operationen auszuführen und die Ergebnisse zurück zum Simulator zu übertragen.

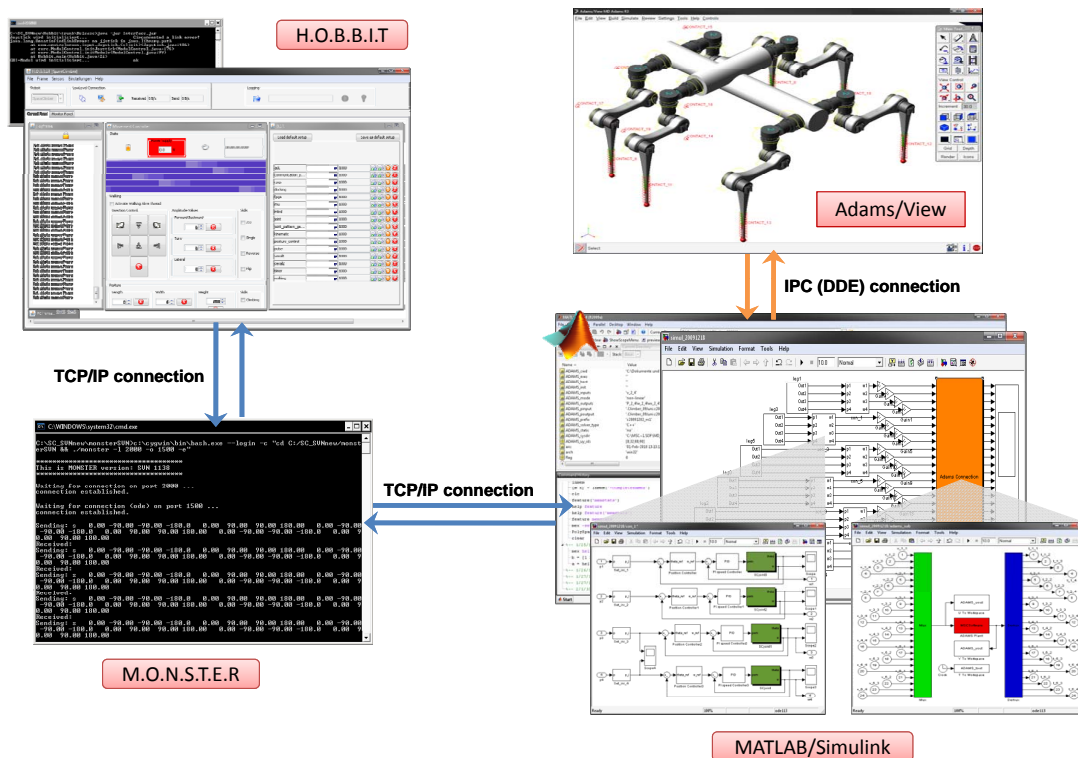


Abbildung 9: Das Co-SimFramework mit HOBBIT, MONSTER, MATLAB/SIMULINK, und ADAM/VIEW

Die Komponenten dieses Frameworks sind:

M.O.N.S.T.E.R Ein selbst entwickeltes Roboter-Mikrokernel-Betriebssystem, das in



realen Robotern eingesetzt wird. Es verbindet Eigenschaften von Echtzeit-Betriebssystemen mit verhaltensbasierter Programmierung, um Probleme der Skalierbarkeit und Reaktivität zu lösen.

HOBBIT Eine Front-End Benutzeroberfläche für M.O.N.S.T.E.R.

MATLAB/Simulink MathWorks „Technical Computing“-Umgebung mit intuitiver Sprache, die verwendet wird, um Systemkomponenten und Steuerungsstrategien zu entwickeln und zu testen.

Adams/View Ein Mehrkörper-Dynamik-Simulator, der die Möglichkeit bietet, virtuelle Prototypen mechanischer Systeme zu erstellen und zu testen. Adams unterstützt die Simulation von Mehrkörpersystemen auf kinematischer und dynamischer Ebene, außerdem werden Simulationsmodelle für Statik und Quasi-Statik unterstützt.

Diese Komponenten werden in Abbildung 9 gezeigt.

Herleitung eines abstrakten Gelenkmodells

Ein exaktes und detailliertes Systemmodell für das SpaceClimber-Gelenk wird nach bekannten physikalischen Gesetzen hergeleitet und in der Simulationsanalyse der elektrischen und mechanischen Komponenten verwendet. Dieses detaillierte Modell ist übermäßig komplex, so dass es nicht möglich war, es in weiteren Simulationen in vernünftiger Simulationszeit zu verwenden, aufgrund der komplexen Natur der Gelenkkomponenten und wegen der unbekanntem Bedingungen wie Sättigungseffekten, Parameter-Drift und Rauschen, die nicht vermeidbar sind und nicht exakt mathematisch beschrieben werden können. Daher ist es nötig, ein einfacheres Modell für das Gelenk zu entwickeln, das genau genug ist, um in weiteren Simulationen und Analysen verwendet zu werden.

Beginnend beim detaillierten Modell, wird ein vereinfachtes abstraktes Modell in Form einer Transferfunktion folgendermaßen hergeleitet:

$$G(s) = \frac{1/K_e}{\tau_m \tau_e s^2 + \tau_m s + 1} \quad (1)$$

wobei τ_m , τ_e , und K_e jeweils die mechanische Zeitkonstante, die elektrische Zeitkonstante und die „actuator back EMF“-Konstante sind. Für das SpaceClimber-Gelenk wird dieses abstrakte Modell numerisch gegeben als:

$$G(s) = \frac{13.33}{5.27e - 7s^2 + 4.486e - 4s + 1} \quad (2)$$

Um dieses abstrakte Modell zu verifizieren, wurde das entwickelte Co-Simulations-Framework verwendet, um das echte Gelenk mit MATLAB/Simulink zu verbinden, und zwar über den Simulink-Interface-Block (gezeigt in Abb. 11) und die grafische Benutzeroberfläche der Überwachungs- und Kontrollsoftware (entwickelt in Qt/C++) des echten Gelenks in einem *Hardware in the Loop*-Aufbau, der in Abb. 10 gezeigt wird.

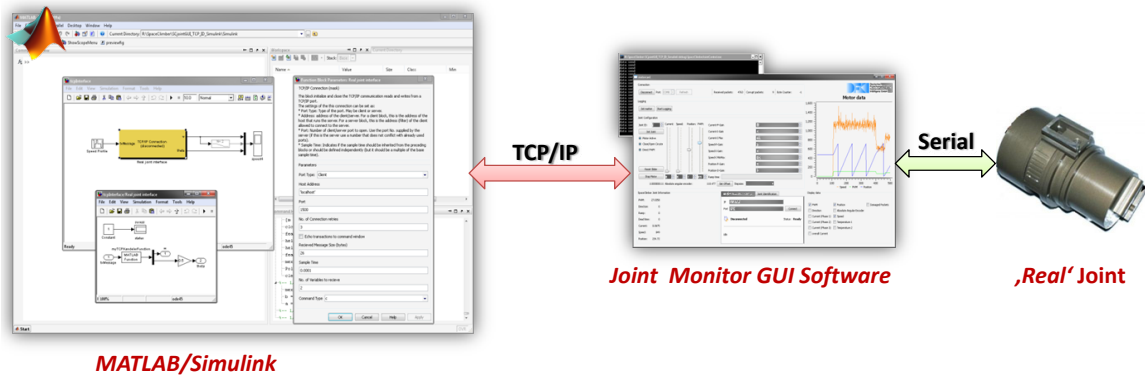


Abbildung 10: Hardware in the loop (HIL)

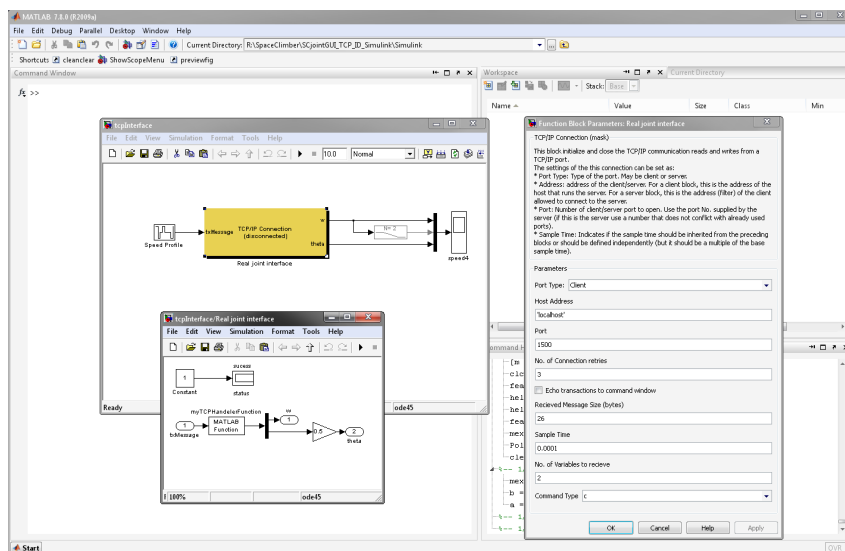


Abbildung 11: The developed MATLAB/Simulink TCP/IP interface block

In diesem Aufbau werden die Modelle und das echte Gelenk über den gleichen Geschwindigkeits-Profil-Block gesteuert und ihre Ausgabewerte werden über den „Scope-Block“ dargestellt. Abbildung 12 zeigt die Ergebnisse der jeweiligen Vergleiche zwischen den verschiedenen verwendeten Modellen und den Resultaten des echten Gelenks. Das beste abstrakte Modell erzeugt Ausgabewerte, die mit einer Genauig-

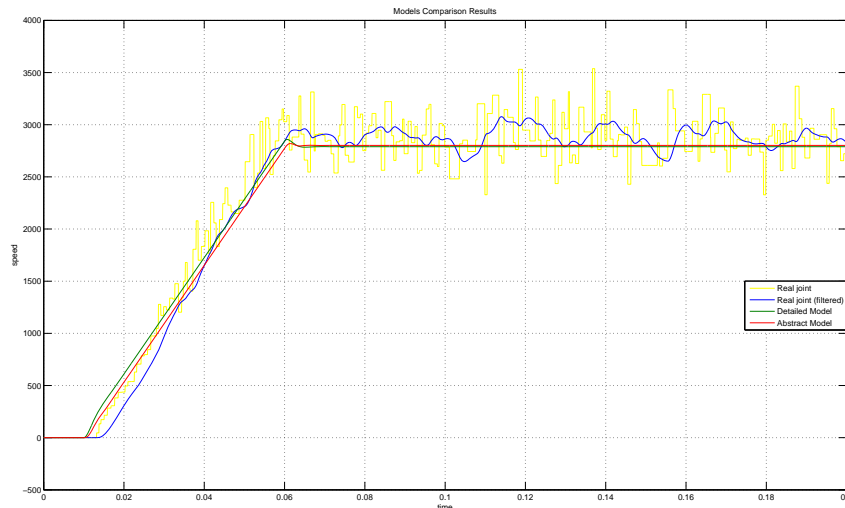


Abbildung 12: Resultate der Vergleiche des Referenzexperiments mit dem detaillierten und den abstrakten Modellen.

keit von 97,9% den Ausgabewerten des echten Gelenks entsprechen (mit einer durchschnittlichen relativen Abweichung von ungefähr 2,1%). Mit diesen Ergebnissen ist das abstrakte Modell des SpaceClimber-Gelenks ein schnelles und genaues Modell, das in Regelkreisen und Simulationen verwendet werden kann.



3.1.12 AP3300: Definition anwendungsspezifischer Erweiterungen des Simulationssystems

3.1.13 AP3310: Darstellung

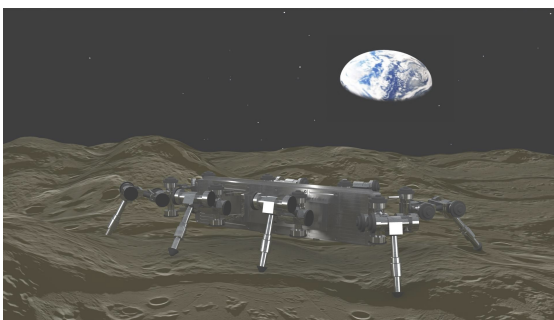
Im Rahmen dieses Arbeitspaketes werden visuelle Ergänzungen für VEROSIM® definiert, welche das intuitive Verständnis bei der Durchführung einer Simulation verbessern sollen. Zusätzlich werden Anforderungen an eine realistische Visualisierung einer extraterrestrischen, lunaren Umgebung gesammelt.

Ergänzungen VEROSIM®

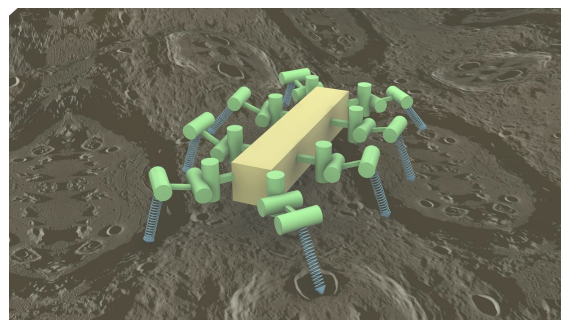
Für die nachfolgend vorgeschlagenen Verbesserungen gilt generell, dass sie wahlweise ein- und ausgeblendet werden können. Alle zusätzlichen Fenster müssen beliebig skaliert, verschoben und auf der Arbeitsfläche platziert werden können.

Reales vs. Physikalisches Modell

Hierbei ist es möglich, zwischen dem realen und dem eigentlichen physikalischen Modell umzuschalten. Dies erlaubt eine Vorführung oder Video Aufnahme, während das reale Modell aktiv ist. Die Visualisierung des physikalischen Modells ermöglicht ein besseres Verständnis des Verhalten des simulierten Roboters. Außerdem vereinfacht es das Finden von Fehlern im physikalischen Modell und damit die Optimierung der Simulation.



(a) Reales Modell



(b) Physikalisches Modell

Abbildung 13: Unterschiedliche Darstellungs-Modi der Roboter

Transparenz

Es ist möglich, die Transparenz der simulierten Komponenten beliebig zu verändern. Dadurch wird es möglich, eine größere Anzahl an Einzelkomponenten gleichzeitig im Blick zu behalten.



Ghost-Modus

Im Ghost-Modus wird eine semi-transparente Version des Roboters aus einem vorherigen Lauf eingeblendet, was den visuellen Vergleich zweier Läufe ermöglicht. Um dies zu realisieren, können die Resultate aus den Läufen abgespeichert werden. Hierbei handelt es sich nicht um eine Simulation mit alten Parametern sondern um eine Aufnahme der Simulationsergebnissen. Im Ghost-Modus kann entweder der jeweils vorherige Lauf oder ein durch den Benutzer angegebener vorheriger Lauf dargestellt werden. Um weiterhin zwischen dem Ghost und dem Original-Roboter zu unterscheiden, ist es im Ghost-Modus nicht mehr möglich, die Transparenz des simulierten Roboters zu verändern. Außerdem soll es möglich sein, die Messergebnisse des Tracking-Systems als Ghost-Roboter darzustellen. Dazu müssen zum Beispiel die Lage des Körpers oder die Endposition der Füße des Ghost-Roboters anhand einer ASCII-Datei gesetzt werden können.

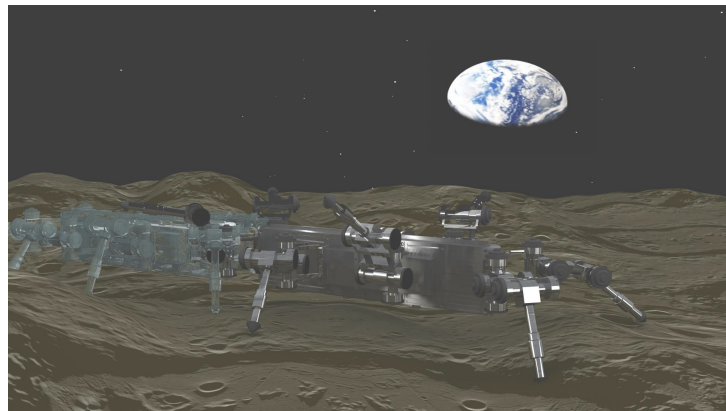


Abbildung 14: Visualisierung des vorherigen Durchlaufs

Karten Darstellung

Während der Steuerung des Roboters in der Simulation kann eine Übersichtskarte der Umgebung eingeblendet werden, was die gezielte Steuerung des Roboters vereinfacht. Dies kann wahlweise eine 2D Ansicht sein, welche die Ansicht einer direkt über den Roboter platzierten Kamera entspricht, und eine 3D Ansicht, die den Roboter aus einer Vogelperspektive zeigt. In beiden Fällen bleibt die Ansicht während der Simulation automatisch auf dem Roboter zentriert. Die Karte bietet außerdem die Möglichkeit, verschiedene Zoom-Stufen einzustellen.

Kraftvektoren

Die Kraftvektoren können dazu genutzt werden, Kräfte zu visualisieren, die zwischen zwei Komponenten und innerhalb von Motoren und Gelenken auftreten. Die Darstellung der Wechselwirkung zweier Komponenten erlaubt zum Beispiel die Kräfte zu visualisieren, die beim Auftreten des Roboters entstehen. Hierbei werden an den Berührungspunkten des Beines die Kraftvektoren entsprechend der Normalen der Berührungsfläche eingezeichnet. Bei Motoren und Gelenken werden die Kraftvektoren

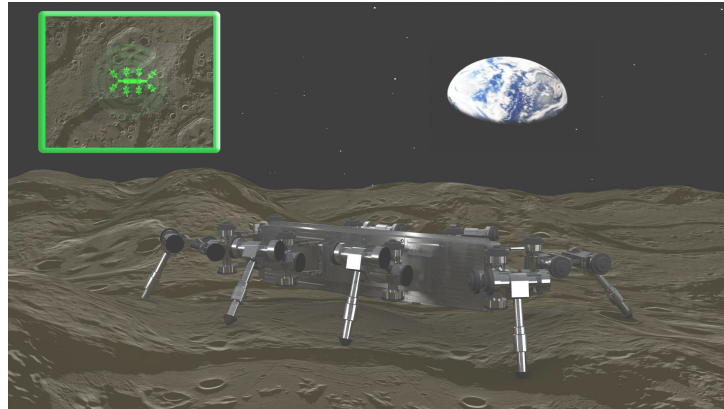


Abbildung 15: Einblenden einer Übersichtskarte

entsprechen der Drehachse eingezeichnet. Die Farbe der Vektoren kann beliebig definiert werden.

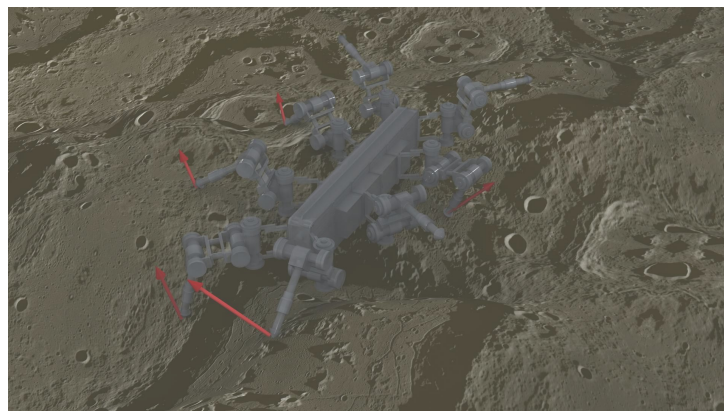


Abbildung 16: Visualisierung der Kontakt-Kräfte

Belastung

Hierdurch ist es möglich, einzelne Komponenten des Roboters entsprechend frei wählbarer Parameter einfärben zu lassen. So können Strukturelemente entsprechend ihrer Belastung oder Motoren entsprechend ihrer Stromaufnahme eingefärbt werden. Die Farbskala reicht dabei von 400nm bis 700nm. Auf diese Weise können sonst nicht sichtbare Vorgänge visualisiert und eine zu große Belastung des realen Roboters vermieden werden.

Zahlen

Durch diese Ergänzung (siehe Abbildung 18) ist es möglich, interne Parameter des Roboters durch eine visualisierte Zahl anzeigen zu lassen. Diese Zahl ist durch eine sichtbare Linie mit der Komponente verbunden, kann beliebig verschoben und skaliert werden und behält im Laufe der Simulation ihre relative Position zum Roboter.

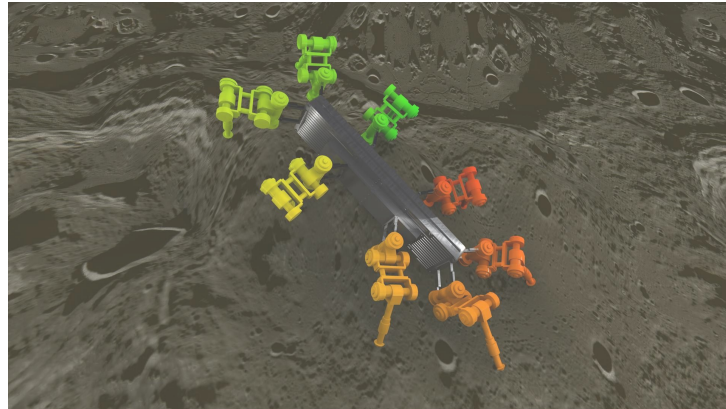


Abbildung 17: Visualisierung der Belastung der Beine

Zu den Parametern gehören Werte wie Struktur-Belastung, Stromverbrauch, Drehzahl und Sensorwerte. Es ist außerdem möglich, den durchschnittlichen Wert eines Parameters berechnen und anzeigen zu lassen. Zu jeder Zahl kann außerdem eine Beschriftung hinzugefügt werden.

Diagramme

Durch diesen Zusatz können die in Abschnitt 3.1.13 beschriebenen Zahlen in Form eines Diagramms (Balkendiagramm, Plot) visualisiert werden. Hierbei wird nicht nur der aktuellste Wert angezeigt, sondern eine Historie über alle vergangenen Werte abgespeichert. In dem Diagramm können einzelne Parameter angezeigt und mehrere kombiniert werden. Sofern die Diagramme zur Visualisierung einzelner Parameter benutzt werden, behalten sie wie die Zahlen ihre relative Position zum Roboter und sind durch eine visualisierte Linie mit der Komponente verbunden. Werden mehrere Parameter in einem Diagramm kombiniert, wird es vom Roboter losgelöst und kann beliebig auf der Arbeitsfläche platziert werden. Die Diagramme können wie die Zahlen skaliert werden.

Visualisierungs-Anforderungen

In diesem Abschnitt werden die Anforderungen an die Visualisierung einer lunaren Umgebung zusammengefasst. Die Oberflächenbeschaffenheit und die Lichtverhältnisse müssen möglichst originalgetreu in der Simulation nachgebildet werden, so dass die Robotersysteme ausführlich getestet und optimiert werden können.

Oberflächenbeschaffenheit

Der Mond ist, resultierend aus Meteoriteneinschlägen, mit unterschiedlich klassifizierten Kratern, einer meterhohen Schicht Regolith und Felsen bedeckt (siehe Abbildung 19). Die Felsen sind dabei je nach Alter unterschiedlich stark durch Regolith bedeckt. Das führt dazu, dass Felsschichten direkt unter einer feinen Regolith-Schicht liegen

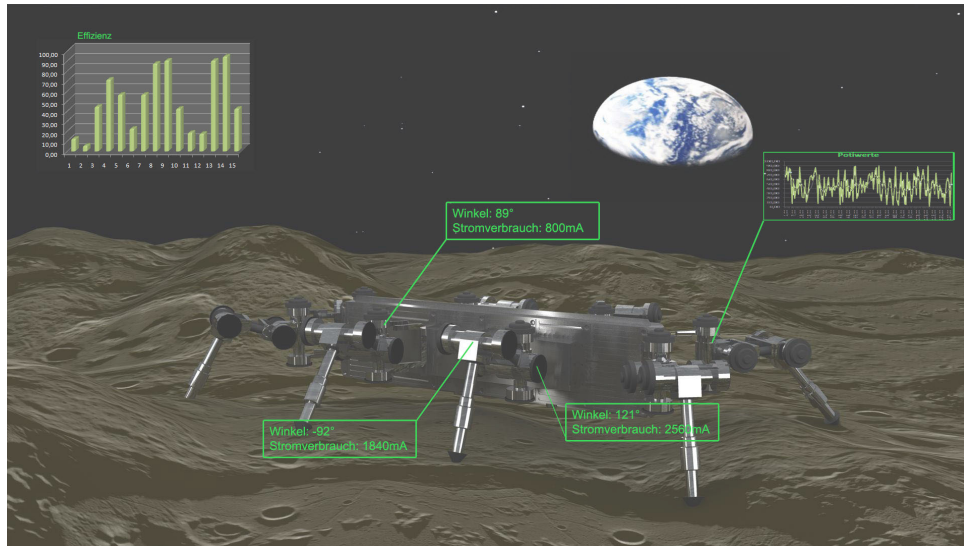


Abbildung 18: Visualisierung der internen Parameter

können. Dadurch können sich weiche Regolith-Untergründe übergangslos mit harten Steinuntergründen abwechseln. Die verwendeten Texturen beziehungsweise deren Färbungen sollten der Dichte des Untergrunds entsprechen. Dadurch erhält der Benutzer eine direkte, visuelle Beschreibung über die Art des Untergrundes, auf der sich der Roboter zum aktuellen Zeitpunkt fortbewegt. Während diese Erweiterung optional ist, musste VEROSIM[®] dazu in die Lage versetzt werden, RGB-Höhenbilder einzulesen und daraus die Kraterumgebungen zu generieren. Die Höhenbilder wurden aus Laserscans des am DFKI modellierten Kraters erzeugt. Da die modellierten Krater bereits mit Felsen bedeckt sind, müssen die Felsen nicht extra mit VEROSIM[®] erzeugt werden können.

Lichtverhältnisse

Da das Licht der Sonne durch keine Atmosphäre gefiltert wird, treten auf dem Mond innerhalb von Kratern und an Felsblöcken stark abgegrenzte Schatten auf (siehe Abbildung 20). Für die Validierung der Bildverarbeitungsalgorithmen ist die Berücksichtigung dieser Lichtverhältnisse sehr wichtig. Obwohl von der hellen und dunklen Seite des Mondes gesprochen wird, werden beide Seiten während eines lunaren Tages gleich lange durch die Sonne beschienen. Aufgrund der stabilen Mondachse werden der Nord- und Südpol permanent vom Sonnenlicht gestreift, was an den Rändern der in diesem Bereichen liegenden Krater gesehen werden kann. Der Boden dieser Krater liegen dagegen in ewiger Dunkelheit. Es muss in VEROSIM[®] möglich sein, Lichtquellen frei in der Szene zu platzieren und deren Parameter (Helligkeit und Emissionswinkel) einzustellen. Insgesamt muss ein weißes Licht, mit parallelen Strahlenwurf und scharfen Schatten simuliert werden können.

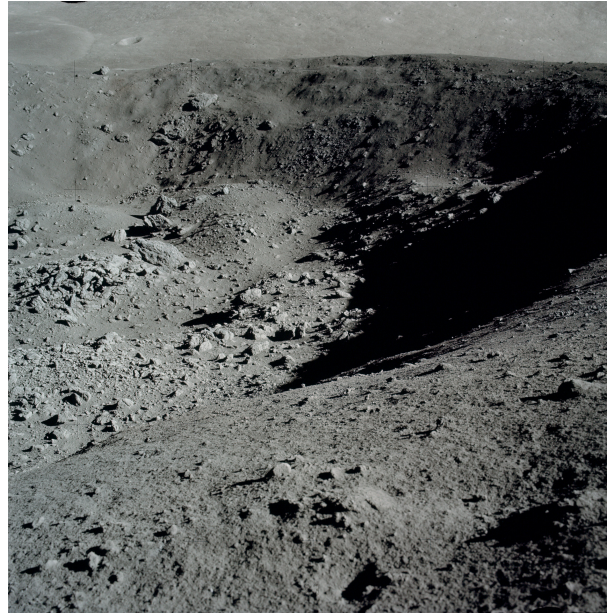


Abbildung 19: Mondstruktur



Abbildung 20: Lichtverhaeltnisse auf dem Mond

3.1.14 AP3320: Dynamiksimulation

Entwicklung von Verfahren zur Berücksichtigung von bodenmechanischen Effekten im Gesamtkonzept der Dynamiksimulation sind definiert in diesem AP.

Wenn sich zwei starre Körper in der Dynamiksimulation berühren, zeigen sie keine Deformationen, sondern durchdringen sich oder stoßen einander ab. Daher muss eine Verbindung zwischen der normalen Starrkörperdynamik und der Bodendynamik in VEROSIM[®] definiert werden. Im Folgenden wird eine entsprechende zusätzliche Dynamik zur Starrkörpersimulation in VEROSIM[®] definiert.

- Die Veränderung der Polygonnetzes der Boden-Oberfläche, die durch die Kontakte des Roboters mit dem Boden herbeigeführt wurde. Gegeben sei ein Fuß-Boden-Kontaktmodell wie in Abb. 21(a). Die Boden-Oberfläche sollte wie bei einer plastischen Verformung verändert werden, so dass das Polygon am Boden wie in Abb. 21(b) verändert wird.
- Berechnung der Normalen-Kräfte an der Kontakt-Fläche.
- Integration der Normalen- und Reibungskräfte auf der Oberfläche, wenn der Fuß und die Oberfläche komplexe Polygon-Modelle sind.
- Kinematische Informationen über die beiden Geometrien, die in Kontakt stehen: Tiefe – Der Abstand zwischen den Kontaktpunkten, Geschwindigkeit – Die Eindringgeschwindigkeit am Kontaktpunkt (Dies ist die relative Geschwindigkeit entlang der gemeinsamen normalen am Kontaktpunkt), und die Bohrgeschwindigkeit – Dies ist die Winkelgeschwindigkeit um die gemeinsame Normale der beiden Geometrien am Kontaktpunkt.

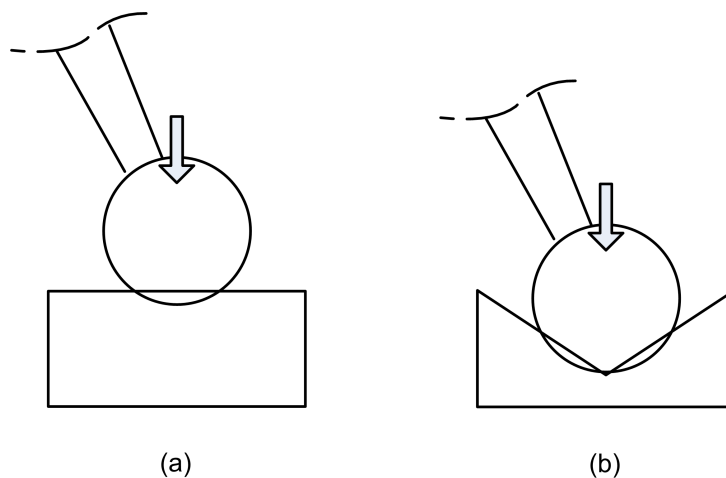


Abbildung 21: Verformung für Fuß-Boden-Kontakt in einer Starrkörpersimulation



3.1.15 AP3330: Sensorsimulation

Entsprechend den Sensor-Anforderungen aus AP2300 wird ein Sensor-Framework benötigt, welches eine komfortable und genaue Umsetzung der unterschiedlichen Sensoren ermöglicht. Die umzusetzenden Sensoren sind in Abschnitt 3.1.15 zusammengefasst, während das benötigte Framework in Abschnitt 3.1.15 behandelt wird.

Sensoren

Dieser Abschnitt beinhaltet die zu simulierenden Sensoren und eine kurze Beschreibung, wie die Umsetzung realisiert werden kann. Die Auswahl beinhaltet die relevanten Sensoren des Spaceclimbers und alle Sensoren, die am DFKI verbreitet eingesetzt werden. Sensoren, die nur mit großem Aufwand simuliert werden könnten, wie Strom-, Temperatur- und Ultraschallsensoren, wurden von der Auswahl ausgeschlossen. Die Tabelle 8 fasst die benötigten Sensoren und Werkzeuge zusammen.

Inertialsensoren Um die Beschleunigungs- und Rotationskräfte auf allen sechs Achsen in der Simulation zu überwachen, wird eine simulierte IMU benötigt. Sie kann realisiert werden, indem das Abfragen der Position und Orientierung ihres lokalen Koordinatensystems ermöglicht wird. Das Rauschen entlang der sechs Freiheitsgrade sollte dabei einzeln definiert werden können. Auf diese Weise kann mit einer einzelnen Komponente das gesamte Spektrum der Inertialsensoren abgedeckt werden. In der Simulation wird zunächst der LIS3LV02DL, die DFKI-IMU und eine Perfekte IMU für Referenzwerte umgesetzt.

Winkelsensoren In der Simulation sollte es ermöglicht werden, die Winkel der einzelnen Gelenke abzufragen. Dabei muss für jeden Winkelsensor der Winkelbereich und eine Abbildung definiert werden können, welche den berechneten Winkel in die Ausgabeinheit des simulierten Sensors (Schritte, Spannung) überführt. Die Abbildung sollte dabei als Funktion oder Graph hinterlegt werden können. Die Definition des Graphen wiederum erfolgt anhand von Referenzwerten, aus denen eine B-Spline-Kurve erstellt wird. Auf die berechneten Ausgabewerte wird anschließend das sensorspezifische Rauschen aufaddiert. Für die Simulation wird zunächst das Gelenk des Spaceclimbers definiert, welches 720 Grad g auf 12000 Schritte s abbildet. In diesem Fall läge der Winkelbereich von 0 bis 720° und die Abbildungsfunktion wäre

$$s = g \cdot 16.\bar{6} + r \quad (3)$$

, wobei r das aktuelle Rauschen repräsentiert.

Das Rauschen des realen Spaceclimber-Gelenks kann mit einem vorhandenen Testprogramm sehr genau detektiert und aufgezeichnet werden. Das Verhalten des Sensors in den Grenzbereichen des Drehbereichs muss besonders sorgfältig simuliert werden. Während es bei Potentiometern zu einem Übersprung kommt, erlauben Inkrementgeber theoretisch auch außerhalb des Wertebereichs ein Nachregeln. So wird



Typ	Sensor	Benötigt	Bemerkung
3-Achsen Beschleunigungssensor	LIS3LV02DL	Abfragen der Position und Orientierung des lokalen Koordinatensystems der IMU, unabhängige Rausch-Definition für alle sechs Freiheitsgrade, Samplingrate	Verbaut im Spaceclimber-Fuß.
IMU	DFKI-IMU		Beinhaltet reales Rauschen, Störanfällig aufgrund der Magnetfeldsensoren.
	Perfekte IMU		Rauschfrei. Kann zum Abfragen von Referenzwerten verwendet werden.
Winkelsensor	Spaceclimber-Gelenk	Definition des Winkelbereichs, Definition der Ausgabe über Funktionen oder Graphen (B-Spline-Kurve), Rauschen, Samplingrate	B-Spline-Kurve wird anhand von Referenzwerten definiert.
Zug-Druck-Kraftsensoren	ATI MINI45	Rauschen, Samplingrate	Kraftmessung auf sechs Achsen.
Drucksensor	MS5541C im Spaceclimber-Fuß	Ausgabewert in Abhängigkeit von Auftrittskraft und -winkel (Funktion mit zwei Eingabeparameter), Abfragen der Untergrundbeschaffenheit, Rauschen, Samplingrate	Sensor wird zusammen mit der verbauten Komponente in die Simulation aufgenommen.
GPS	Magellan MB500	Abfragen der Position des lokalen Koordinatensystems, Rauschen, Samplingrate	Anzahl der Satelliten zum Zeitpunkt der Referenzexperimente kann nicht berücksichtigt werden.
Kamera	VRmC3+ PRO	Interne Parameter (Brennweite, Verzerrung, Auflösung), Framerate	Stereokamerasystem des Spaceclimbers.
Laserscanner	Hokuyo URG-04LX	Scanbereich, Abfragen und berücksichtigen der Untergrundfarbe, Rauschen in Abhängigkeit von der gemessenen Entfernung, Samplingrate	Verbreitet eingesetzter Abstandssensor.
PMD-Kamera	PMD [vision] O3	Rauschen, Samplingrate	Es muss in den Referenzexperimenten festgestellt werden, ob die gemessenen Entfernungen von der Untergrundfarbe beeinflusst werden.

Tabelle 8: Definition benötigter Sensoren und Werkzeuge



das Spaceclimber-Gelenk zurück auf 12000 geregelt, sobald der Grenzwert überschritten wird. Fällt der Wert jedoch unter 0 kommt es zu einem Überlauf der intern genutzten Container-Variable. Als Folge dessen, wird versucht die Position 65535 anzusteuern, wodurch das Gelenk beschädigt werden kann. Ein derartiges Verhalten muss ebenfalls in der Simulationsumgebung berücksichtigt werden können.

Zug-Druck-Kraftsensoren Die Simulation sollte es ermöglichen, die robotischen Systeme mit ringförmigen Zug-Druck-Kraftsensoren auszustatten, welche die translatorischen und rotatorischen Kräfte auf sechs Achsen messen. Zusätzlich sollte es möglich sein, den berechneten Kräfte ein leichtes Rauschen hinzuzufügen. Als Referenzsensor wird der ATI MINI45 dienen.

Drucksensoren Das Verhalten der Drucksensoren hängt stark von der Art und Weise ab, wie sie im Roboter verbaut wurden. Der im Spaceclimber-Fuß eingesetzte Drucksensor MS5541C wurde in Polyurethan eingegossen, wodurch seine Messwerte von der Dicke und Härte des verwendeten Kunstharzes abhängen. Aus diesem Grund sollte dieser Drucksensor in Verbindung mit der verbauten Komponente getestet und in die Simulation aufgenommen werden.

GPS In der Robotik eingesetzte IMUs eignen sich aufgrund der sich aufsummierenden Ungenauigkeiten generell nicht zur Positionsbestimmung. Aus diesem Grund wird für die Outdoor-Navigation GPS eingesetzt. Dies kann in der Simulation realisiert werden, indem die Position des lokalen Koordinatensystems der IMU-Komponente abgefragt werden kann. Als Referenzsensor wird der Magellan MB500 verwendet werden.

Kameras Als Kameras werden die im Spaceclimber eingesetzten VRmC3+ PRO anhand der internen Kameraparameter simuliert. Zusätzlich sollte es möglich sein, sowohl Farb- als auch S/W-Kameras zu simulieren. Durch das Bereitstellen einer Kamera-Komponente können sowohl Einzelkameras als auch Stereokamera-Systeme simuliert werden. Ein direkter, nicht rein visueller Vergleich zwischen den simulierten und realen Kamerabildern ist nicht ohne weiteres möglich. Das gerenderte Bild basiert auf den verwendeten internen Kameraparametern, den im Kamerabild dargestellten Gegenständen und deren Texturen und den Lichtverhältnissen. Ein qualifizierter Vergleich, zum Beispiel durch Verwendung von Differenzbildern, wäre also sinnlos, da die Ursache für die Differenz undefiniert bleibt.

Laserscanner Als Abstandssensor soll der am DFKI verbreitet eingesetzte Hokuyo URG-04LX simuliert werden. Trotz seiner Anfälligkeit gegenüber farbigen und schwarzen Untergründen, kann er in der Simulation - im Vergleich zu den anderen Abstandssensoren - am realistischsten nachgebildet werden. Sein Verhalten bei unterschiedlichen Untergrundfarben wird in den Referenzexperimenten bestimmt werden. Dagegen



wird der Einfluss unterschiedlicher Materialien aufgrund der Komplexität nicht berücksichtigt werden. Der Laserscanner wird über eine Strahlenverfolgung realisiert, wobei der Scanner entsprechend einer Bereichsdefinition eine Liste an Entfernungswerten zurückliefert. Der Laserscanner kann Entfernungen zwischen 20 bis 4094 mm messen. Liegen die zurückgelieferten Entfernungen unterhalb 20 mm, beschreiben die Werte einen aufgetretenen Fehler.

PMD-Kameras Als PMD-Kamera soll die PMD [vision] O3 in der Simulation umgesetzt werden. Dies entspricht für jeden Pixel der Kamera einer Entfernungsmessung, also 64 x 48 gleichzeitigen Entfernungsmessungen. Hierfür kann wie bei der Simulation eines Laserscanners Strahlenverfolgung eingesetzt werden. Anschließend wird auf den einzelnen Entfernungswerten das sensorspezifische Rauschen aufaddiert. Es muss in den Referenzexperimenten ermittelt werden, ob beziehungsweise in welchem Maße die gemessenen Entfernungen von den Untergrundfarben beeinflusst werden.

Sensorframework

Das Sensorframework soll das komfortable Definieren und nachträgliche Modifizieren der simulierten Sensoren ermöglichen. Bei der Definition eines Sensors müssen zuerst die generellen Informationen wie Name, Sensortyp, Stärke und Art des Rauschens und maximale Samplingrate festgelegt werden. Die weiteren anzugebenden Daten richten sich nach dem gewählten Typ des Sensors. So muss bei einem *Winkelsensor* der Winkelbereich festgelegt werden und die Art und Weise, wie der Winkel in die Ausgabeeinheit des Sensors umgerechnet werden soll. Die Konvertierung des Eingabeparameters (der Eingabeparameter) kann durch die Definition einer Funktion, einer Skript-Sprache oder eines Graphen erfolgen. Abbildung 22 zeigt ein Beispiel, wie die GUI des Sensorframeworks aussehen könnte. Die nachfolgende Auflistung umfasst die möglichen Parameter und ihre Bedeutung.

Samplingrate Die Samplingrate gibt die maximal mögliche Anzahl an Sensorwert-Abfragen pro Sekunde an.

Rauschen Zur Definition des Rauschens sollte lineares und gaußsches Rauschen bereitgestellt werden. Außerdem sollte es möglich sein, das Rauschen sowohl auf Basis eines festen Grenzbetrages als auch auf Basis eines prozentualen Betrages des aktuellen Messergebnisses zu definieren. Auf diese Weise kann ein von den Messwerten abhängiges, wie es generell bei Abstandssensoren auftritt, und unabhängiges Rauschen generiert werden.

Konvertierung Anhand der Konvertierung werden die perfekten Werte aus der Simulation in die Ausgabeeinheit des jeweiligen Sensors konvertiert. Die Anzahl der Eingabeparameter unterscheidet sich je nach Typ des simulierten Sensors. Bei einzelnen Eingabeparametern, wie sie durch den Winkelsensor geliefert werden, kann die Konvertierung anhand eines Graphen erfolgen. Der Graph kann über das Setzen und Laden von Referenzpunkten, durch die eine B-Spline-Kurve gezeichnet wird, definiert werden. Bei mehreren Eingabeparametern, wie sie der

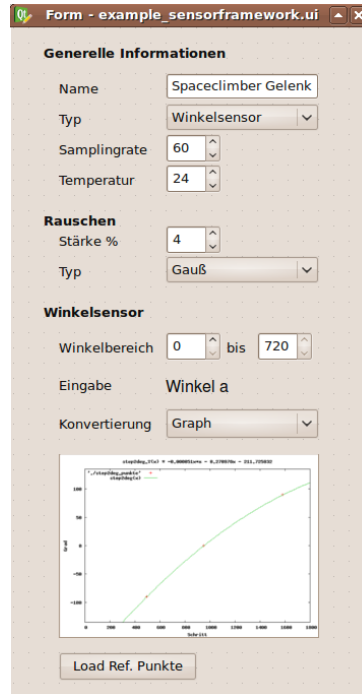


Abbildung 22: Beispiel Sensorframework

Drucksensor des Spaceclimber-Fußes liefert (Auftrittswinkel und -kraft), kann die Konvertierung durch die Definition einer Funktion oder über eine Skriptsprache erfolgen, in der die Eingabeparameter verwendet werden. Eine Skriptsprache erlaubt eine komfortablere und umfangreichere Konvertierung als es Funktionen ermöglichen und das Umsetzen besonderer Eigenarten, so wie sie beispielsweise beim Spaceclimber-Gelenk auftreten, sobald der untere Grenzbereich überschritten wird.

3.1.16 AP3340: Simulation der Antriebe

Das SpaceClimber-Gelenk (gezeigt in Abb. 23) ist ein leichter Motor mit integrierter Elektronik und Intelligenz, der hohe Drehmomente und eine breite Palette möglicher Bewegungen für das Bein ermöglicht. Außerdem ist dieses Gelenk für weitere Weltraum-Anwendungen geeignet.

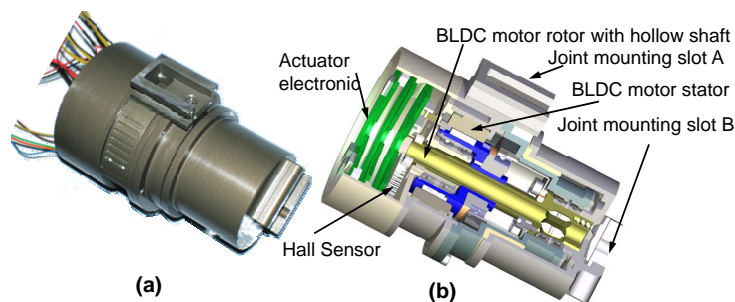


Abbildung 23: SpaceClimber Joint, (a) Der echte Motor. (b) Detaillierter innerer Aufbau des Motors.

Es wurde ein korrektes und detailliertes Modell des SpaceClimber Gelenkmotors anhand bekannter physikalischer Gesetze entwickelt (siehe [1]). Dieses Modell wird bei Simulationen zur Analyse der elektrischen und mechanischen Komponenten, zur Simulation des Gelenkverhaltens und auch zur Entwicklung seiner Controller eingesetzt. Dieses detaillierte Modell ist unnötig komplex, aufgrund der Komplexität der Gelenk-Komponenten und durch andere Effekte wie Sättigungseffekte, Parameter-Drift und Rauschen, welche sich nicht vermeiden lassen. Daher war es nicht möglich, dieses Modell in weiteren Simulationen zu nutzen, ohne dass diese zu langsam wurden. Daher wurde ein vereinfachtes Modell entwickelt, das genau genug ist, um in weiteren Simulationen und Analysen verwendet zu werden.

Das detaillierte (und komplexe) Modell des SpaceClimber-Gelenks wurde weiter vereinfacht, um ein *abstraktes* Modell des Motors zu erhalten. Dieses abstrakte Modell hat eine allgemeine Form und kann zur Modellierung weiterer Arten von DC-Motoren verwendet werden:

$$G(s) = \frac{a_1 s + a_2}{s^2 + b_1 s + b_2} \quad (4)$$

dabei sind a_1 , a_2 , b_1 , and b_2 Konstanten des Modells, die von der Struktur des Gelenks und den elektrischen sowie mechanischen Komponenten abhängen.

Das abstrakte Modell wurde über Simulink-Blöcke wie in Abb. 24 gezeigt mit seinen low-level Positions- und Geschwindigkeits-Controllern konstruiert.

In Abb. 24 sind V_b , i_m , ω und P jeweils die EMK-Spannung, die Gesamtstromstärke im Motor, die Winkelgeschwindigkeit und die Position. Die Motor-Konstanten im elektrischen Teil sind die EMK-Konstante K_b , K_m ist die Drehmoment-Konstante, R und L sind die Induktivität und der Widerstand des Läufers. Das Motor-Drehmoment ist T_m . Die mechanischen Konstanten sind die Trägheit J der Reibungskoeffizient B . Das Stör-Drehmoment durch eine Last-Interaktion mit dem Motor ist T_L .

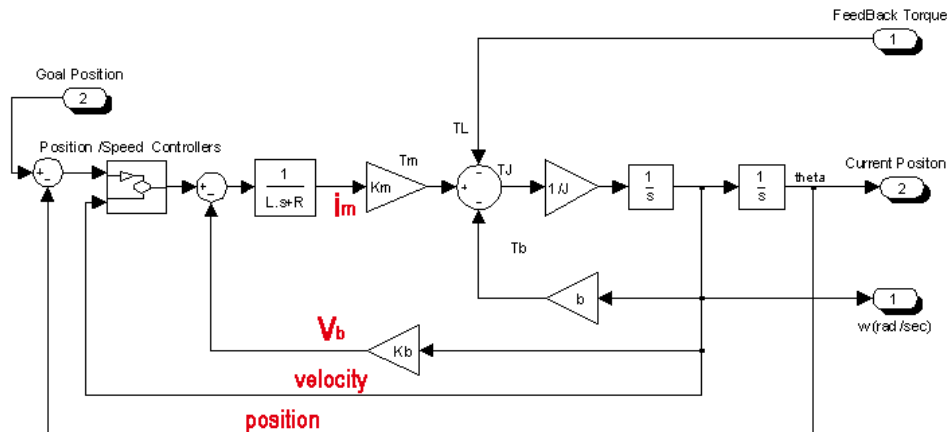


Abbildung 24: Abstraktes Modell des Gelenkmotors

Controller

Bei Anwendungen in der Weltraumrobotik sind schnelle Reaktionen, Unempfindlichkeit gegenüber Parameter-Veränderungen und schnelle Wiederherstellung der Leistungsfähigkeit nach Lastfällen von entscheidender Bedeutung. Folglich sind die richtige Auswahl und Optimierung der Controller-Parameter erforderlich, um erhebliche Überschreitung und Schwingungen in Anwendungen zu vermeiden, bei denen eine präzise Kontrolle wichtig ist. Eine Feineinstellung dieser Controller benötigt korrekte Modelle des elektromechanischen Systems. Darüber hinaus ist eine genaue Modellierung auch ein Thema in Machbarkeitsstudien, bei denen das Systemverhalten mittels Simulation vorhergesagt und bewertet wird.

PID Controller sind für die Steuerung von Motoren in der Robotik und auch in der Industrie weit verbreitet. Diese PID-Controller haben folgende allgemeine Übertragungsfunktion:

$$G_c(s) = k_p + \frac{k_i}{s} + k_d s \quad (5)$$

dabei sind k_p , k_i , und k_d jeweils Proportionalitäts-, Integral- und Ableitungs-Konstanten des Controllers.

Diese Übertragungsfunktion existiert in verschiedenen weiteren Formen, aber die gegebene Form in Gleichung 5 ist die allgemeinste und besser für Implementationen geeignet. Dieser PID wird auf der Ebene jedes Gelenkmotors einzeln eingestellt. Grundsätzlich liefert ein PD Controller Positions- und Geschwindigkeits-Feedback und hat gute Closed-Loop-Eigenschaften, wenn er auf ein Doppel-Integrator-System angewandt wird.

Der PID-Regler hat eine lange Geschichte, seit Ziegler and Nichols' PID-Einstellungsregeln 1942 veröffentlicht wurden. Im Grunde liegt der Hauptvorteil der PID-Controller in ihrer Einfachheit und ihrer klaren physikalischen Bedeutung. Eine ein-

fache Steuerung ist einer komplexen Steuerung vorzuziehen, zumindest in der Industrie, wenn die Leistungsverbesserung, die durch komplexere Controller erreicht wird, nicht signifikant genug ist.

Wenn die in Gleichung 5 gegebene Übertragungsfunktion implementiert wird, sind einige praktische Aspekte zu beachten, z.B. nur die Ableitungen der Prozess-Ausgaben zu berechnen, proportional nur auf den Teil des Fehlers zu reagieren und "Anti-Windup".

Das SpaceClimber-Gelenk wird über zwei kaskadierte Regelkreise gesteuert. Es gibt einen inneren Geschwindigkeits-Regelkreis für konstante Geschwindigkeiten mit einem PI-basierten Geschwindigkeits-Controller, der über einen LQR-Ansatz designed und in seinen Koeffizienten eingestellt wurde. Der äußere Regelkreis steuert die gewünschte Joint-Bewegung. Sein PD-basierter Controller wurde über Trial-and-Error designed und eingestellt.

Diese Regelkreise werden in **Abbildung 24** gezeigt.

Power Coupling

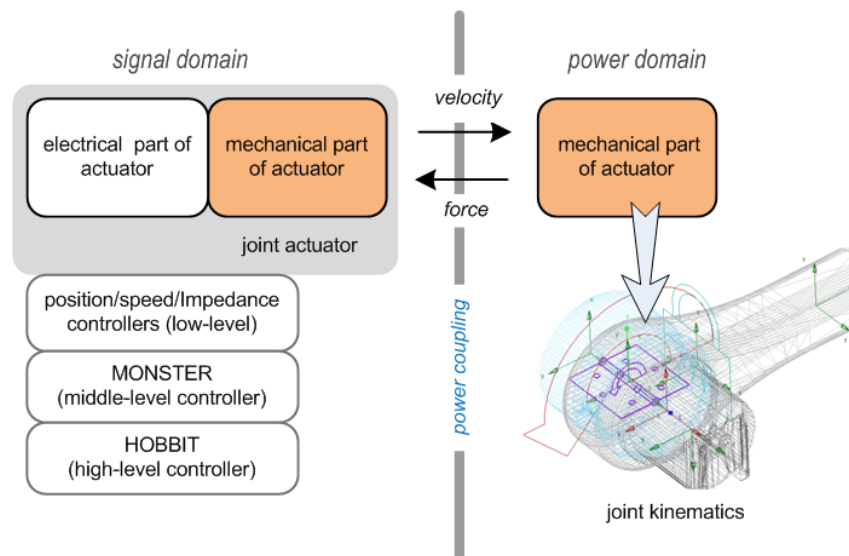


Abbildung 25: Szenarien einer bi-direktionalen Leistungskopplung zwischen dem Signalmodell und dem kinematischen Modell.

Im Allgemeinen haben Echtzeit-Physiksimulationen eine Zeitverzögerung zwischen Eingabe und Ausgabe. Eine solche Verzögerung führt oft dazu, dass das Simulationssystem instabil wird oder Eingaben verliert. Es seien die Interfaces zwischen den Gelenkmotor-Controllern und der Kinematik des Roboters betrachtet. Für gewöhnlich werden Controller in der Signaldomäne beschrieben, während Roboter-Kinematiken in der Leistungsdomäne ($velocity \times force$) beschrieben werden.

Speziell bei der Simulation von Robotern mit Beinen ist es wichtig, die Probleme von Zeitverzögerungen zwischen den Signalmodellen wie Gelenk-Controllern und den kinematischen Modellen wie Gelenk- und Bein-Dynamik. Betrachtet seien die Verbindungen, über die das Closed-Loop-Signal-Modell, das in **Abbildung 24** gezeigt wird,



tragen alle 0.1ms Signale. Damit das SpaceClimber Gelenkmotormodell stabil läuft, wurden die verschiedenen Sampling-Raten für die Blöcke wie im Folgenden definiert:

- BLDC MotorBlock verwendet 0.1ms.
- Speed Controller verwendet 1ms.
- Position Controller und Lowpass Filter verwenden 10ms.

Dies sind die Systemparameter für das SpaceClimber-Gelenk:

- L (Induktivität des Läufers): $605 \cdot 10^{-6}$ H
- R (Widerstand der Ankerwicklung): 0.521Ω
- K_m (Motor-Drehmoment-Konstante): 0.075 N·m/A
- J (Trägheitsmoment): $4.9 \cdot 10^{-6}$ kg·m²
- b (Viskoser Reibungskoeffizient): $1 \cdot 10^{-3}$ N·m·s/rad
- K_b (EMK-Konstante): 0.075 V·s/rad

3.1.17 AP3350: Bodenmechanik

Die Simulation der Bodenmechanik für Laufroboter ist derzeit noch ein relativ offenes Feld. Im Bereich der radgetriebenen Systeme existieren viele Arbeiten, die sich mit der Bodenmechanik-Simulation beschäftigen. Diese Arbeiten gelten sowohl bei der Modellierung als auch für die Verifikation der entworfenen Modelle als richtungsweisend. Auf Seiten des DFKI's wurden zwei Ansätze zur Modellierung der Bodenmechanik näher betrachtet. Diese werden im weiteren kurz erläutert.

Ansatz 1: Fuß-Boden-Kontakt-Modell basierend auf der Bekker Theorie

Es wird ein Fuß-Boden-Kontakt-Modell definiert. Dieses Modell verwendet eine Punkt-Kontakt-Dynamik zwischen einem kugelförmigen Fuß und dem Boden. In diesem Modell werden die Normalspannung und die Schubspannung beim Vortrieb dynamisch definiert, ohne dabei durch Änderung des Polygon-Netzes die Bodenverformung umzusetzen. Wenn das Roboter-Bein mit seinem kugelförmigen Fuß den homogenen Boden mit einer bestimmten Kraft wie in Bild 27(a) berührt, werden nach der Zeit T1 am Boden eine Normalspannung und eine Schubspannung wie in Bild 27(b) erzeugt, und sie werden dynamisch entsprechend der Kontakt-Kraft und ihrer Richtung geändert.

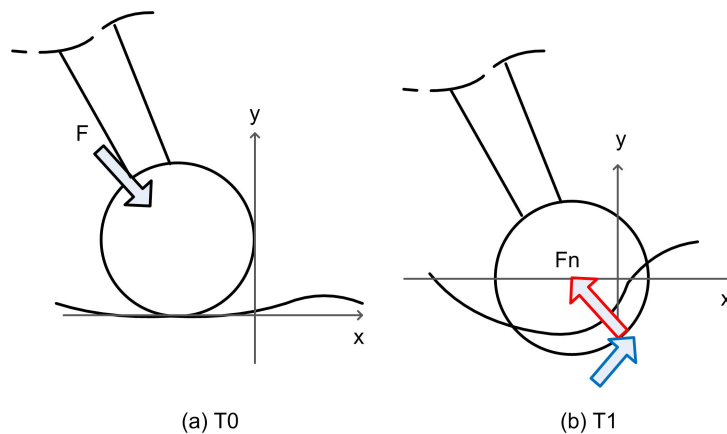


Abbildung 27: Normalspannung und Schubspannung beim Fuß-Boden-Kontakt

Die dynamische Normalspannung entlang der X-Achse und der Y-Achse kann durch Modifikation der penalty-basierten Methode durch die Boden-Deformations-Eigenschaften folgendermaßen modelliert werden:

$$F_{yN} = \begin{cases} k_y dy + p_y \dot{d}y & (dy > 0 \text{ and } \dot{d}y \geq 0) \\ 0 & (dy \leq 0 \text{ or } \dot{d}y < 0) \end{cases}$$

$$F_{xN} = \begin{cases} k_x dx + p_x \dot{d}x & (dy > 0) \\ 0 & (dy \leq 0) \end{cases}$$



Dabei ist F_{yN} die Normalspannung entlang der Y-Achse; F_{xN} ist die Normalspannung entlang der X-Achse; k_x und k_y sind die Härte des Bodens; p_x und p_y sind die Dämpfungsfaktoren des Bodens. Die Normalspannung kann verwendet werden, um die dynamische Schubspannung am Kontakt-Punkt zu berechnen. Das Bekker-Beweglichkeits-Modell beschreibt den Zusammenhang zwischen Normalspannung und Schubspannung am Boden durch folgende Erweiterung des Mohr-Coulomb-Kriteriums:

$$\tau_{max} = c + \sigma \tan \phi \quad (6)$$

Dabei ist τ_{max} die maximale Schubspannung des Bodens bei einer gegebenen Normalspannung σ , c ist die Kohäsionsspannung des Bodens und ϕ ist der Winkel der inneren Scherfestigkeit des Bodens. Wenn die Schubbelastung durch den Kontakt höher ist als die maximale Schubspannung, tritt ein Rutschen auf. Dieser Zusammenhang wurde von Janosi und Hananoto [26] entdeckt und eine generell anerkannte Näherung, um die Schubspannung τ zu bestimmen, ist:

$$\tau = \tau_{max} (1 - e^{-j/K}) \quad (7)$$

Dabei ist j die Scherverformung und K ist der "modulus" der Scherverformung. Gleichung 7 kann verwendet werden, um die Schubspannung des Fuß-Boden-Kontaktes zu berechnen und die horizontale und vertikale Schubspannung werden wie folgt beschrieben:

$$\tau_x = (c + F_{yN} \tan \phi) (1 - e^{-j_x/K}) \quad (8)$$

$$\tau_y = (c + F_{xN} \tan \phi) (1 - e^{-j_y/K}) \quad (9)$$

Daher können die Schubspannungs-Modelle (Eq 8 und Eq 9) des Fuß-Boden-Kontaktes in das Normalspannungs-Modell integriert werden, basierend auf dem penalty-basierten Modell-Ansatz. Der Vorteil dieses Kontakt-Modells ist, dass die Normalspannung und die Schubspannung durch den Fuß-Boden-Kontakt dynamisch definiert werden ohne komplexe Änderung des Polygon-Netzes zur Repräsentation der Boden-Verformung.

Eine detaillierte Beschreibung findet sich in der Publikation: Yong-Ho Yoo, Larbi Abdenbaoui, Michael Rohn, Malte Roemmermann, Frank Kirchner: Modelling and Simulation of Mobile Six-Legged Robot with Leg-Soil Contact In Proceedings of the European regional Conference of the International Society for Terrain-Vehicle Systems, (ISTVS-09), 05.10.-08.10.2009, Bremen.

In diesem AP wurde das Konzept weiter entwickelt. Das Ziel ist, das Punkt-Kontaktmodell, das auf der Bekker-Theorie basiert, auf ein 3D-Flächen-Kontaktmodell zu erweitern. Das energiebasierte Partikel-Kontaktmodell und seine Leistungskopplung werden als Möglichkeit vorgeschlagen, dieses Ziel zu erreichen.

Eine Boden-Oberfläche in Kontakt mit einem starren Fuß hat plastische Eigenschaften wie in Abb. 28. Betrachtet sei das Kontaktverhalten zwischen dem Boden und dem Fuß. Wenn der starre Fuß in den Boden eindringt, gibt der Boden eine Gegenkraft an den starren Fuß zurück und verformt sich dabei, wobei die durch den Kontakt veränderte Form des Bodens beibehalten werden muss, wenn der Fuß sich wieder aus dem Kontakt zurückzieht.

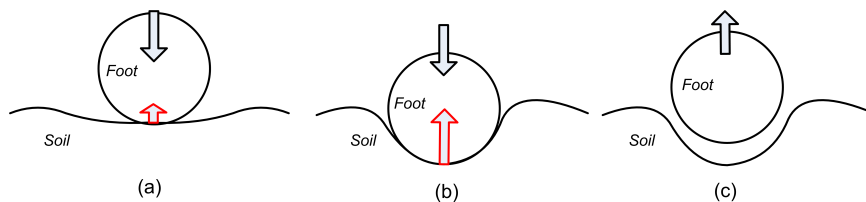


Abbildung 28: Das plastische Verhalten einer Boden-Oberfläche bei Kontakt

Ein plastisches Terramechanik-Partikel (PTP) wird vorgeschlagen um das dreidimensionale Verhalten bei einem Kontakt zwischen Roboter und Bodenoberfläche zu simulieren. Es wird in Abb. 29 gezeigt und die Eigenschaften des PTP-Modells sind die folgenden:

- Ein PTP ist ein physikalisches Modell das bei Kontakt ein dreidimensional plastisches Verhalten hat
- Eine Bodenoberfläche besteht aus einem oder mehreren PTPs
- Gegenkräfte, die von PTPs zurückgegeben werden, können miteinander kombiniert und so auf das Kontakt-Objekt aufgebracht werden
- PTPs sind voneinander unabhängig (d.h., die Gegenkraft eines PTPs kann nicht auf ein anderes PTP übertragen werden)
- Jedes PTP hat plastische x-y-z Parameter die dreidimensionale plastische Eigenschaften definieren können
- Wenn PTPs mit einem starren Körper in Kontakt stehen, geben PTPs eine Gegenkraft aus Bekker "pressure-sinkage"(Gl.10) an den Starrkörper zurück

$$p = \left(\frac{k_c}{b} + k_\varphi \right) z^n \quad (10)$$

Dabei ist p der Druck, k_c ist der Kohäsions-Modulus der Bodenverformung, k_φ ist der Reibungs-Modulus der Bodenverformung, n ist der bodenabhängige empirisch bestimmte Exponent und b ist der Kontakt.

- Wenn ein Kontakt zwischen einem Starrkörper und dem Boden besteht, geben PTPs eine Gegenkraft aus der Schubspannung (Gl. 11) an den Starrkörper zurück. Diese Kraft basiert auf der Gleichung von Janosi und Hanamoto:

$$\tau = (AC_0 + \sigma \cdot \tan\phi) \{1 - e^{(-j/K)}\} \quad (11)$$

Dabei ist j die Schubverformung, K ist der Modulus der Scherverformung, τ ist die Schubspannung des Bodens für eine gegebene Normalspannung σ , C_0 ist die Kohäsionsspannung des Bodens, A ist die Kontaktfläche, ϕ ist der Winkel der inneren Scherfestigkeit des Bodens.

- Wenn ein Kontakt zwischen einem Starrkörper und dem Boden besteht, üben PTPs eine Gegenkraft aus der Bulldozing-Kraft auf den Starrkörper aus: Eine horizontale Kraft (Abb. 30 (a)) und eine Ausgrabungskraft (Abb. 30(b)).

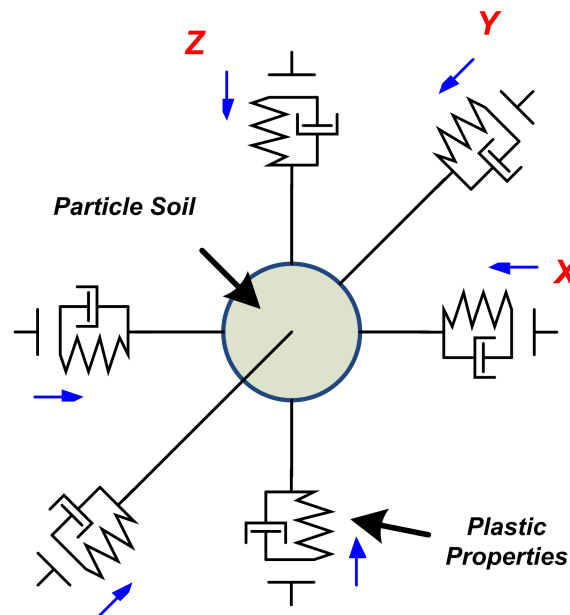


Abbildung 29: Das plastische Terramechanik-Partikel

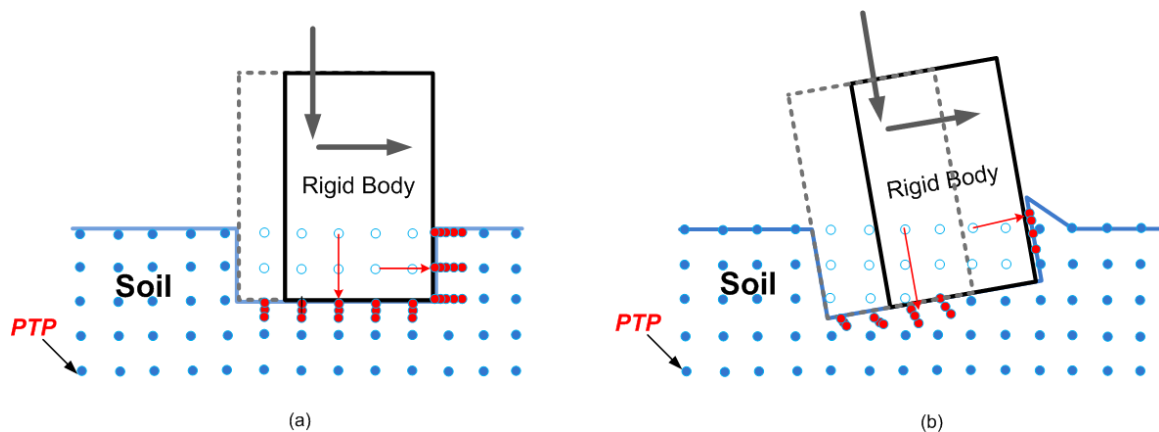


Abbildung 30: Bulldozing-Kraft mit den plastischen Terramechanik-Partikeln: (a) ist das horizontale Verhalten und (b) ist das Ausgrabungsverhalten

Ansatz 2: Bodenmechanik basierend auf Neuronalen Netzen

Grundlegend soll bei diesem Ansatz ein neuronales Netz trainiert werden, welches einen physikalischen Kontaktzustand (Eindringtiefe, relative Geschwindigkeit, relative Orientierung, etc.) auf Kontaktkräfte abbildet. Um das Prinzip zu untersuchen wurde zunächst ein vereinfachtes Szenario untersucht. Dabei werden zunächst die Beine (bzw. Füße) des Roboters als Zylinder betrachtet. Mindestens für Laufroboter mit ähnlichen Extremitäten wie sie der Scorpion hat (siehe Abbildung 3.1.17), kann diese Abstraktion gemacht werden. Im Rahmen einer ersten Untersuchung wurden

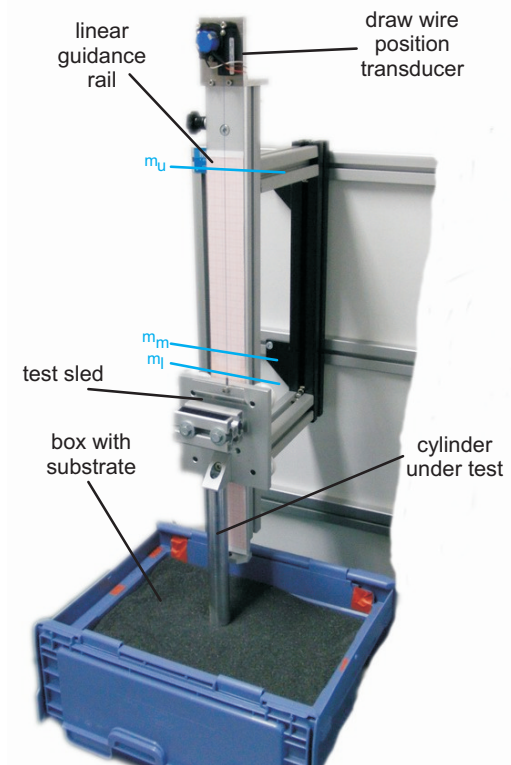
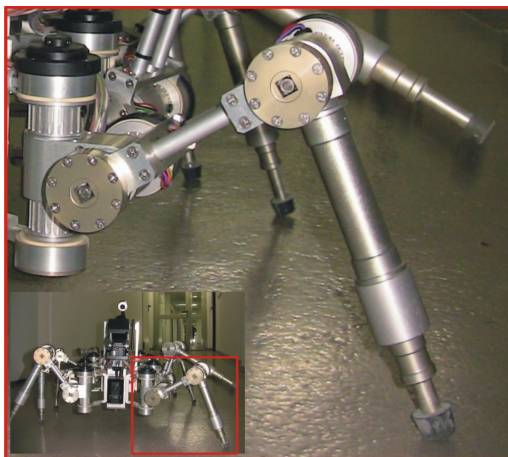


Abbildung 31: Links: Der Scorpion Roboter mit grafisch hervorgehobenem Bein. Rechts: Der Falltest-Aufbau.

zwei Experimente aufgebaut. Im ersten Experiment wurden verschiedene Zylinder aus unterschiedlichen Höhen in eine Box fallen gelassen. Die Box wurde in beiden Experimenten verwendet und war mit einem Regolith-Simulant gefüllt. Das zweite Experiment war ähnlich zu dem ersten, nur wurden hier die Zylinder (mit unterschiedlichen Eindringtiefen) seitlich durch die Box gezogen. Über entsprechende Encoder wurde die Position der Zylinder gemessen und abgespeichert. Die Experimente wurden mehrfach durchgeführt, um ein statistisches Verhalten der unterschiedlichen Zylinder und Experimentkonfigurationen zu ermitteln. In Abbildung 3.1.17 ist der Aufbau des ersten Experimentes zu sehen.

Abbildung 3.1.17 zeigt die Architektur des neuronalen Netzes, welches für beide Experimente verwendet wurde. Grundsätzlich sollte das Netz eine Abbildung der Eindring-

tiefe, des Zylinderdurchmessers und der Zylinder-Geschwindigkeit auf Kontaktparameter der Simulation lernen. Dieser Ansatz hat für das einfach gehaltene Szenario sehr gut funktioniert und kann dennoch für ein komplexeres Szenario deutlich verbessert werden. Zum einen kann die Netzarchitektur zusätzlich optimiert werden und zum anderen kann das Netz weitere physikalische Parameter als Eingabe erhalten und eine Abbildung direkt auf Kontaktkräfte darstellen. Eine detaillierte Beschreibung findet sich in der Publikation "Malte Roemmermann, Daniel Kuehn, Florian Cordes, Yong-Ho Yoo, Frank Kirchner: *Concept Evaluation of Modeling Terrain Mechanics by a Neural Network*, In 11th European Regional Conference of the International Society for Terrain-Vehicle Systems, (ISTVS-09), 05.10.-10.10.2009, Bremen".

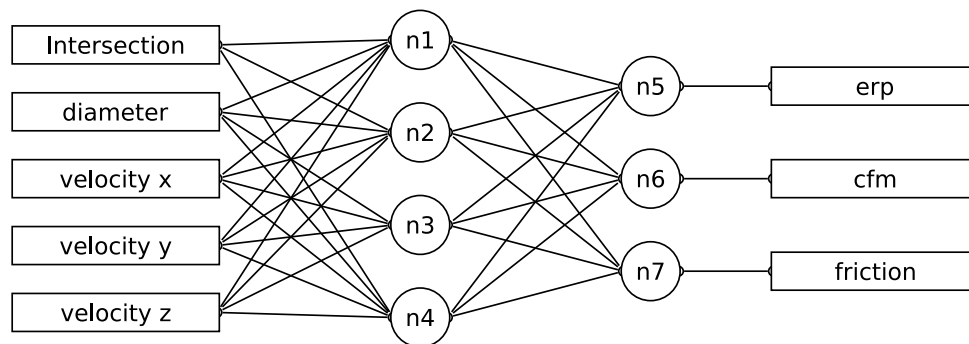


Abbildung 32: Architektur des verwendeten neuronalen Netzes zum Erlernen der Bodenmechanik für einen vereinfachten Anwendungsfall.

Die Ergebnisse haben erwiesen, dass der Ansatz grundsätzlich funktionieren kann. Für das komplexe Gesamtverhalten bei der Bodeninteraktion soll die bestehende Kontaktdynamik in VEROSIM[®] komplett durch ein neuronales Netz ersetzt werden. Die Ausgabe des Netzes ist die Kraft \vec{f}_b , die durch die Bodeninteraktion auf den Fuß des Roboters wirkt. Die Eingabeparameter des Netzes sind:

1. Die Eindringtiefe des Fußes in das Substrat d .
2. Das Quaternion welches die Rotation des Fußes beschreibt q
3. Der Bewegungsvektor des Fußes \vec{v} .
4. Der Winkelgeschwindigkeit des Fußes $\vec{\omega}$

Um ein optimales Ergebnis erzielen zu können, wird ein Optimierungsverfahren eingesetzt, welches sowohl die Architektur des neuronalen Netzes als auch die Parameter (Gewichte, Schwellwerte, etc.) optimiert. Der Algorithmus, der zu diesem Zweck eingesetzt wird, ist EANT (Evolutionary Artificial Neural Networks).

Im Gegensatz zu der analytischen Umsetzung der Bodendynamik werden für die Verwendung eines neuronalen Netzes bereits zur Entwicklung des Netzes Referenzexperimente benötigt. Dabei wird eine Teilmenge der Experimente verwendet, um das Bodendynamik-Netz zu entwickeln, während die restlichen Experimentdaten zum Verifizieren des Netzes benötigt werden.



3.1.18 AP3360: CAD-Import

In diesem Arbeitspaket wurde definiert welche Aspekte beim Import eines CAD-Modelles für die Simulation wichtig sind. In Zusammenarbeit mit dem Projektpartner wurde beschlossen, dass die Umsetzung dieser Definitionen durch ein Plugin direkt in der CAD-Anwendung am besten zu realisieren ist. Da in dem Projekt nicht für alle CAD-Anwendungen ein solches Plugin entwickelt werden kann, wurde eine CAD-Anwendung nämlich *Inventor* ausgewählt.

Punkte die von dem Import erfüllt werden müssen, sind folgende:

- Übername des visuellen Modelles
- Übername des vereinfachten Physikalischen Modelles (wenn vorhanden)
- Direktes Erzeugen eines VEROSIM[®] Modelles aus dem CAD-Programm heraus
- Übername von Gewichten

Des Weiteren sind ein paar Bedingungen an das CAD-Modell gebunden, damit der Import möglich wird. Diese sind:

- Erstellung der Modelle in Inventor
- Kennzeichnung der Modelle als visuelles und physikalisches Modell
- Angabe von Gewichten

3.1.19 AP3400: Definition Referenzexperimente

Innerhalb dieses Arbeitspaketes wurden die Referenzexperimente zu den Sensoren, den Aktuatoren, der Bodendynamik und dem Gesamtsystem definiert.

Referenzexperimente zu den Sensoren

In diesem Abschnitt werden die Experimente zu den aus AP2300 definierten Sensoren beschrieben. Anhand der Experimente wird das spezifische Sensorrauschen, der Einfluss der relevanten Umgebungsfaktoren und die Abbildungsfunktion, welche aus den Eingabeparametern des Sensors den Ausgabewert berechnet, ermittelt. Zu allen Experimenten muss die Temperatur zum Zeitpunkt der Durchführung notiert werden.

Inertialsensoren Die Inertialsensoren werden getestet, indem sie auf einem Brett befestigt werden, welches mit reflektiven Markern bestückt ist. Anhand der Marker kann durch den Einsatz des hauseigenen Tracking-Systems die Translation und Rotation



des Brettes aufgenommen und mit den Werten des Inertialsensors verglichen werden. Während der Experimente wird das Ruheverhalten und das Verhalten der Inertialsensoren während einer Bewegung/Drehung entlang aller sechs Freiheitsgrade ermittelt. Anhand dieses Versuchsaufbaus werden der 3-Achsen-Beschleunigungssensor LIS3LV02DL und die DFKI-IMU untersucht werden. Idealerweise muss versucht werden, das Koordinatensystem, welches anhand der Marker im Vicon-System erstellt wird, direkt innerhalb des Inertialsensors zu platzieren.

Winkelsensoren Die Referenzexperimente zum Spaceclimber-Gelenk werden mit dem vorhandenen Testprogramm durchgeführt, mit dem die Stromwerte, die Temperatur und Winkelwerte der Gelenke ausgelesen werden können. Es werden alle 12000 Positionen angesteuert und für jede Position 100 Sensorwerte abgefragt. Anhand der gemessenen Abweichungen wird die Gauß-Funktion erstellt, welche das Rauschen des Sensors beschreibt.

Zug-Druck-Kraftsensoren In den Experimenten muss das Rauschverhalten des Sensors auf den drei translatorischen (F_x, F_y, F_z) und den drei rotatorischen (T_x, T_y, T_z) Achsen bestimmt werden. Das Rauschen für F_z kann ermittelt werden, indem der Sensor mit 1 bis 29 kg beschwert wird. Das Rauschen für F_x und T_y kann gemessen werden, indem der Sensor mit der x-Achse nach oben an der Wand befestigt wird und mit bis zu 5 Nm belastet wird. Wird der Sensor mit der y-Achse nach oben befestigt, kann das Rauschen für F_y und T_x gemessen werden. Bei diesem Versuchsaufbau werden die Gewichte an einer Stange befestigt, die parallel zur z-Achse am Sensor befestigt wird. Der benötigte Drehmoment, um das Rauschen bezüglich T_z zu messen, kann erzeugt werden, indem die Stange entlang der y-Achse am Sensor befestigt wird und anschließend mit Gewichten beschwert wird. Hierbei zeigt die x-Achse nach oben. Das Rauschen aller Achsen sollte zusammengefasst werden, sofern sie sich nicht zu deutlich unterscheiden. Die angegebenen Kräfte beziehen sich auf die Sensor-Kalibrierung SI-145-5.

Drucksensoren Die Ausgabewerte des Drucksensors hängen von der Aufttrittskraft, dem Aufttrittswinkel und unter Umständen von dem Material des Untergrundes ab. Aus diesem Grund werden zuerst Experimente mit unterschiedlichen Untergründen (Sand, Kies und ein fester Boden) durchgeführt. Hierbei wird der Fuß mit Gewichten von 0,5 bis 10 kg (in Schritten von einem halben Kilogramm) beschwert und auf verschiedenen Untergründe gedrückt. Weichen die Messergebnisse deutlich voneinander ab, müssen die nachfolgenden Experimente für alle drei Untergrund-Arten durchgeführt werden. Bei diesen Experimenten wird der Fuß gekippt (in fünf Grad-Schritten von 5 bis 45°) und mit den gleichen Gewichten wie zuvor auf den Untergrund gedrückt. Bei jeder Messung werden 1000 Sensorwerte aufgenommen. Insgesamt müssen 200 (600 falls die Beschaffenheit der Untergründe berücksichtigt werden muss) verschiedene Messungen vorgenommen werden. Anhand der aufgenommenen Daten wird die Abbildungsfunktion erstellt, welche aus den Eingabeparametern (Kraft und Aufttritts-



winkel) den Ausgabewert des Sensors generiert. Zusätzlich wird aus allen Messungen das Rauschverhalten des Sensors abgeleitet. Die genaue Zusammensetzung des Materials, in dem der Drucksensor eingegossen wurde, und dessen Dichte muss notiert werden.

GPS Hier wurde geplant, die Referenzexperimente zum Magellan MB500 außerhalb des DFKIs durchzuführen. Hierzu sollte das mobile Qualisys-Tracking-System eingesetzt und der GPS-Sensor mit reflektiven Markern bestückt werden. Anschließend sollte das Ruheverhalten und das Verhalten des Sensors bei einer Positionsveränderung entlang aller drei Freiheitsgrade aufgenommen und mit den Referenzwerten des Tracking-Systems verglichen werden. Anhand der Daten könnte das Rauschverhalten des Sensors bei der Positionsbestimmung ermittelt werden. Da der SpaceClimber Roboter jedoch keinen GPS-Empfänger besitzt und aufgrund seiner Ausrichtung auf den Space-Bereich voraussichtlich auch nicht mit einem solchen ausgerüstet werden wird, wurde von der tatsächlichen Durchführung der zugehörigen Referenzexperimente abgesehen.

Kameras Wie in Abschnitt 3.1.15 beschrieben, können simulierte und reale Kamerabilder nur schwer miteinander verglichen werden, weswegen an dieser Stelle keine Referenzexperimente definiert werden können.

Laserscanner Bei diesen Experimenten muss sowohl das Rauschverhalten des Laserscanners als auch der Einfluss unterschiedlich farbiger Untergründe auf die Messergebnisse bestimmt werden. Zur Bestimmung des Rauschverhaltens werden von 80 bis 4000 mm (in 80 mm Schritten) je 100 Entfernungsmessungen durchgeführt. Bei diesen Messungen wird ein neutral-weißer Untergrund benutzt. Laut Datenblatt kommt es unterhalb einem Meter zu einer Abweichung von ± 10 mm und oberhalb einem Meter zu einer Abweichung, die 1% der gemessenen Entfernung entspricht. Um den Einfluss der Untergrundfarben zu überprüfen, wird in einer Entfernung von einem halben Meter ein ausgedruckter Farbverlauf (Farbe und Graustufen) angebracht. Anschließend werden für jeden Scanner-Schritt 100 Entfernungsmessungen durchgeführt. Anhand des bekannten Messwinkels kann die zu erwartende Entfernung und die Abweichung berechnet werden. Sollten die unterschiedlichen Auftrittswinkel die Messergebnisse verfälschen, muss der Farbverlauf auf einem Halbkreis aufgebracht werden, in dessen Zentrum der Scanner steht. In diesen Experimenten muss außerdem geklärt werden, ob aus unterschiedlichen Untergrundfarben ein unterschiedliches Rauschverhalten resultiert.

PMD-Kameras Das Rauschen der PMD-Kamera wird ermittelt, indem die Kamera in einer aufrechten Position vor einer weißen Wand aufgestellt wird. Anschließend wird die Kamera schrittweise zurück bewegt und für jede Entfernung werden 100 Bilder aufgenommen. Aus diesen Messungen kann für jeden Pixel das Rauschen in Abhängigkeit von der Entfernung ermittelt werden, wodurch wiederum das durchschnitt-



liche Rauschen der Kamera berechnet werden kann. Anschließend werden Referenzmessungen vor einer farbigen Wand durchgeführt. Sollten sich die vor der weißen und farbigen Wand aufgenommenen Entfernungen stark unterscheiden, müssen wie beim Laserscanner Farbverläufe aufgenommen werden. Die zu erwartenden Entfernungen für jeden Pixel können auch hier anhand des vertikalen und horizontalen Öffnungswinkels der Kamera berechnet werden.

Referenzexperimente zu den Aktuatoren

Um ein möglichst genaues Echtzeit fähiges Modell der Aktuatoren bilden zu können, soll zunächst ein detailliertes Aktuator-Modell des SpaceClimber Roboters an die realen Roboter-Gelenke angeglichen werden. Dabei dient das PWM Signale als Eingabeparameter und die zu vermessenden Werte sind die Stromaufnahme des Gelenks und die rotatorische Geschwindigkeit hinter dem Getriebe (bzw die Gelenkposition). Dieses Experiment soll mit unterschiedlichen Lasten an dem Gelenk durchgeführt werden. Abgeleitet aus dem detailliertem Gelenkmodell soll ein Echtzeit fähiges abstraktes Modell erstellt werden. Dieses Modell wird dann die Gelenkansteuerung beinhalten und somit die Zielposition (eventuell auch die Sollgeschwindigkeit) als Eingabeparameter verarbeiten. Innerhalb eines Vergleichsexperimentes sollen mit verschiedenen Lasten eine Folge von Zielpositionen angefahren werden. Gemessen und verglichen wird dann die Gelenkposition, -geschwindigkeit und das durch das Gelenk aufgebrauchte Drehmoment.

Gelenk-Referenzexperiment

Das Referenzexperiment für das Gelenk kann auf dem vorhandenen Motor-Prüfstand mit dem in Abb. 33 gezeigten Aufbau durchgeführt werden, um die hergeleiteten Modelle zu überprüfen.

Um das Modell zu verifizieren, wird die Reaktion des simulierten Modells mit den gemessenen Ausgangsgrößen des echten Gelenks unter den gleichen Eingangsgrößen verglichen. Die Sum of Square Of The Errors (SSE) Regel wird verwendet, um zu berechnen, wie exakt sich das abstrakte Modell im Vergleich zum echten Gelenk verhält.

Die SSE ist definiert als:

$$SSE = \frac{\sum_{k=1}^N [y(k) - \hat{y}(k)]^2}{\sum_{k=1}^N y^2(k)} \times 100 \quad (12)$$

Dieses Kapitel erläutert die Definition realer und virtueller Referenzexperimente, mit denen die Genauigkeit des abstrakten Gelenk-Motor-Modells überprüft werden soll.

Abb. 34(a) zeigt den realen Experiment-Aufbau, bei dem eine Umlenkrolle an dem Gelenk angebracht wurde, so dass diese einen angebrachten Draht aufwickeln und damit

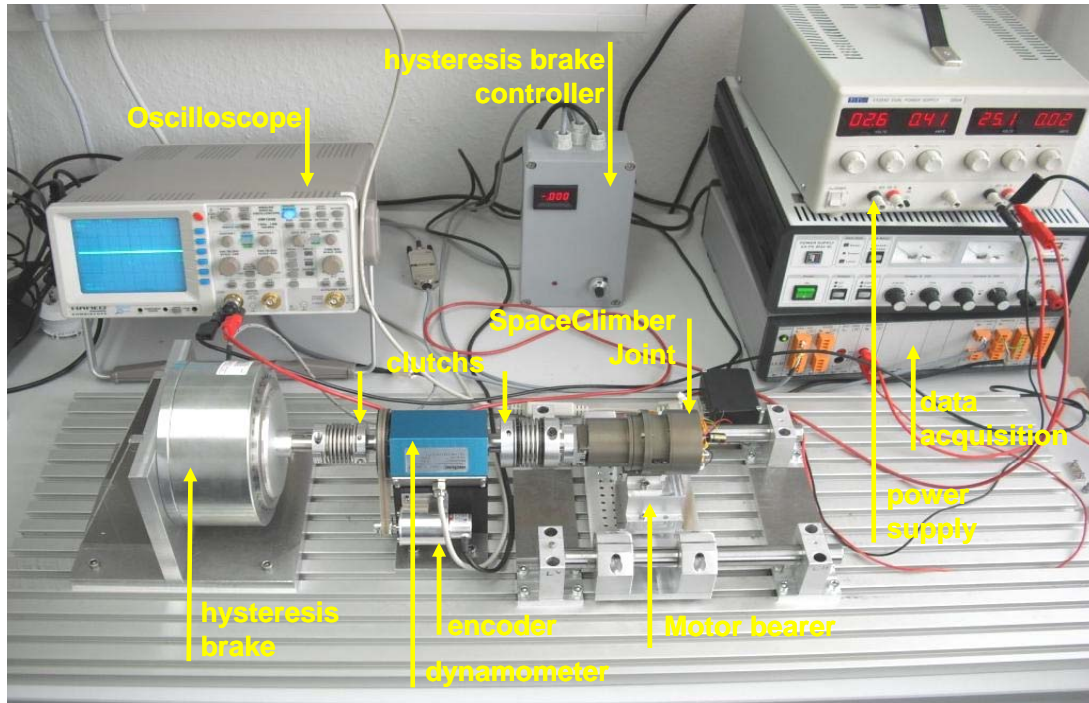


Abbildung 33: Aufbau des Gelenk-Referenzexperimentes

einen Rahmen anheben kann, wenn sich das Gelenk dreht. An diesem Rahmen können verschiedene Gewichte angebracht werden. Mit einem spezifischen PWM-Signal wird der Motor so angesteuert, dass er das angebrachte Gewicht anhebt. Dann werden die Ausgabewerte (Geschwindigkeit und Stromstärke) gemessen und aufgezeichnet. Verschiedene Gewichte sollten verwendet werden und jedes Experiment sollte für jedes spezifische PWM-Signal mehrfach wiederholt werden

Wenn wir einen Gelenk-Motor und seinen Controller in VEROSIM[®] simulieren, sollten wir das Problem der Verbindung zwischen den Signalkomponenten wie dem Motor-Controller und kinematischen Modellen wie der Masse des Gelenks und den Beinverbindungen in VEROSIM[®] berücksichtigen. Sei beispielsweise ein Gelenk-Motor über eine Verbindung mit einem Bein verbunden und nur das Geschwindigkeits-Signal werde vom Motor auf das Bein übertragen, dann fehlt hier im Gegensatz zur realen Verbindung die Rückkopplung vom Bein zurück zum Motor. Um solche Probleme zu lösen, können wir eine bidirektionale *Leistungs(Geschwindigkeit \times Kraft)-Kopplung* zwischen den Signalmodellen und den kinematischen Modellen verwenden. Abbildungen 35(a) und (b) zeigen zwei Szenarien für Leistungs-Kopplung, bei denen die Kopplung zwischen dem elektrischen und dem mechanischen Teil des Motors liegt. Dabei werden die elektrischen Komponenten (z.B. Motor-Widerstand und Induktivität) als Signalmodelle realisiert und die mechanischen Komponenten (z.B. interne Motorträgheit, Reibung und Getriebe) als kinematische Modelle realisiert. Bei den Kopplungs-Szenarios, die in Abb. 35(c) und (d) gezeigt werden, liegt die Kopplung am Rande der mechanischen Teile des Gelenks. Daher wird der gesamte Gelenk-Motor als Signalmodell umgesetzt (Transfer-Funktion) und ein Duplikat des mechanischen Teils des Motors

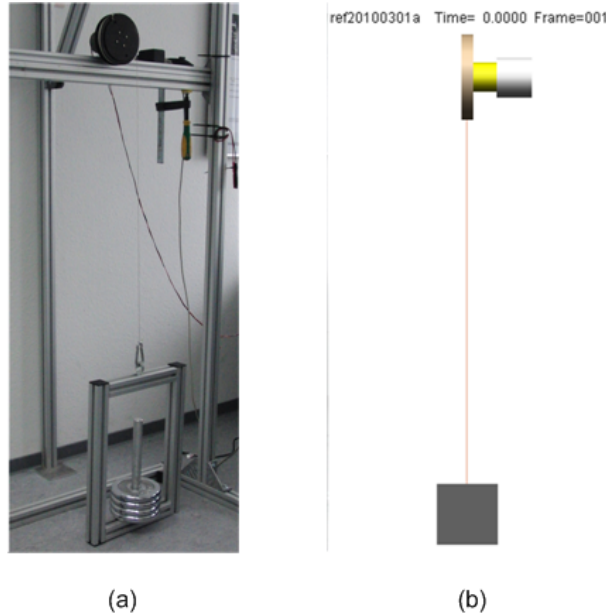


Abbildung 34: (a) realer Experiment-Aufbau, (b) virtuelles Referenzexperiment für das SpaceClimber-Gelenk

wird als kinematisches Modell mit der übrigen Kinematik des Roboters verbunden.

Im virtuellen Referenzexperiment wird das gleiche Experiment definiert wie für den realen Gelenk-Motor. Um das Gelenk-Motor-Modell bidirektional mit der Umlenkrolle zu verbinden, die das Gewicht hebt, sollte die oben erläuterte *Leistungs(Geschwindigkeit \times Kraft)-Kopplung* zwischen den Modellen verwendet werden. Abb. 34(b) zeigt das virtuelle Experiment für das Gelenk. Das Gelenkmodell in VEROSIM[®] wird entweder die Eingabewerte für die Geschwindigkeit und die Ausgabewerte für das Drehmoment messen oder umgekehrt. Masse und Volumen des Gelenks und der Umlenkrolle sollten aus den realen Modellen abgeleitet werden, aber die physikalischen Effekte des Drahtes würden ignoriert werden.

Referenzexperimente zu der Bodendynamik

Die Bein-Bodeninteraktion stellt eine der Hauptanforderungen an das Simulationssystem dar. Um diese in der Realität und der Simulation miteinander vergleichen zu können soll ein spezielles Testbed aufgebaut werden. Dieses ähnelt den Testumgebungen die vielerorts zum vermessen der Rad-Bodeninteraktion im Einsatz sind. Dabei wird eine Kiste (ca. 1,5m x 1,5m x 0.5m) aufgebaut die mit verschiedenen Substraten gefüllt werden kann. Im Vordergrund für das Projekt stehen Substrate die als Simulatoren von Mond-Regolith genutzt werden. Seitlich an der Kiste soll ein Industrieroboter fixiert werden (zur Zeit ist der Mitsubishi PA10 vorgesehen). Dieser Roboter muss über einen 6DOF Kraftsensor am Endeffektor verfügen. Für die Experimente sollen verschiedene Beine bzw. Füße an dem Endeffektor befestigt werden und in dem Substrat vermessen werden. Das Vermessen reicht von einfachen linearen Trajektorien mit konstanter Eindringtiefe bis hin zu nachgebildeten Stemmphasen wie sie bei einem Laufverhalten

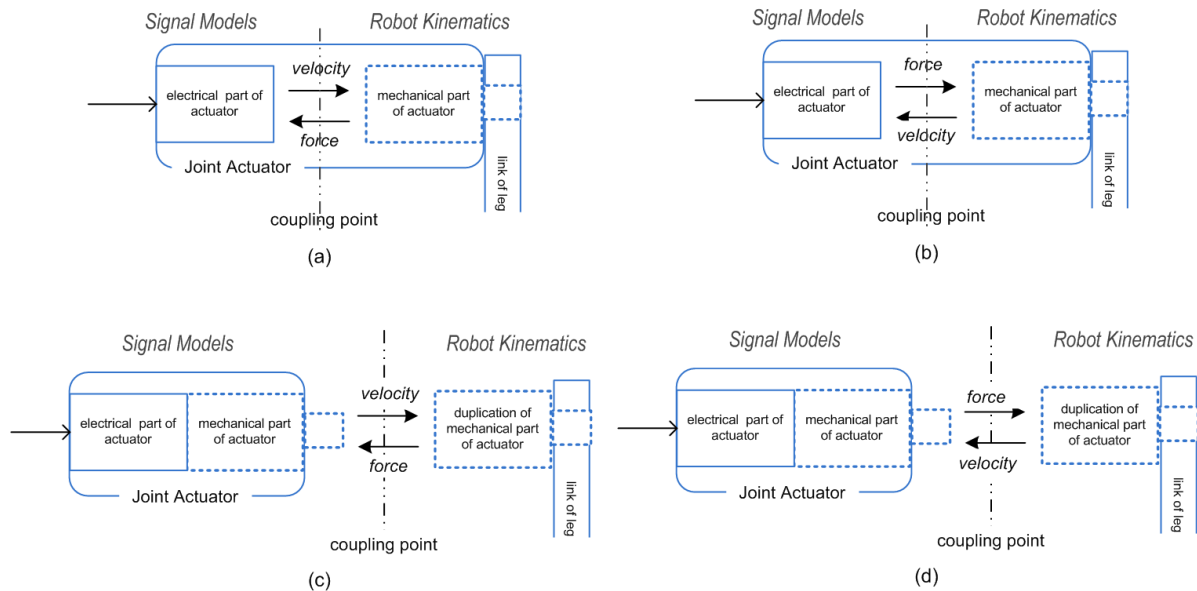


Abbildung 35: Szenarios für bidirektionale Leistungs-Kopplung zwischen Signalmodell und kinematischem Modell

auftreten. Um die Bein-Bodeninteraktion zu messen werden die Daten des Kraftsensors aufgenommen und archiviert. Diese Experimente werden analog in der Simulation durchgeführt, so dass die gemessenen Kraftwerte zum Bewerten der Simulationsgenauigkeit verwendet werden können.

Für die Referenzexperimente soll ein Unterschenkel des SpaceClimber Roboters verwendet werden. Der SpaceClimber ist der aktuellste Lauf- und Kletterroboter im DFKI und ist durch seine Eigenschaften (Kinematik, Kraft, Präzision, etc) am interessantesten für Mondmissionen. In Abbildung 3.1.19 ist der Unterschenkel in zwei verschiedenen Lagen zu dem Substrat abgebildet. Die Lagen unterscheiden sich durch den unterschiedlichen Winkel α des Unterschenkels. Die Pfeile in der Abbildung verdeutlichen die einzelnen Aktionen, die im folgenden aufgelistet sind:

- A_1 : Eintauchen in das Substrat.
- A_2 : Bewegen durch das Substrat.
- A_3 : Verdrehen in dem Substrat.

Wobei β eine Rotation der Aktionsrichtung beschreibt, die in der Regel zwischen 0° und α liegt. Diese Aktionen sollen einzeln und in Kombination durchgeführt werden. Eine Auflistung der jeweiligen Experimentkonfigurationen kann der Tabelle 9 entnommen werden. Die Winkel α und β sollen mit Schrittweiten von 10° variiert werden. Damit ergibt sich eine Gesamtzahl von 188 Experimenten aus Tabelle 9. Jedes einzelne Experiment soll 20 mal durchgeführt werden um Mittelwerte errechnen zu können.

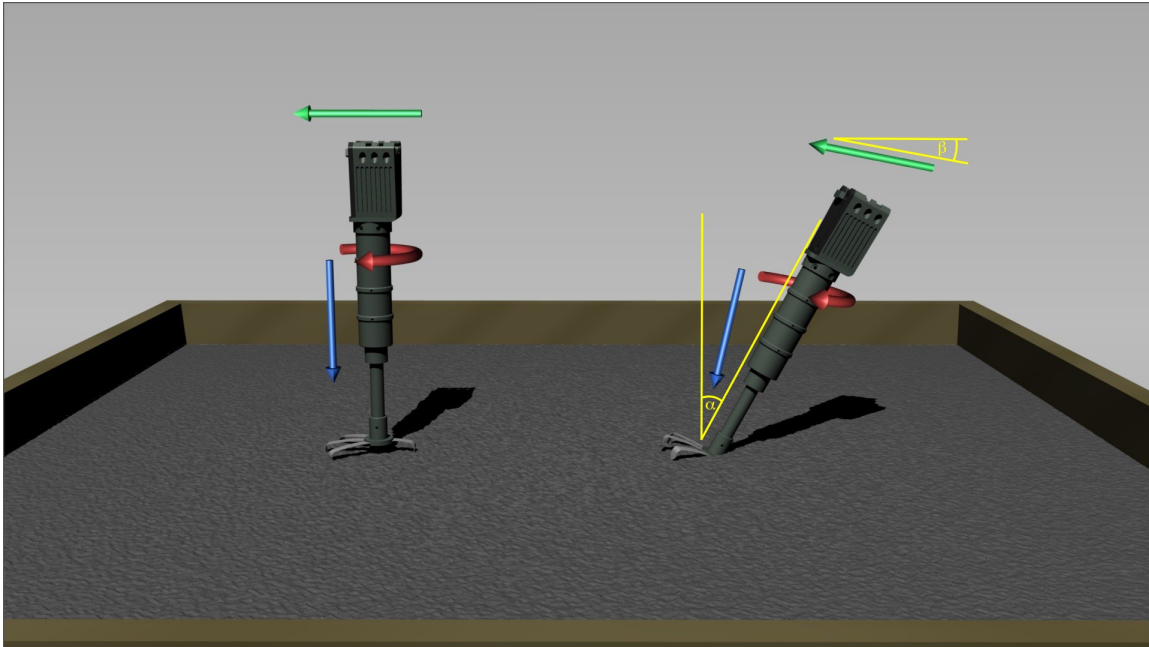


Abbildung 36: Experimentverdeutlichung der Bein-Bodeninteraktion.

Referenzexperimente zum Gesamtsystem

In diesem Abschnitt werden die Referenzexperimente definiert, bei denen der gesamte Roboter getestet wird. Bei diesen Experimenten wird ein Tracking-System verwendet, um die Positionen und Bewegungen der Füße, Beine und des Körpers des Roboters aufzunehmen. Zusätzlich werden die Stromwerte, die Potentiometerwerte der Gelenke, der Neigungswert des Roboters, der Gyrowert und die Kompression der Linearfedern beim Scorpion bzw. die Werte aus den Kraftmessdosen an den Beinaufhängungen beim Spaceclimber aufgenommen. Der Schwerpunkt bei diesen Experimenten liegt auf das Angleichen der Körperposition, -lage und -orientierung.

Laufen Bei diesen Experimenten wird der Roboter über unterschiedliche Untergründe bewegt. Dazu gehören sowohl harte Untergründe als auch Kies und Sand. Es ist außerdem möglich, die Steigung der Teststrecken bis zu einem Winkel von 45 Grad zu variieren. Die Ergebnisse dieser Experimente werden außerdem dazu benutzt, die Simulation von Untergründen in VEROSIM[®] anzupassen.



Experiment	A_1	A_2	A_3	α	β
Das Eindringen (A_1) stellt den Übergang von der Schwung- in die Stemmphase eines Beines dar und geschieht abhängig vom Laufmuster mit verschiedenen Winkeln des Beines und der Fußtrajektorie.	x x x			$0^\circ - 60^\circ$ 0° $0^\circ - 60^\circ$	0° $0^\circ - 40^\circ$ $0^\circ - 40^\circ$
Innerhalb der Stemmphase wird eine Fortbewegung des Roboters aus dem Widerstand des Substrates erzeugt. Dieser Widerstand kann vermessen werden indem das Bein durch das Substrat bewegt wird. Auch hier treten, abhängig vom Laufmuster, verschieden Winkel für α und β auf.		x x x		$0^\circ - 60^\circ$ 0° $0^\circ - 60^\circ$	0° $0^\circ - 40^\circ$ $0^\circ - 40^\circ$
Je nach Beinkinematik und Laufmuster findet eine Verdrehung des Fußes innerhalb der Stemmphase statt. Die dabei auftretenden Kräfte sollen über Aktion A_3 vermessen werden.			x x x	$0^\circ - 60^\circ$ 0° $0^\circ - 60^\circ$	0° $0^\circ - 40^\circ$ $0^\circ - 40^\circ$
Zuletzt sollen die Aktionen A_2 und A_3 kombiniert untersucht werden da beide gleichzeitig in der Stemmphase eines Beines auftreten.		x x x	x x x	$0^\circ - 60^\circ$ 0° $0^\circ - 60^\circ$	0° $0^\circ - 40^\circ$ $0^\circ - 40^\circ$

Tabelle 9: Tabellarische Übersicht der geplanten Bein-Boden Experimente.

3.1.20 AP3500: Entwurf von Verfahren zur Auswertung von Simulationsergebnissen hinsichtlich der Optimierung des Designs

Um eine gute Vorstellung davon zu erhalten, welche Informationen aus den Referenzexperimenten benötigt werden, wurden diese zusammen mit dem grundsätzlichen Optimierungsablauf formalisiert. Diese Abstraktion erlaubt zudem eine systematische Herangehensweise zur Definition der einzelnen Experimente. Abbildung 37 stellt den generellen Optimierungsablauf grafisch dar, wobei die Sensordaten aus den realen Experimenten nach der Verarbeitung in eine Datenbank geschrieben werden, während die Daten aus der Simulation direkt für das Optimierungsverfahren verwendet werden können.

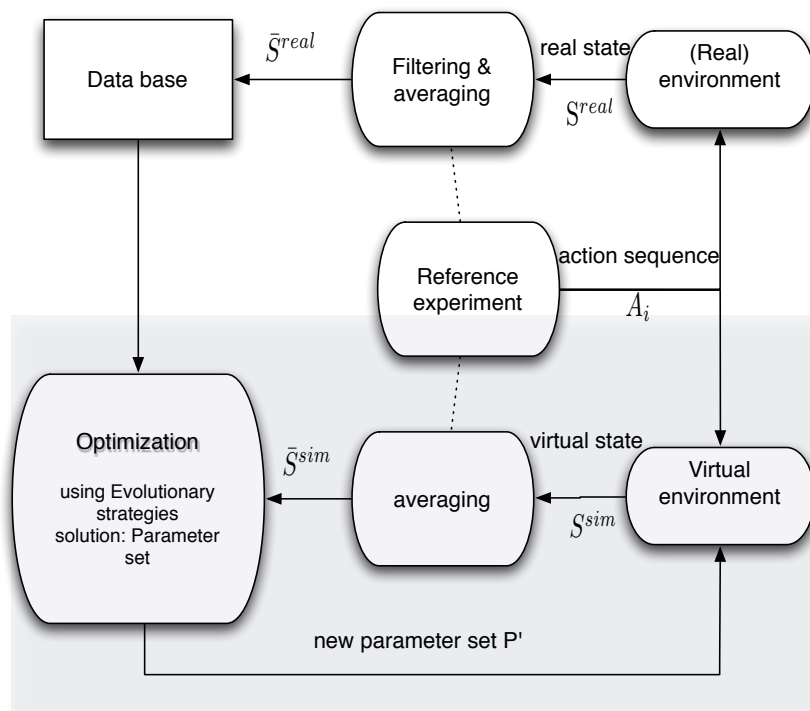


Abbildung 37: Illustration of the optimization approach

Der gesamte Optimierungsprozess besteht aus folgenden Schritten:

Definition Zu einem gegebenem Problem wird ein Referenzexperiment definiert. Dieses Experiment wird real und in der Simulation aufgebaut. Ein Referenzexperiment wird durch folgende Elemente definiert:

- Eine Menge von zu optimierenden Parametern $P = \{p_o, \dots, p_n\}$. Dies ist ein wichtiger Schritt und ergibt sich durch eine Analyse des Modells und die Identifizierung der relevanten Parameter. Zu diesem Schritt gehört auch das Definieren eines Wertebereiches für jeden Parameter p_i mit $r(p_i) = [min_{p_i}, max_{p_i}]$.



- Eine Menge von Aktionssequenzen: $A = \{A_1, \dots, A_m\}$. Wahlweise können die Aktionen explizit definiert werden. In diesem Fall ist jede Aktionssequenz eine Reihe von Tupeln, die eine Teilaktion und ihre Ausführungszeit beinhalten. Die Ausführungszeiten sind relativ zu der Startzeit der Aktionssequenz.

$$A_i = [(a_{i_0}, t_{i_0}), \dots, (a_{i_{w_i}}, t_{i_{w_i}})]$$

- Ein Vektor von Sensorwerten: $S = \{s_0, \dots, s_l\}$. Diese Werte müssen in dem realen und dem simulierten Experiment gleichermaßen gemessen werden. Um die Daten aus beiden Experimenten miteinander vergleichen zu können, sollte eine feste Messfrequenz pro Sensorwert definiert werden.
- Zuletzt muss eine Bewertungsfunktion $F(p)$ festgelegt werden. Das Ziel ist dann, einen Parametersatz P^* zu finden, der die Funktion $F(p)$ optimiert (minimiert oder maximiert).

Erstellen der experimentellen Daten Für ein auf der vorherigen Formalisierung basierendes Referenzexperiment können die Experimentdaten durch das Ausführen der Aktionen $A_i \in A$ erhoben werden. Das heißt, während der Ausführung der Aktionen werden die zugehörigen Sensorwerte $S^{real,i}$, mit

$$S^{real,i} = [(s_{00}^{real,i}, \dots, s_{l_0}^{real,i}), \dots, (s_{0t_i}^{real,i}, \dots, s_{lt_i}^{real,i})].$$

gemessen und gespeichert. Dies muss mehrmals durchgeführt werden, um Durchschnittswerte $\overline{s_{xy}^{real,i}}$ für ein Experiment zu erhalten. Die Ergebnisse sollen dann in eine Datenbank gespeichert werden und können somit für die jetzige Optimierung, aber auch für zukünftige Analysen verwendet werden.

Optimierungsschleife. Nach der Generierung der Experimentdaten aus den real durchgeführten Experimenten kann auf Seiten der Simulation die Optimierungsschleife ausgeführt werden. Diese beinhaltet folgende drei Schritte:

1. Zunächst wird in der Simulation die gleiche Aktionssequenz für einen gegebenen Parametervektor P ausgeführt und die Sequenz von simulierten Sensorwerten wird ermittelt:

$$S^{sim,i} = [(s_{00}^{sim,i}, \dots, s_{l_0}^{sim,i}), \dots, (s_{0t_i}^{sim,i}, \dots, s_{lt_i}^{sim,i})]$$

2. Danach wird eine Bewertung des Parametervektors berechnet, die als Feedback für die Optimierung verwendet wird. Die Bewertung basiert auf einer Fitnessfunktion, die beispielsweise über die Standardabweichung der realen und simulierten Sensorwerte definiert werden kann:

$$F(P) = \frac{1}{m} \sum_{i=1}^m \frac{1}{t_i} \sum_{j=1}^{t_i} \frac{1}{l} \sum_{k=1}^l |s_{kj}^{sim,i} - \overline{s_{kj}^{real,i}}|$$



3. Ein Optimierungsverfahren wird eingesetzt, welches versucht, einen Parametersatz P' zu finden, der die Fitnessfunktion $F(P)$ minimiert oder maximiert. Solange kein Stoppkriterium erfüllt ist, wählt das Optimierungsverfahren einen neuen Parametersatz und führt Schritt 1 und 2 aus.



3.1.21 AP4000: Implementierungsphase

3.1.22 AP4100: Benutzerschnittstelle Simulationssystem

3.1.23 AP4110: Visualisierung / Bedienung / Interaktion

Am 27.04.2010 fand ein Treffen mit dem Projektpartner RIF/MMI statt, bei dem anhand der Inhalte von AP3110 erörtert wurde, welche Eigenschaften für dieses Arbeitspaket in VEROSIM[®] durch RIF/MMI zu implementieren sind. Grundsätzlich soll es möglich werden, die Simulation des SpaceClimber-Roboters interaktiv zu beeinflussen, indem beispielsweise über einen Datenhandschuh während der Simulation Steine gegriffen und an eine andere Stelle gelegt werden oder dem SpaceClimber so ein neuer Zielpunkt vorgegeben wird.

Dieses Arbeitspaket wird vom RIF koordiniert und auch die durchzuführenden Arbeiten liegen zum größten Teil beim Projektpartner.

Im Bereich der Visualisierung wurden folgende Funktionen implementiert:

Kraftvektoren Es wurde die Möglichkeit implementiert in die 3D-Szene auftretende Kräfte als farbige unterschiedlich große Vektoren direkt an den Gelenken einzuzeichnen. So ist es möglich die aktuell auf die Gelenke wirkenden Kräfte zu sehen.

Heightmap Um Landschaften wie zum Beispiel den DFKI-Krater in die Simulation zu übernehmen wurde ein Import für eine gerasterte Höhenkarte mit der zugehörigen Visualisierung implementiert.

Ansicht physikalisches Modell Um eine schnelle Berechnung der Physik eines Modells zu erreichen, wird das physikalische Modell zum Beispiel eines Roboters aus vereinfachten Geometrien aufgebaut. Mit diesen einfachen Geometrien wie Boxen, Zylinder usw. wird der reale Roboter möglichst nahe nachgebildet. Der Vorteil zu einem detaillierten Modell zum Beispiel direkt aus den CAD-Daten hat den Vorteil, dass die Kollisionen und damit Kräfte auf den Primitiven sehr viel schneller zu berechnen ist. In der Regel kostet ein detailliertes Modell mehr Rechenzeit als es in der Genauigkeit bringt. Ein Problem hierbei ist jedoch, dass wenn sich visuelle und physikalische Modell (durch einen Fehler beim Modellieren) unterscheidet dies nur schlecht festgestellt werden kann. Hierfür würde das Umschalten zwischen dem visuellen und physikalischen Modell implementiert.

Im Bereich der Interaktion wurde vom RIF der im DFKI vorhandene Datenhandschuh CyberTouch an die Simulation angebunden. Es ist nun möglich eine virtuelle Hand in Kombination mit einem InterSense-Trackingsystem in der virtuellen Welt zu bewegen. Diese Hand kann zum Interagieren mit der virtuellen Welt genutzt werden. So ist es möglich die Hand in die Nähe eines Steines oder anderen Objektes zu bewegen. Sofern die Hand geöffnet ist, können Objekte angestoßen werden. Sobald die Hand geschlossen (Faust) wird, wird die Hand mit dem nächsten Objekt welches berührt wird verbunden. So kann ein Stein in den Weg des Roboters gelegt werden, oder aber



der Roboter an eine andere Stelle platziert werden. Sobald die Hand wieder geöffnet wird, wird die Verbindung wieder gelöst.

Des Weiteren wurde zur Steuerung der Kamera eine so genannte 3D-Maus angebunden. Mit dieser 3D-Maus ist es komfortabel möglich alle 6 Achsen der Kamera simultan zu steuern.

Für Joysticks und Gamepads können Objekte im Simulationssystem erzeugt werden und deren Ausgänge über das IO-Framework beliebig mit anderen Komponenten verbunden werden.

3.1.24 AP4120: Entwicklungsumgebung für Erweiterungen

Bei dem Treffen mit dem Projektpartner RIF/MMI am 27.04.2010 wurde festgelegt, welche Systeme zur Erweiterung des Simulationssystems dem DFKI in welcher Form zur Verfügung gestellt werden. Dies sind einmal das Skriptsystem SOML++ sowie das Plugin-System, das hauptsächlich für Erweiterungen an VEROSIM[®] durch das DFKI genutzt werden wird. Durch Weiterentwicklungen, Dokumentation und direkte Unterstützung wird das DFKI durch den Partner RIF/MMI bei der Nutzung dieser Erweiterungssysteme unterstützt werden.

Die Entwicklungsumgebung für Erweiterungen wurde vom RIF dem DFKI zur Verfügung gestellt. Mit Hilfe des SSolutionExplorers für Entwicklung von VEROSIM[®] Plugins und der zugehörigen Dokumentation konnten am DFKI erforderliche Plugins für VEROSIM[®] entwickelt werden.

3.1.25 AP4200: Implementierung der Schnittstellen zwischen realer Steuerung und Simulationssystem

Die Anbindung der Roboter-Steuerprogramme HOBBIT und MONSTER an VEROSIM[®] wurde entwickelt.

Der Kommunikations-Rückweg von VEROSIM[®] zu MONSTER wurde implementiert und konnte validiert werden. Sensorwerte aus VEROSIM[®] werden seither im GUI des HOBBIT Controllers angezeigt.

3.1.26 AP4300: Projektspezifische Erweiterungen am Simulationssystem

3.1.27 AP4310: Darstellung

Bei dem Treffen mit dem Projektpartner RIF/MMI am 27.04.2010 wurde festgelegt, welche Darstellungsfunktionen für das Projekt zusätzlich in VEROSIM[®] implementiert werden sollen. Dazu gehören beispielsweise ein Ghost-Modus, Umschalten zwischen physischem Modell und grafischem Modell, Transparenz, Kartendarstellung, Anzeige von Kraftvektoren, Darstellung mechanischer Belastung über entsprechende Farben,



Darstellung verschiedener Werte als Zahlen und Diagramme, Auf Height-Field basierende Darstellung eines Mondkraters und Schattendarstellung.

Der graphische IO-Editor von VEROSIM® wurde verbessert. Damit lassen sich leichter Verbindungen zwischen Untermodellen herstellen und verwalten.

Es wurden Möglichkeiten geschaffen, um Sensorwerte direkt in der 3D-Visualisierung darzustellen. Zum einen können in beliebige Qt-Widgets – und damit auch das VEROSIM® Oszilloskop - in Form einer Textur in die 3D-Visualisierung eingebracht werden. Zum anderen bietet die Klasse VSPluginSimpleVisualizationMetaphors::IO Billboard die besonders einfache Möglichkeit, alle Ausgänge mit Name und Wert eines Sensors in textueller Form auf einem Billboard darstellen zu lassen.

3.1.28 AP4320: Dynamik

Der Partner RIF/MMI entwickelte die generellen Dynamik-Simulationskomponente von VEROSIM® weiter, insbesondere unter Berücksichtigung der im Rahmen von AP4350 einzubindenden bodenmechanischen Effekte. Entsprechende Anforderungen wurden auch bei dem Treffen mit dem Projektpartner RIF/MMI am 27.04.2010 thematisiert.

Für die Referenzexperimente entwickelte RIF/MMI eine Differential-Zwangsbedingung zur Nachbildung einer Seilwinde.

3.1.29 AP4330: Sensorik

Es wurden vom Partner RIF/MMI zunächst Simulationen von 3 verschiedenen Typen von Laserscannern realisiert, die auch real bei RIF/MMI vorliegen (Riegler, Sick, Zoller & Fröhlich).

Eine umfangreiche Menge an Sensoren auf Basis der Dynamiksimulation zur Ausgabe von Kräften und Momenten (in Verbindungen, Motoren, Kontakten) sowie kinematischer Größen (Winkel, Geschwindigkeiten, Beschleunigungen) wurde vom Partner RIF/MMI implementiert. So wurde beispielsweise das Modell eines IMU-Sensors implementiert, ebenso Winkelsensoren, Zug-Druck-Kraftsensoren und Drucksensoren.

Ein umfangreiches Laserscanner-Modell dient als Referenzmodell für das Sensorsimulations-Framework. Ein flexibles I/O-Netzwerk erlaubt die Konstruktion unterschiedlicher Signalpfade für unterschiedliche Datensensoren und –filter für einen Sensor.

Ebenso wurde ein allgemeines ideales Kameramodell implementiert.

3.1.30 AP4340: Antriebe

Das Gelenkmodell, das in AP3340 definiert wurde, wurde in Matlab/Simulink auf 24 Gelenke erweitert und diese Gelenke wurden in ADAMS mit der gesamten SpaceClimber-Kinematik verbunden (Abb. 38). In diesem Modell bewegt jeder Aktuator durch ein Getriebe den Rest des Roboter-Gelenks, inklusive der übrigen Verbindungen und Lasten des Roboters. Das Beinmodell wurde in ADAMS aufgebaut und über Co-Simulation in Simulink durch den Untersystemblock für Beine simuliert. Das Framework verbindet sich zuerst mit MONSTER, um die gewünschten Gelenkpositionen zu erzeugen, die dem gewünschten Bewegungsmuster entsprechen, das in HOBBIT ausgewählt wurde. Abbildung 39 zeigt typische Trajektorien für das Gelenk des SpaceClimber Roboters, die MONSTER für ein Vorwärts-Laufen erzeugt hat. Die gewünschten Trajektorien für 24 Gelenke dienen als Referenzeingaben für das Gelenkmodell. Die Gelenke werden entsprechend angesteuert und die resultierenden Gelenk-Geschwindigkeiten werden für den gesamten Roboter auf Adams/View übertragen, um den gesamten Roboter zu bewegen.

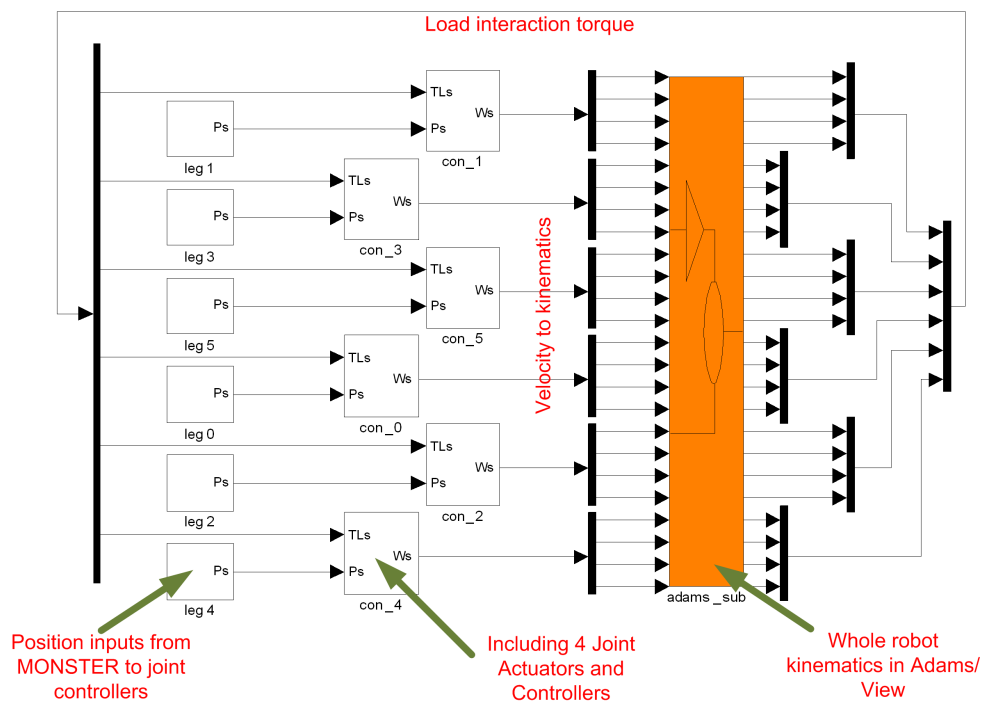


Abbildung 38: Das Gesamtmodell in der Matlab/Adams Co-Simulation

Es wurden zwei Experimente durchgeführt, um das Verhalten der High-Level Simulation mit dem echten Roboter zu vergleichen. Im ersten Experiment wurde ein Referenzlaufmuster aus MONSTER an den echten Roboter und an den simulierten Roboter gesendet, dann wurde das Verhalten des echten und des simulierten Roboters gemessen bezüglich der Gelenkposition, des PWM-Eingangssignals des Motors, der Motor Stromstärke und des Feedback-Drehmomentes.

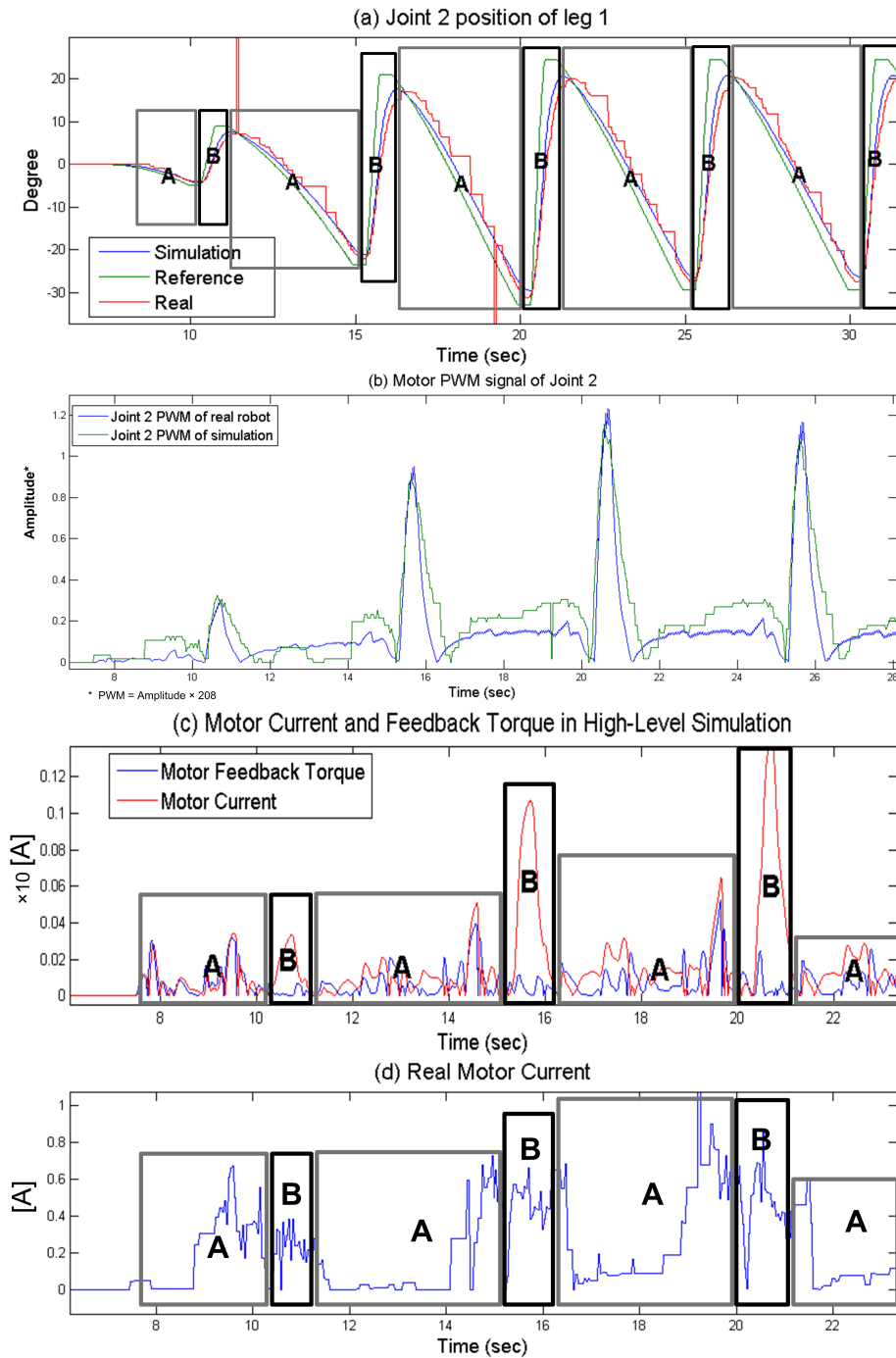


Abbildung 39: Die Ergebnisse des Laufverhaltens des ersten Beins.



Ein Vergleich zwischen einem Laufverhalten des echten Roboter-Gelenks mit den Simulationsergebnissen wird in Abb. 39(a) gezeigt. In der Standphase 'A' hat der Roboter-Fuß Kontakt mit dem Boden und in der Schwungphase 'B' ist der Roboter-Fuß in der Luft. Die Impulse im Signal der Position des echten Gelenks resultieren aus Messfehlern. Abb. 39(b) zeigt einen Vergleich des PWM Eingangssignals des echten Roboters mit dem Simulationsergebnis und Abb. Fig. 39(c) zeigt die Simulationsergebnisse der Gesamtstromstärke des Motors und des Feedback-Drehmomentes. Es ist bemerkenswert, dass in der Standphase 'A' die Stromstärke dem Feedback-Drehmoment folgt, während die Stromstärke in der Schwungphase 'B' nicht dem Feedback-Drehmoment folgt. Dieses Ergebnis ist sehr realistisch und nützlich um zu erkennen, ob der Roboter Kontakt mit einem Gegenstand hat oder nicht. Die Stromstärke des echten Gelenk-Motors wird in Abb. 39(d) gezeigt. Sie zeigt einen etwas anderen Kurvenverlauf als der simulierte Gelenk-Motor, weil die echte Kontaktdynamik zwischen dem Fuß und dem Boden anders ist als das Starrkörper-Kontaktmodell der High-Level-Simulation. Dies ist immer noch eine Herausforderung für unsere weitere Arbeit.

Um den SpaceClimber Motor zu simulieren benötigt man die vier in AP3340 definierten Blöcke: BLDC MotorBlock, Speed-Controller, Position-Controller und Low-Pass-Filter. Dazu wurden VEROSIM[®] einige Erweiterungen hinzugefügt. Neben einem Saturation- und RateLimiter-Plugin wurde auch der Low-Pass-Filter als VEROSIM[®] -Plugin entwickelt. Damit lassen sich nun auch der Speed- und Position-Controller direkt in VEROSIM[®] aus primitiveren Blöcken aufbauen. Diese Herangehensweise ermöglicht die flexible Anpassung der Controller ohne besondere Programmierkenntnisse. Für den letzte aus AP3340 fehlenden Block, den BLDC MotorBlock, wurde der generierte C-Code direkt durch ein VEROSIM[®] -Plugin gewrappt. Dies gewährleistet eine ausreichende Performanz und dass das geforderte Timing von 0.1 ms eingehalten wird.

3.1.31 AP4350: Bodenmechanik

Plastisches Terramechanik-Partikel in VEROSIM®

Der Plastisches Terramechanik-Partikel (PTP) Ansatz wurde zunächst in vereinfachter Form als Plugin in VEROSIM® implementiert. Dabei sind die Partikel in ihrer Bewegung auf die Z-Achse beschränkt und folgen der Pressure-Sinkage Relation von Bekker. Visualisiert wird der Boden durch eine Heightmap.

Mit diesem vereinfachten Modell ist es bereits möglich die Interaktion mit dem SpaceClimber in einer einfachen Testumgebung in Echtzeit zu simulieren. Dies wurde auch im Meilensteintreffen am 24. März in einer Live-Demonstration vorgeführt. In Abbildung 40 ist der SpaceClimber auf zwei unterschiedlich beschaffenen Böden, die dieses Modell nutzen, zu sehen.

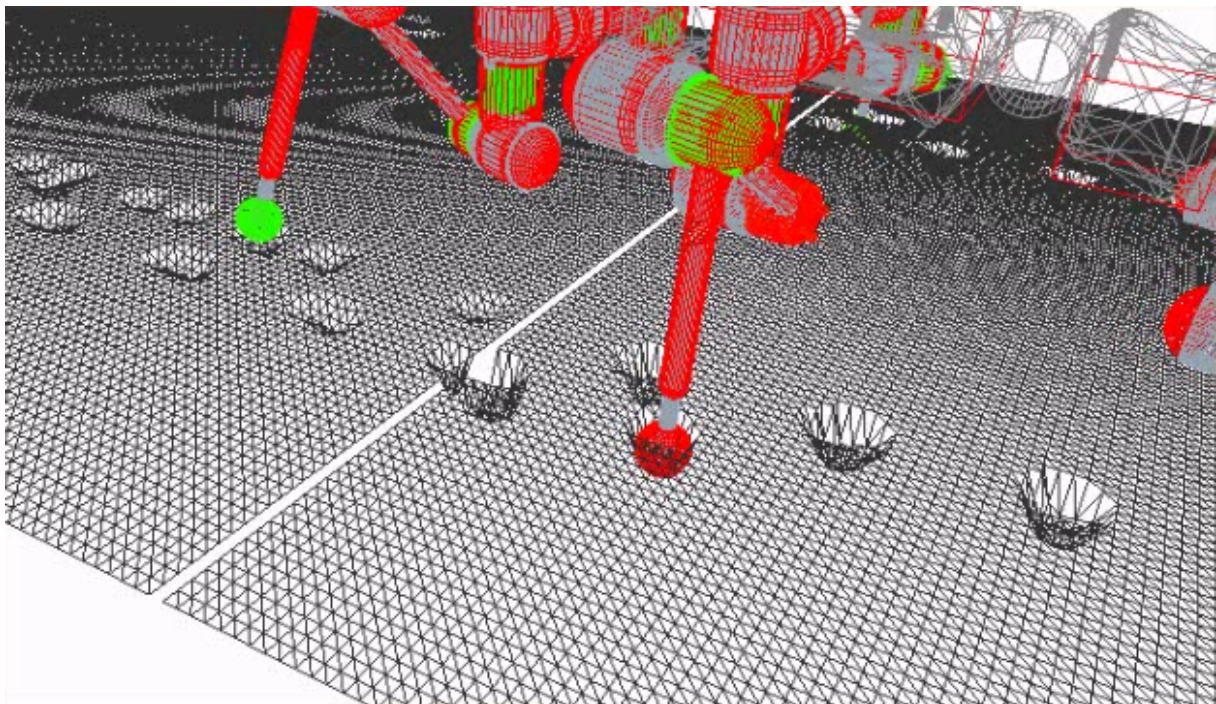


Abbildung 40: Der SpaceClimber auf dem vereinfachten PTP-Modell in VEROSIM®. Die rechte Hälfte des Bodens ist weicher als die Linke. Man sieht deutlich, dass der Roboter dort tiefer einsinkt.

Nach den viel versprechenden Ergebnissen des vereinfachten Modells, wurde die Implementation des Plastischen Terramechanik-Partikel (PTP) Ansatzes fortgesetzt. In Vorbereitung weiterer bodenmechanischer Effekte wie zum Beispiel Bulldozing wurde die Einschränkung auf vertikale Bewegungen der Partikel aufgehoben. Zusätzlich lassen sich nun beliebige Anzahl von Partikelschichten erzeugen. Zur Visualisierung wurde ein neues VEROSIM®-Plugin zum Rendern von Punkten implementiert, welches auch große Mengen an Punkten effizient darstellen kann.

Ein Problem frei beweglicher Partikel ist, dass eine naive Kollisionserkennung für jedes einzelne Partikel durchgeführt werden muss. Für große Mengen an Partikeln wie sie

für den PTP Ansatz benötigt werden ist dieser Zeitaufwand weit jenseits von dem was in einer Echtzeitsimulation möglich ist. Um die Performance des Ansatzes zu erhöhen wurde eine Erweiterung implementiert, die eine Einteilung der Partikel in disjunkte Bereiche erlaubt. Dadurch kann die Kollisionserkennung um mehrere Größenordnungen schneller durchgeführt werden, so dass der Ansatz wieder echtzeitfähig wird.

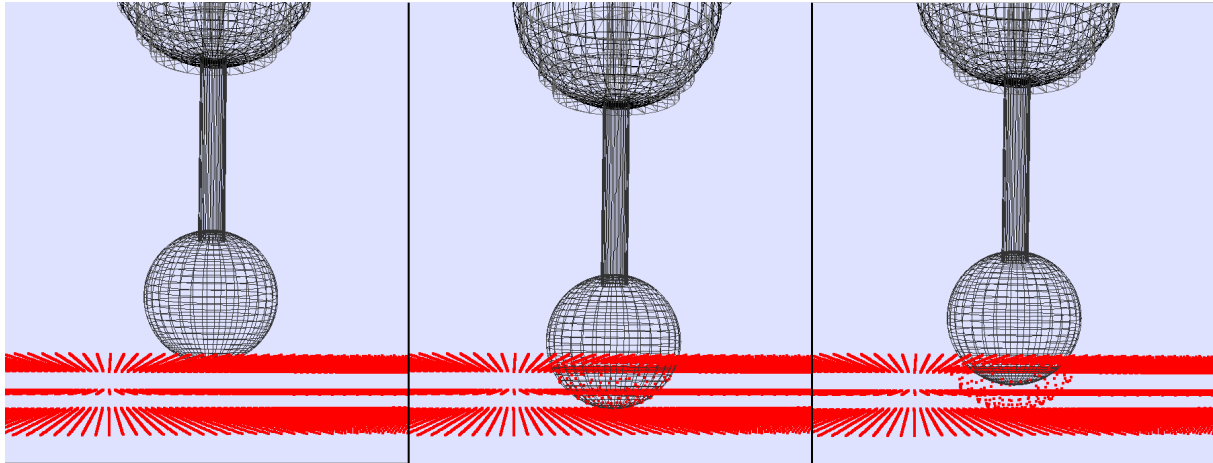


Abbildung 41: Das PTP-Modell in VEROSIM® bei der Durchführung eines Referenzexperiments.

Im folgenden wurde die Beschränkung auf vertikale Kräfte aufgehoben. Zusätzlich zur vertikalen Kraft

$$F_v(z) = A \cdot k \cdot z^n \quad , \quad (13)$$

mit A der Fläche eines Partikels, der Eindringtiefe z und der zwei Bodenparameter k und n , üben die einzelnen Partikel nun auch laterale Kräfte aus. Zum Einen wird die Reibung F_f basierend auf der Gleichung von Janosi und Hananoto berechnet,

$$F_f(F_v, j) = (A \cdot C_0 + F_v \cdot \tan(\phi)) (1 - e^{-j/K}) \quad (14)$$

wobei A wieder die Fläche eines Partikels und F_v die vertikale Kraft ist. C_0 , ϕ und K sind die Kohäsion, der Winkel der inneren Reibung und der Modulus der Scherverformung des Bodens. j ist die laterale Auslenkung des Partikels aus der ursprünglichen Lage. Zum Anderen üben die Partikel ein Kraft

$$F_b(j) = (V \cdot B) (1 - e^{-j/K}) \quad (15)$$

aus, die den Effekt des Bulldozing modellieren soll. Dabei ist V das Volumen das durch ein Partikel repräsentiert wird und B ein experimentell zu bestimmender Bulldozing-Parameter. j und K sind genau wie in Gleichung (14) definiert. Daraus ergibt sich dann eine resultierende laterale Kraft von

$$F_l(F_v, j) = (A \cdot C_0 + F_v \cdot \tan(\phi) + V \cdot B) (1 - e^{-j/K}) \quad . \quad (16)$$

Dabei ist zu beachten, dass es sich sowohl bei F_v als auch bei F_l um ein Maximalkräfte handelt, die von einem Partikel ausgeübt werden können. In der Simulation wird die

Summe der Partikelkräfte dann so beschränkt, dass sie den betreffenden Körper nur bremsen und maximal zum Stillstand bringen, ihn jedoch niemals beschleunigen.

Zur weiteren Steigerung der Performance wurde die Kollisionsdetektion zwischen den Partikeln und anderen Körpern in separate Threads ausgelagert. Dadurch kann die Kollisionsdetektion zum Beispiel beim SpaceClimber für alle sechs Füße weitestgehend parallel durchgeführt werden, vorausgesetzt es steht genügend Rechenleistung (CPUs) zur Verfügung.

Beim Vergleich der Simulationsergebnisse zu den lateralen Referenzexperimenten mit den realen Daten wurden einige Limitationen des bisherigen Ansatz mittels Plastischer Terramechnik Partikel (PTP) deutlich. Zum einen hat die Dichte der Partikel in der Simulation einen Einfluss auf die physikalischen Eigenschaften des Bodens. In dem einfachen Fall der vertikalen Kraft mit Exponent $n = 1$ ergibt sich der simulierte Druck zu

$$p(z)_{theo} = k \cdot z^n$$

$$\xrightarrow{n=1} p(z)_{sim} = \frac{k}{2} \cdot \left(\frac{z^2}{\Delta z} - z \right)$$

wobei Δz der vertikaler Abstand zweier Partikelschichten ist. Dieser Zusammenhang ist in Abbildung 42 veranschaulicht.

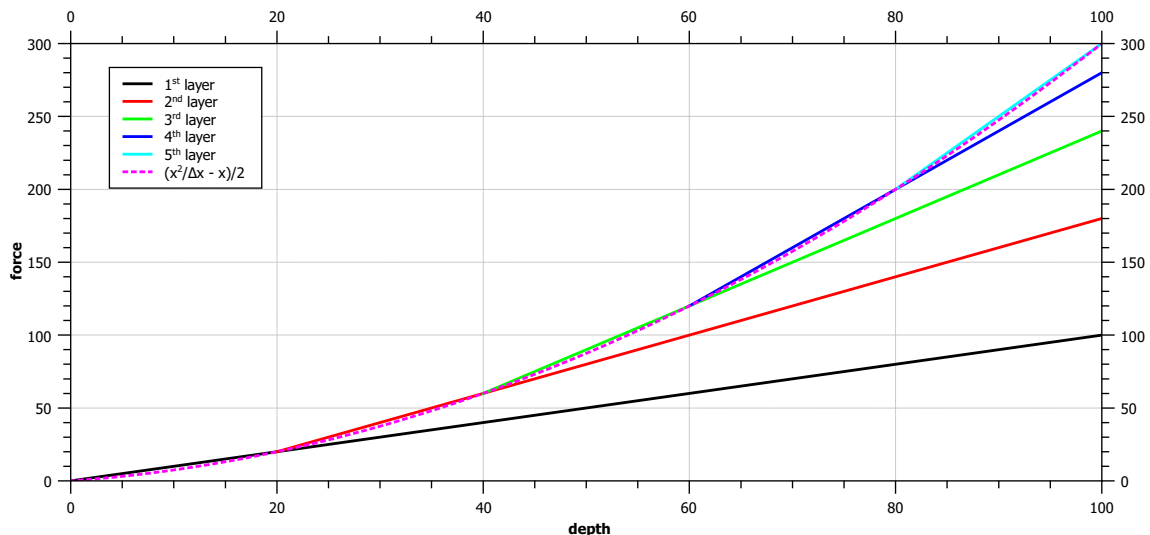


Abbildung 42: Auswirkung mehrerer Partikelschichten im PTP Modell. Hier ist die vertikale Kraft für $n = k = 1$ und $\Delta z = 20$ dargestellt.

Weitere Schwierigkeiten des Ansatzes sind die analytische Beschreibung der Kontaktkräfte und die präzise Bestimmung der dazu gehörigen Bodenparameter.

Um diese Limitationen zu beseitigen wurden einige Änderungen am PTP Modell vorgenommen. Zur Beseitigung der Dichteabhängigkeit des Druckes wird der Gegendruck des Bodens im neuen Modell nun nicht mehr von den Partikeln selbst erzeugt, sondern von virtuellen Zellen, die aus der Diskretisierung des Bodens hervorgehen und mit Partikeln gefüllt sind (vgl. Abb. 43).

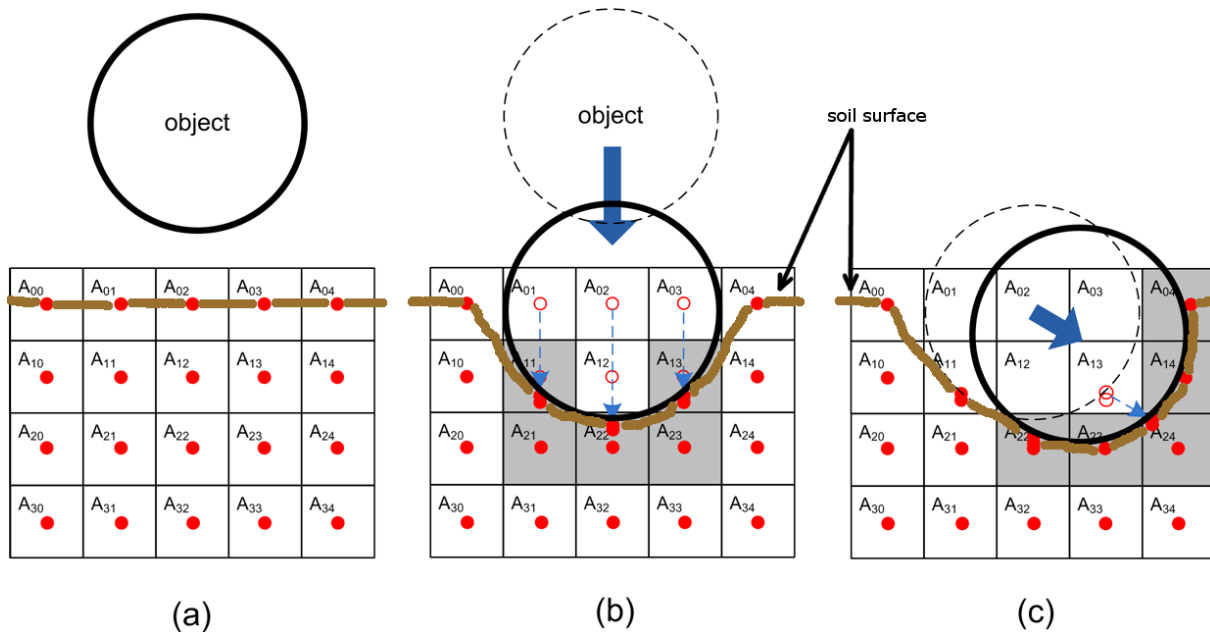


Abbildung 43: (a) Der Boden wird in drei Raumdimensionen diskretisiert (hier sind nur zwei dargestellt) und jede Zelle mit einem Partikel befüllt. (b) und (c) Objekte sind im Stande Partikel zu verschieben. Ist ein Objekt in Kontakt mit mindestens einem Partikel einer Zelle so übt diese Zelle eine Kraft auf den Körper aus. Die Größe dieser Kraft ist abhängig von der Anzahl der in der Zelle befindlichen Partikel.

Die Größe des Druckes einer Zelle, wird bei Kontakt mit einem Objekt durch die Anzahl der in der Zelle befindlichen Partikel bestimmt. Dadurch wird nicht nur der Boden diskretisiert, sondern auch die Gegenkraft des Bodens. So ist es möglich auf ein analytisches Modell der Bodenmechanik zu verzichten und die erforderlichen Größen statt dessen in einer Wertetabelle abzulegen. Konkret bestimmt man durch einfache Experimente einen Zusammenhang zwischen Eindringtiefe und Gegendruck und sampelt die Daten entsprechend der gewählten Partikeldichte. Dieser Prozess ist in [Abbildung 44](#) dargestellt.

Eine höhere Auflösung der Partikel geht so nur mit einer feineren Diskretisierung der Pressure-Sinkage Kurve und damit einer besseren Annäherung an die realen Gegebenheiten einher. Ein weiterer Vorteil dieser Methode ist, dass die Erzeugung einer solchen Wertetabelle beim Vorfinden eines bislang unbekannten Bodens während einer Mission vorgenommen werden kann.

In [Abbildung 45](#) wird ein exemplarisches Ergebnis dieses Bodenmechanikmodells gezeigt. Die Wertetabelle wurde dabei in einem Experiment mit einer flachen kreisförmigen Platte ($\varnothing = 7 \text{ cm}$) gewonnen, während das Experiment zur Validierung des Modells mit dem kugelförmigen SpaceClimber-Fuß durchgeführt wurde.

Neuronales Netz

Die Implementierung der neuronalen Netze wurde im ersten Quartal 2011 weitestgehend abgeschlossen. Als Basis dient ein einfaches Perzeptron mit Eingabegewichten,

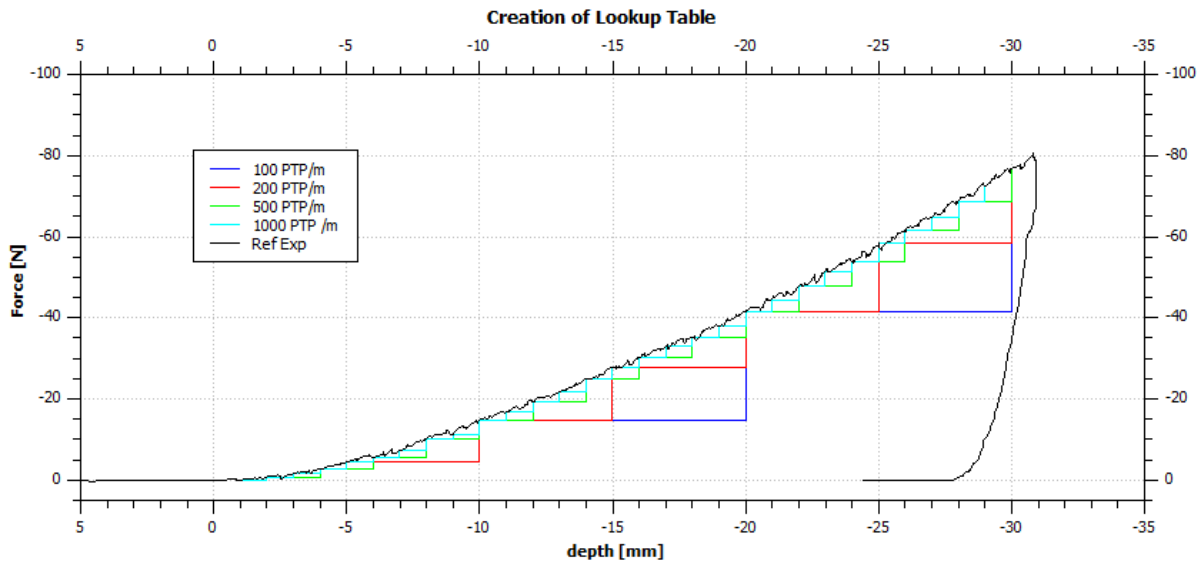


Abbildung 44: Sampeln einer Pressure-Sinkage Kurve für verschiedene Partikeldichten.

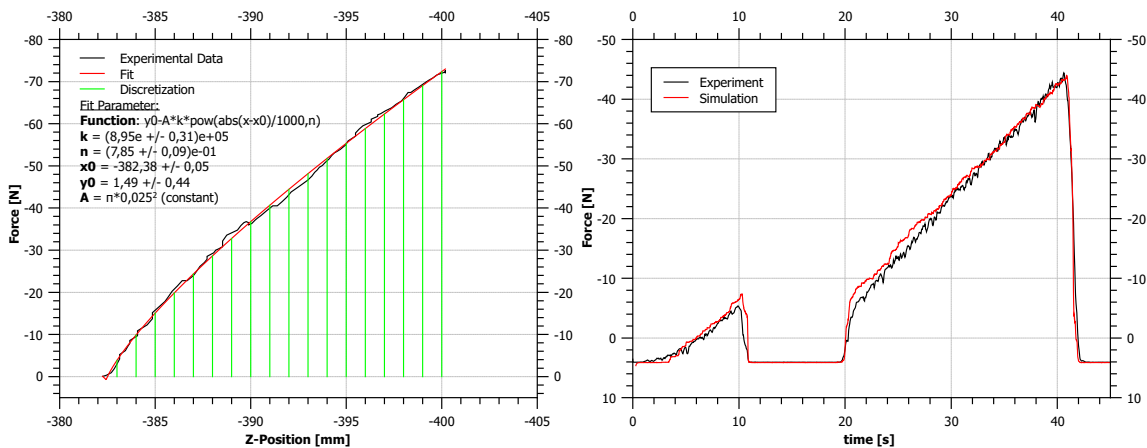


Abbildung 45: Links: Mit einer Platte ($\varnothing = 7$ cm) durchgeführtes Druckexperiment (schwarz) mit einem Fit (rot) und einer Diskretisierung (grün). Rechts: Druckexperiment mit einem Kugelfuß (schwarz) und Simulation des gleichen Experiments basierend auf der Wertetabelle aus der linken Grafik (rot).

Übertragungsfunktion, Aktivierungsfunktion und Ausgabegewicht. Abbildung 46 zeigt den Aufbau eines Neurons (Perzeptrons) graphisch.

Für die Optimierung der neuronalen Netze wurde eine bereits entwickelte generalisierete Implementierung der genetischen Algorithmen verwendet. Die Abbildung der Graphen (Phenotype) auf ein neuronales Netz ist beispielhaft in Abbildung 47 dargestellt. Welche Übertragungsfunktion verwendet wird und die Gewichte der Inputs sind in den Parametern der Knoten und Kanten des Genotypes kodiert. Erste einfache 1x1 Abbildungen wurden als Testfunktionen verwendet und die Ergebnisse sind in Abbildung 48 dargestellt.

Innerhalb des zweiten Quartals 2011 wurde ein Netzwerk in AP5300 an Hand der real

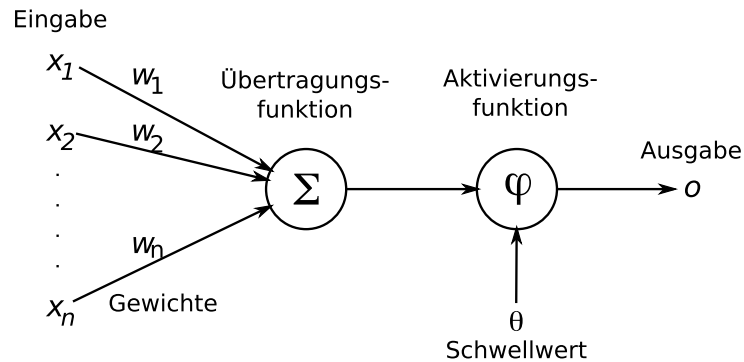


Abbildung 46: Perzeptron.

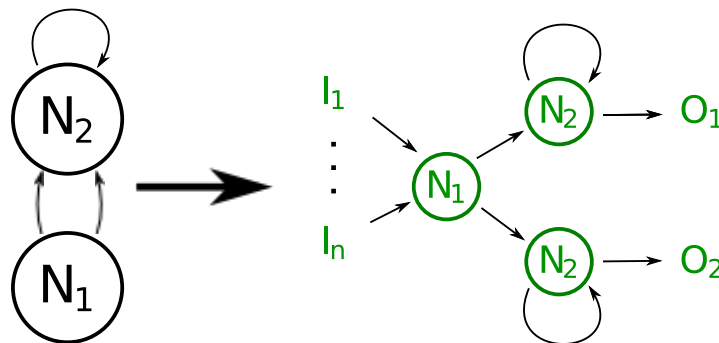


Abbildung 47: Kodierung.

gemessenen Referenzdaten optimiert. Um den Ansatz implementierungstechnisch abzuschließen wurde ein Plugin für VEROSIM[®] entwickelt, welches das neuronale Netz mit der Dynamiksimulation verbindet. Um eine bestmögliche Integration des neuronalen Netzes zu erzielen, werden die aus dem Netzwerk ermittelten Normalkräfte nicht direkt auf die Kollisionskörper angewendet sondern über die Materialeigenschaften indirekt in VEROSIM[®] aufgebracht. Auf diesem Weg können die Normalkräfte innerhalb eines Simulationsschrittes erzeugt werden und somit bei der Fehlerbehandlung berücksichtigt werden. Dies erhöht die Stabilität der Simulation bei der Kontaktbehandlung, erlaubt dadurch größere Simulationsrechenschritte, was wiederum eine höhere Simulationsgeschwindigkeit ermöglicht.

Zur Integration der Normalkräfte in die Materialeigenschaften werden die Kräfte in Federkonstanten umgerechnet. Diese ergibt sich aus der Normalkraft und der Eindringtiefe. Zusätzlich wurde ein Dämpfungsparameter gewählt der ein Schwingen der Kontakte verhindert.

Durch die Implementierung eines Fußabdruckes wird verhindert, dass Objekte bei verringerter Last wieder aus dem Boden gedrückt werden. Dafür wird für jeden Kontaktpunkt (Fußabdruck) die maximal entstandene Eindringtiefe gespeichert und die Eindringtiefe des Kontaktpunktes um die Fußabdrucktiefe reduziert. Für die Berechnung der Federkonstante wird dann die reduzierte Eindringtiefe verwendet. Jeder neue Fußabdruck wird mit einem zufälligen Varianzparameter in eine Liste eingefügt. Durch diese Implementierung können keine Objekte wieder aus dem Boden gedrückt werden

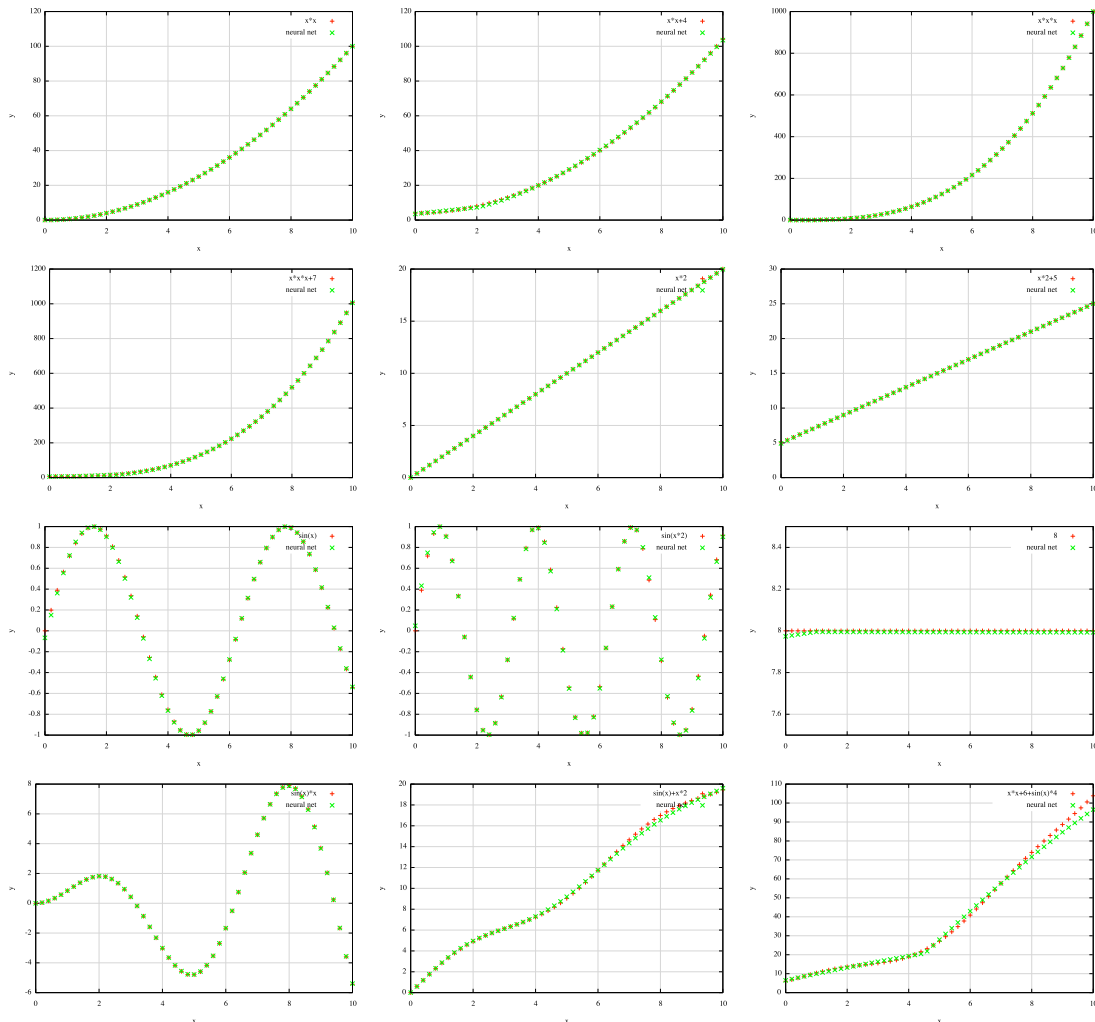


Abbildung 48: Testfunktionen für das neuronale Netz.

und auch das Wiedereindringen in einen Fußabdruck ist ermöglicht.

Visualisierung Fußabdrücke

Zum Visualisieren der Bodendynamik wird eine sehr hochaufgelöste Mesh benötigt, da viele kleine Flächen notwendig sind um die Verformung des Untergrundes bzw. Fußabdrücke im Sand darstellen zu können. Die benötigte Auflösung auf einer Fläche von 1x1 Meter kann performant in Echtzeit dargestellt werden. Größere Umgebungen, ca. ab 5x5 Meter, beinhalten bei der hohen Auflösung zu viele Flächen um sie in Echtzeit zu visualisieren.

Um dennoch in größeren Landschaften die Bodendynamik darstellen zu können wurde eine Methode implementiert, die dynamisch die Auflösung der Bodendarstellung anpasst (siehe Abbildung 49). Die Implementierung wurde zunächst in einem allein-stehenden kleinem OpenGL Program umgesetzt und in einem nächsten Schritt in

VEROSIM® integriert. Insgesamt wurden drei verschiedene Ansätze implementiert, die im Folgenden kurz erläutert werden.

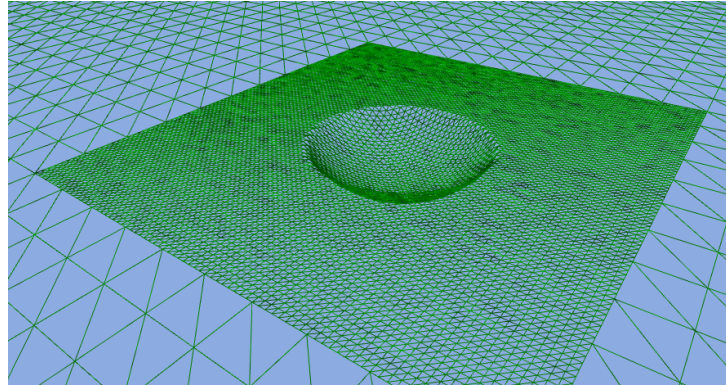


Abbildung 49: Beispiel der dynamischen Verfeinerung der Meshauflösung zum Darstellen der Bodenverformung.

1. GenericSurface: Bei der generischen Umsetzung wird die Mesh in neun Felder unterteilt, wobei das neue mittlere Feld am Fussmittelpunkt ausgerichtet ist (siehe Abbildung 50). Bei dieser Umsetzung wird die Mesh anfänglich optimal unterteilt, wenn aber weitere Unterteilungen der niedrig aufgelösten Felder notwendig werden, werden die Felder sehr unregelmäßig und im Endeffekt nicht optimal aufgeteilt. Zudem passen durch unterschiedliche Auflösungen die hochaufgelösten Felder nicht zwingend zueinander. Dieses wird an den Kanten bei zwei benachbarten Felder sichtbar.

L	L	L
L	H	L
L	L	L

Abbildung 50: Darstellung der generischen Aufteilung. Die mit L markierten Felder sind niedrig aufgelöst und das mit H markierte Feld besitzt eine hohe Auflösung zum Darstellen der Fussabdrücke.

2. GridSurface: Bei der Gitter basierten Umsetzung wird eine feste Auflösung für die hochaufgelösten Felder gewählt. Dabei ist die Kantenlänge der groben Felder ein vielfaches der Kantenlänge der hochaufgelösten Felder. Durch diese Implementierung passen die Kanten zweier benachbarten hochaufgelösten Felder zueinander und es entstehen keine sichtbaren Unregelmäßigkeiten. Abbildung 51 veranschaulicht den Gitter basierten Ansatz. Der Nachteil dieses Ansatzes ist die maximale Größe der Zellen, die sich aus der Auflösung der feinen Felder ergibt. Dadurch ist die Größe der Umgebung wieder durch die Anzahl der darstellbaren Felder beschränkt.

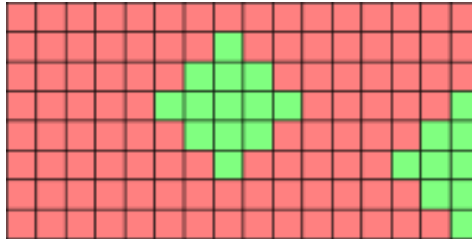


Abbildung 51: Prinzip der Gitter basierten Umsetzung. Die rot markierten Felder sind grob aufgelöst während die grünen Felder eine hohe Auflösung haben.

3. GenericGrid Diese Umsetzung kombiniert die beide vorherigen Ansätze. Grundsätzlich wird die Mesh wie bei dem generischem Ansatz aufgeteilt. Dabei werden die Eckpunkte der einzelnen Felder immer auf ein fiktives Gitter gelegt (siehe Abbildung 52). Dadurch werden die Vorteile beider Ansätze kombiniert.

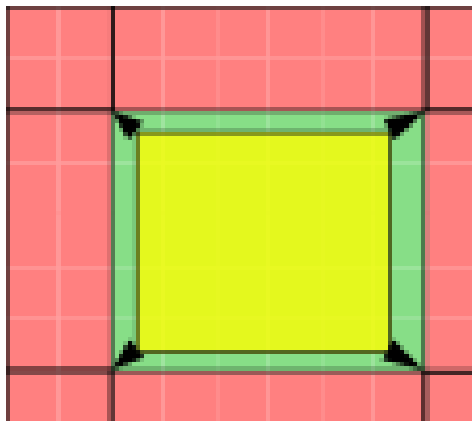


Abbildung 52: Prinzip der generischen Gitter basierten Umsetzung. Die rot markierten Felder sind grob aufgelöst. Das gelbe Feld ist der angeforderte hochauflösende Bereich der auf das Gitter gestreckt wird (um die grünen Bereiche erweitert wird).

Dieses Konzept zur dynamischen Anpassung der Heightmapauflösung wurde nachfolgend als VEROSIM[®]-Plugin umgesetzt. Die Kollisionserkennung wird dabei nach wie vor von einer VEROSIM[®] Heightmap durchgeführt, die jedoch nicht visualisiert wird. Eine erkannte Kollision wird dann an das entsprechende Bodenmechanikplugin weitergeleitet um die entsprechenden Kräfte zu berechnen und in die Simulation einzubringen. Das Bodenmechanikplugin teilt die Eindringtiefe und -position dem Visualisierungsplugin mit, welches sich dann um die dynamische Anpassung der Auflösung und effiziente Darstellung mittels OpenGL kümmert. Ein einfaches Beispiel der dynamischen Heightmap ist in Abbildung 53 dargestellt.

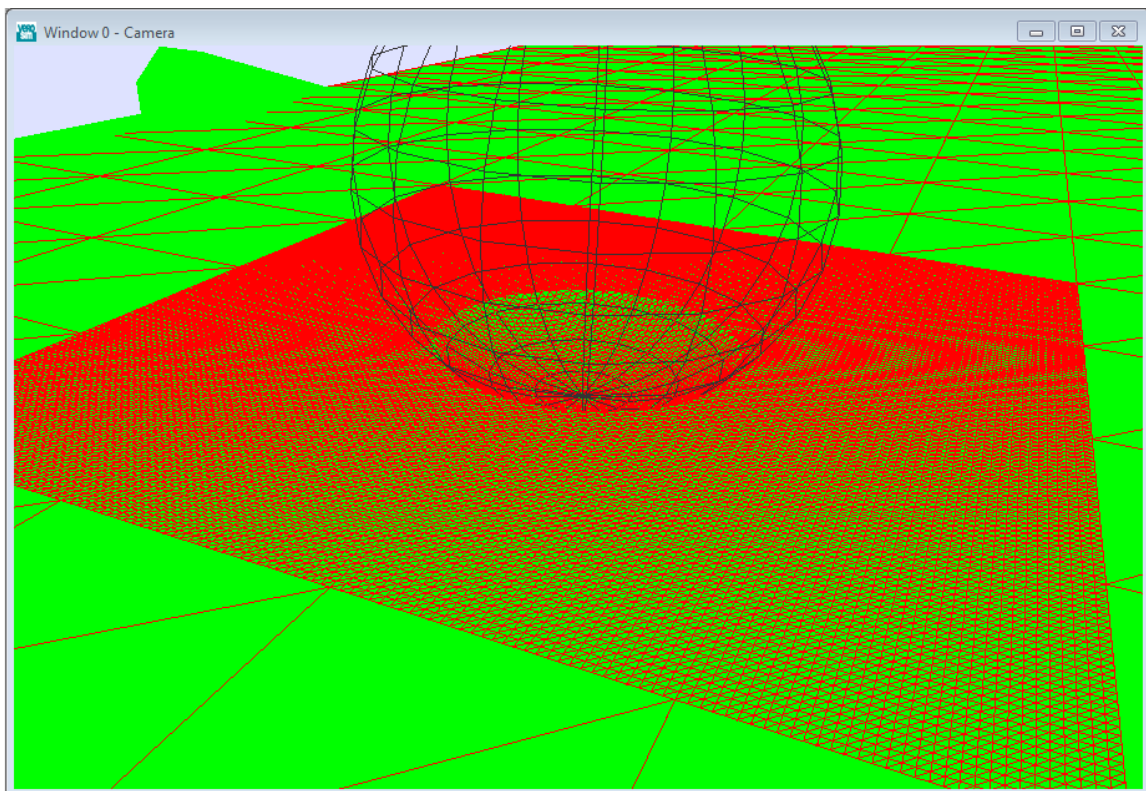


Abbildung 53: Die grob aufgelöste Heightmap (außen) wird an Kollisionpunkten lokal durch eine hochaufgelöste Heightmap ersetzt (mitte). Dies ermöglicht eine detaillierte Darstellung der Fußabdrücke selbst in sehr großen Umgebungen.



3.1.32 AP4360: DFKI-CAD-Modellimport

Es können über das VEROSIM[®]-Inventor-Plugin Geometrie & Materialien von Inventor nach VEROSIM[®] exportiert werden. Kinematische Constraints erfordern einen größeren Aufwand. Bei einfachen Beispielen mit DH-Parametern funktioniert der Export der Constraints.

Das Inventor-VEROSIM[®]-Plugin wurde von MMI/RIF anhand eines Referenzmodells (Scarabeus-Bein) erweitert, so dass die im Ausgangsmodell enthaltenen, „abstrakten“ Inventor-Constraints in technisch orientierte Dynamik-Gelenktypen in VEROSIM[®] (Bsp.: „Drehgelenk“) übersetzt werden können. Als physikalische Grundform wird derzeit eine Approximation anhand der achsenorientierten Hüllkörperboxen (AABBs) genutzt. Damit ist die automatische Generierung eines VEROSIM[®]-Simulationsmodells aus einem Inventor-CAD-Modell möglich.

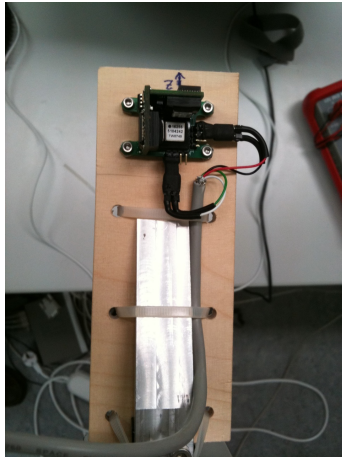
3.1.33 AP4400: Modellierung / Aufbau Referenz-experimente (real & virtuell)

3.1.34 AP4410: Aufbau der Referenzexperimente im realen Testbed

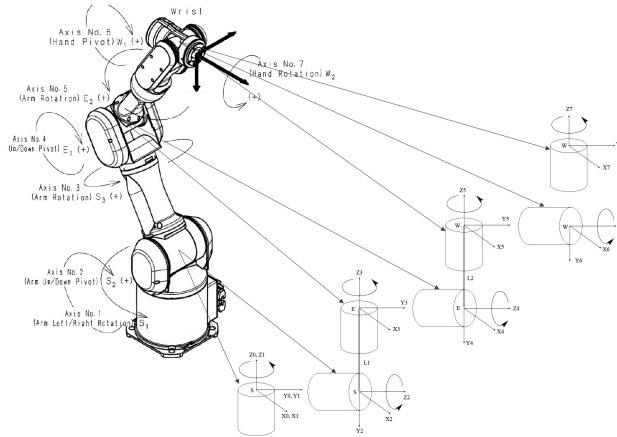
Sensoren Im Rahmen dieses Arbeitspaketes wurde recherchiert, zu welchen der ausgewählten Sensoren bereits Experimente durchgeführt wurden, und anschließend wurden die Ergebnisse beziehungsweise die verwendeten Tools zusammengetragen. Zusätzlich wurde sichergestellt, dass alle weiteren benötigten Experimente direkt am DFKI durchgeführt werden können.

Inertialsensoren LIS3LV02DL und DFKI-IMU AM DFKI wurden bereits Referenzexperimente mit der DFKI-IMU und einem kommerziellen Inertialsensor (Xsens) durchgeführt. Dazu wurden beide Sensoren am Endeffektor des Roboterarms Mitsubishi PA10 befestigt und vorgegebenen Lagen, Drehraten und Trajektorien angesteuert. Anschließend wurden die Werte der getesteten IMUs mit der Lage des Endeffektors des Roboterarms verglichen. Abbildung 54(a) zeigt die DFKI-IMU, wie sie beim Experiment genutzt wurde, und Abbildung 54(b) eine Skizze des verwendeten Roboterarms. Für Experimente mit dem drei-achsigen LIS3LV02DL ist sowohl die Hardware als auch die zum Auslesen benötigte Software vorhanden (siehe *Drucksensor MS5541C*).

Winkelsensor SpaceClimber Gelenk Zu dem im SpaceClimber eingesetzten Gelenken ist die SCJointControlGUI vorhanden, mit dem alle relevanten Daten der Gelenke ausgelesen und graphisch visualisiert werden können. Auf diese Weise können sehr genaue Referenzwerte für die Simulation gesammelt werden. Abbildung 55(a) zeigt die GUI und 55(b) einen damit generierten Beispiels-Graphen, der die Positionsveränderung bei einem PWM-Wert von 250 visualisiert.

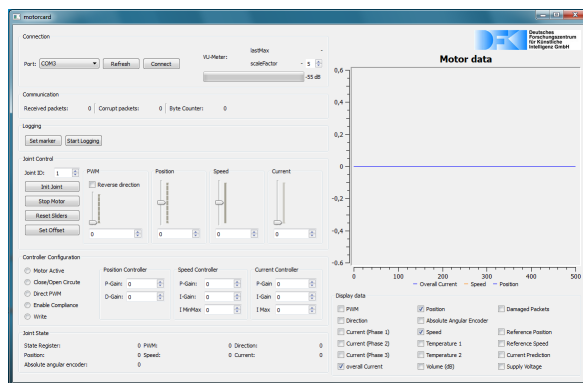


(a) DFKI-IMU im Testeinsatz

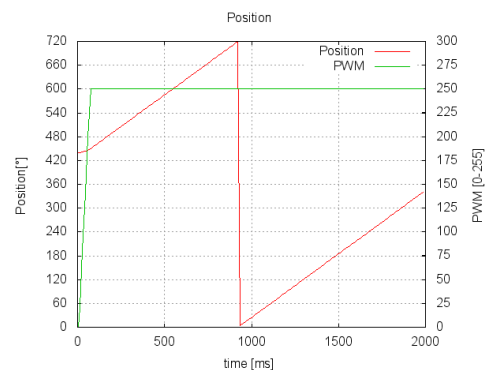


(b) Mitsubishi PA10

Abbildung 54: IMU Testumgebung



(a) Control GUI



(b) PMW-Positions Graph

Abbildung 55: SpaceClimber Gelenk

Zug-Druck-Kraftsensor ATI MINI45 Um mit den ATI MINI45 zu arbeiten, wurde im Rahmen des SpaceClimber Projektes die ForceTorqueSensorGUI implementiert. Sie ist in der Lage über ein zusätzliches Sensor-Board 6-achsige Kraftsensoren auszulesen und die Daten aufzuzeichnen. Sie eignet sich ideal dazu, um das Rauschverhalten der ATI MINI45 zu ermitteln. Abbildung 56 zeigt die Oberfläche der ForceTorqueSensorGUI.

Drucksensor MS5541C Die im SpaceClimber verbauten Sensoren, zu denen neben den vier Drucksensoren auch vier Dehnungsmessstreifen, ein Temperatursensor und der 3-Achsen-Beschleunigungssensor LIS3LV02DL gehört, können mit der *Foot Sensor Visualizer GUI* ausgelesen werden. Da die Drucksensoren direkt im Fuß eingegossen sind, wird ihr Verhalten zusammen mit den Beinexperimenten ermittelt, weswegen der dort genutzte Testaufbau verwendet wird. Abbildung 57 zeigt die Foot Sensor Visualizer GUI.

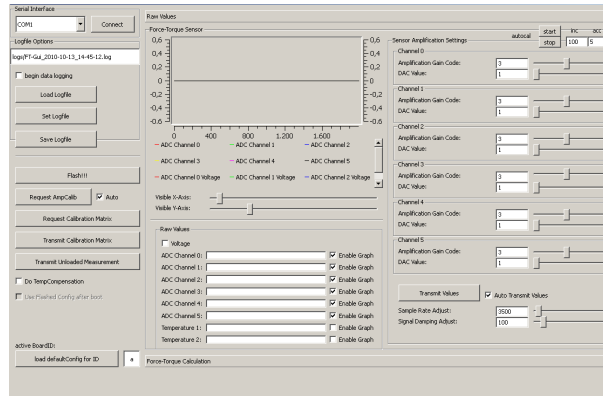


Abbildung 56: Force Torque Sensor GUI

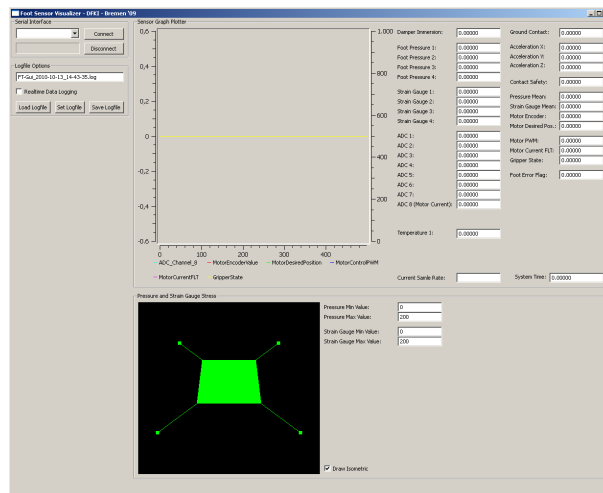


Abbildung 57: Foot Sensor Visualizer GUI

Laserscanner Hokuyo URG-04LX Zum Laserscanner Hokuyo URG-04LX wurden bereits im Rahmen des Projektes LUNARES Experimente durchgeführt, wozu der in Abbildung 58 dargestellte Teststand benutzt wurde. Aufgrund Ungenauigkeiten in den Messungen, resultierend aus einer nicht-festen Scanner-Position, wurde dieser Teststand verbessert und kann für weitere Experimente eingesetzt werden. Neben dem dargestellten farbigen Teststreifen sind Teststreifen mit unterschiedlichen Grauverläufen vorhanden. Da aber neben den Farbexperimenten zusätzlich das Rauschverhalten des Scanners bei unterschiedlichen Entfernungen experimentell ermittelt werden soll, wurde der in AP5410 beschriebene Versuchsaufbau genutzt.

Referenzexperimente über Messungen von Drehmoment und Kraft

Um die Modelle und Simulationen in Bezug auf Kraft- und Drehmoment-Interaktionen näher zu verifizieren, werden entsprechende Referenzexperimente aufgebaut. Der erste Experimentaufbau verwendet ein SpaceClimber Gelenk, das mit einem ATI Mini45 sechs Achsen Kraft- und Drehmoment-Sensor ausgestattet ist, der als Verbin-

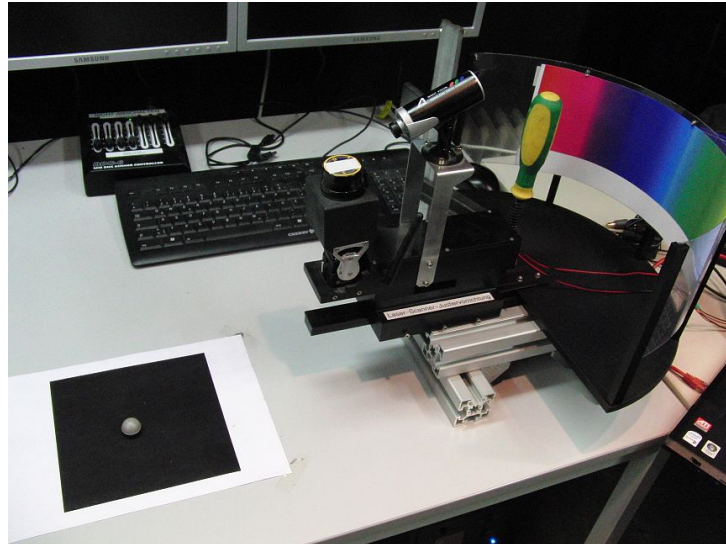


Abbildung 58: Angefertigter Hokuyo-Teststand

dungselement zwischen dem Körper und dem Schultergelenk jedes Beins verwendet wird (siehe [Abbildung 59](#)).



Abbildung 59: SpaceClimber-Gelenk, ausgerüstet mit einem ATI Mini45 sechs Achsen Kraft- und Drehmoment-Sensor

Das ATI Sensorsystem FT9446, Mini45 SI-580-20 (gezeigt in [Abbildung 60](#)) ist ein robuster und leichter Sensor mit einem Durchmesser von 45mm, einer Höhe von 16mm und einem Gewicht von 90g. Er kann Drehmomente bis zu 20Nm auf allen Achsen messen und Kräfte bis 580N auf der X- und Y-Achse, und bis zu 1160N auf der Z-Achse. Die maximale Überlastung ist ein Drehmoment von 110Nm und eine Kraft von 5100N auf allen Achsen.

Das PCB-Board des Kraft-Drehmoment-Sensors sendet ständig Daten an das Suzaku-S FPGA über eine unabhängige LVDS-Leitung. Die gemessenen Kräfte und Drehmomente werden in Registern gespeichert, auf die über Software zugegriffen werden kann.

Um die gewünschten Referenzexperimente vorzubereiten und Kraft und Drehmoment



Abbildung 60: ATI Mini45 sechs Achsen Kraft- und Drehmoment-Sensor

auf den Sensor innerhalb seines Betriebsbereiches aufzubringen, wird das Experiment wie in Abbildung 61 gezeigt aufgebaut.

In diesem Aufbau werden das Gelenk und der Kraft-Drehmoment-Sensor über zwei verschiedene serielle Interfaces jeweils mit der Gelenk-Überwachungs-GUI und der Kraft-Drehmoment-Sensor-GUI verbunden. Die Gelenk-Überwachungs-GUI wird verwendet, um den Gelenkmotor zu steuern und seine Sensorwerte zu lesen.

Ein Modell mit verschiedenen Verbindungen und Bausteinen wird in MATLAB/Simulink aufgebaut, das sich über Standard-TCP/IP-Protokolle mit den beiden GUIs verbindet. Dies erleichtert den Entwurf und den Aufbau weiterer Experimente und ermöglicht das Testen verschiedener Szenarien.

Die Daten der Experimente werden automatisch grafisch auf dem Bildschirm protokolliert und für weitere Analysen in Dateien gespeichert

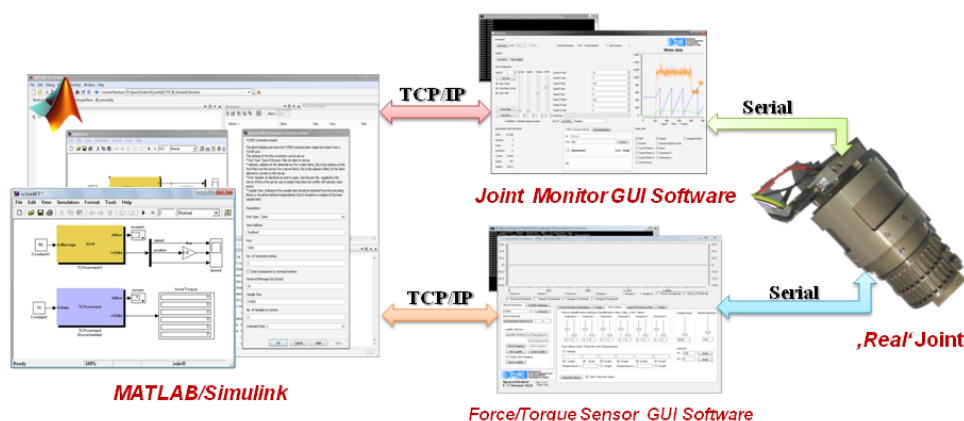


Abbildung 61: Drehmoment-Kraft-Sensor

Schunk Arm

Für die Bodendynamikexperimente soll der Schunk Roboterarm verwendet werden. Bisher wurde immer der Mitsubishi PA-10 vorgeschlagen, der aber durch Nutzung von anderen Projekten im DFKI schon relativ stark ausgelastet ist. Der Schunk Robo-

terarm hat wie der Mitsubishi sieben Freiheitsgrade und eine Kraftmessdose am Endeffektor. Unter dem Windows Betriebssystem stehen Treiber und eine Lösung der Inversen Kinematik bereits zur Verfügung. Außerdem existiert eine Steuerungssoftware die sowohl an den realen Roboter angebunden werden kann, als auch an ein Modell des Roboters in der Simulation. Somit kann die Steuerung des Schunk Roboters vollständig in der Simulation entwickelt und getestet werden bevor die Software an das reale System gekoppelt wird. Abbildung 62 zeigt ein Foto des realen Schunk Roboters und Abbildung 63 ein Bildschirmfoto aus der Simulation. In Abbildung 64 ist eine erste Steuerungssoftware dargestellt die sowohl an den realen als auch an den simulierten Roboter angebunden werden kann.



Abbildung 62: Realer Schunk Roboterarm mit sieben Freiheitsgraden.

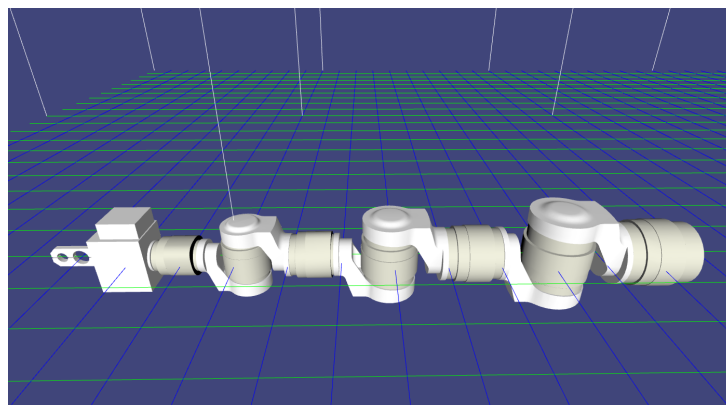


Abbildung 63: Bildschirmfoto des simulierten Schunk Roboterarms.

Bodendynamik Test Bed

Zur Durchführung der Bodendynamik-Referenzexperimente wurde die im AP3400 definierte Testumgebung aufgebaut. Die Testumgebung ist in Abbildung 65 zu sehen und beinhaltet einen Schunk LWA3 Robotikarm des *Mr. SemProm* sowie eine schwarze Kiste mit den Ausmaßen von 1m Länge x 1m Breite x 40cm Höhe. Die Kiste wurde mit

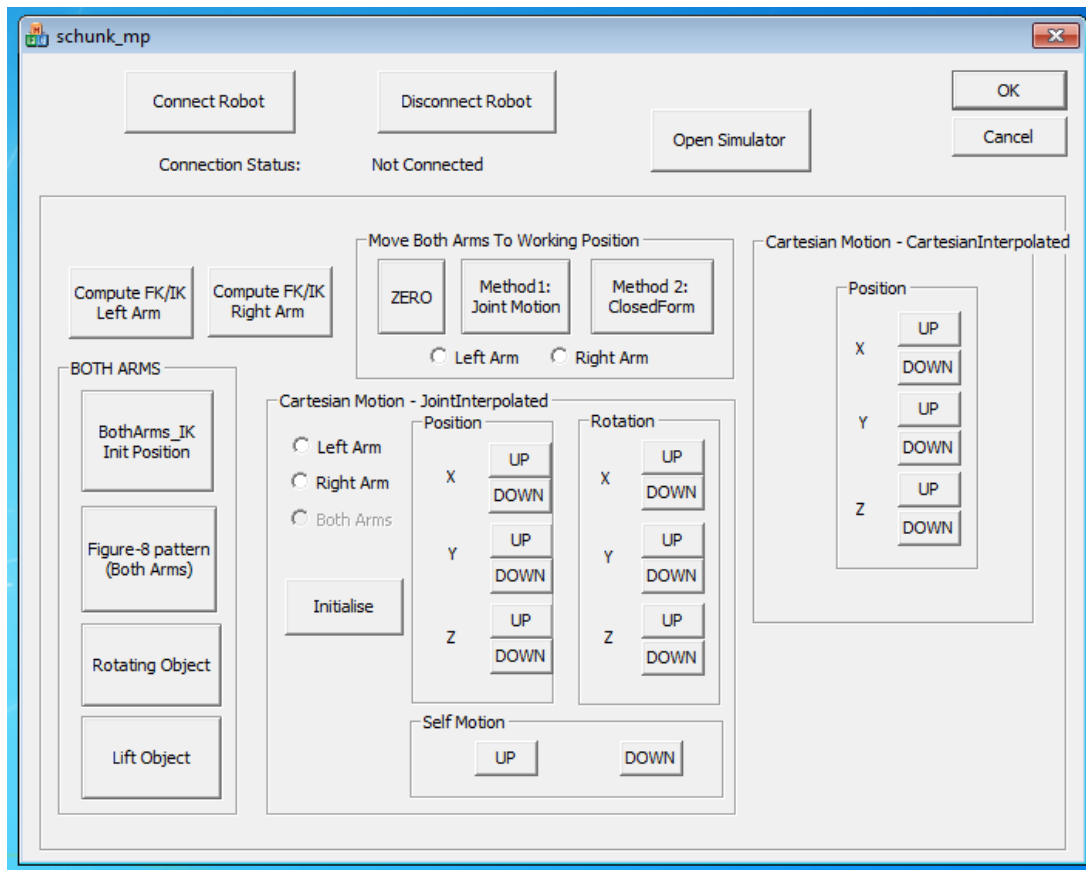


Abbildung 64: Erste Steuerungssoftware für den Schunk Roboterarm.

einem Substrat gefüllt, welches ähnliche Eigenschaften aufweist wie der Mond Regolith.

Für den Schunk LWA3 Robotikarm wurde ein Adapter entworfen um die Füße des SpaceClimber Roboters befestigen zu können. Über diesen Adapter können ebenfalls beliebige Geometrien (Zylinder, Kugeln oder weitere Füße) montiert und getestet werden. Der Adapter ist über eine 6-Achsen Kraftmessdose mit dem Roboterarm verbunden, wodurch das Aufnehmen von den entstehende Kontaktkräften ermöglicht wird. Abbildung 66 zeigt ein Foto der Fuss Montage, während in Abbildung 67 der CAD Entwurf des Adapters dargestellt wird.

Entwicklung der Kontrollsoftware

Für die Ansteuerung des Roboterarms wurde eine Bibliothek entwickelt, die es weiteren Projekten erlaubt, die in VirtualCrater entwickelte Ansteuerung mit geringem Aufwand weiter zu verwenden. Grundlage dieser Bibliothek ist die im Projekt SemProm entwickelte Roboteransteuerung. Zudem wurde eine grafische Benutzeroberfläche entworfen um das System zu testen und die benötigten Referenzexperimente durchführen zu können. Im Weiteren werden beide Softwarekomponenten vorgestellt.

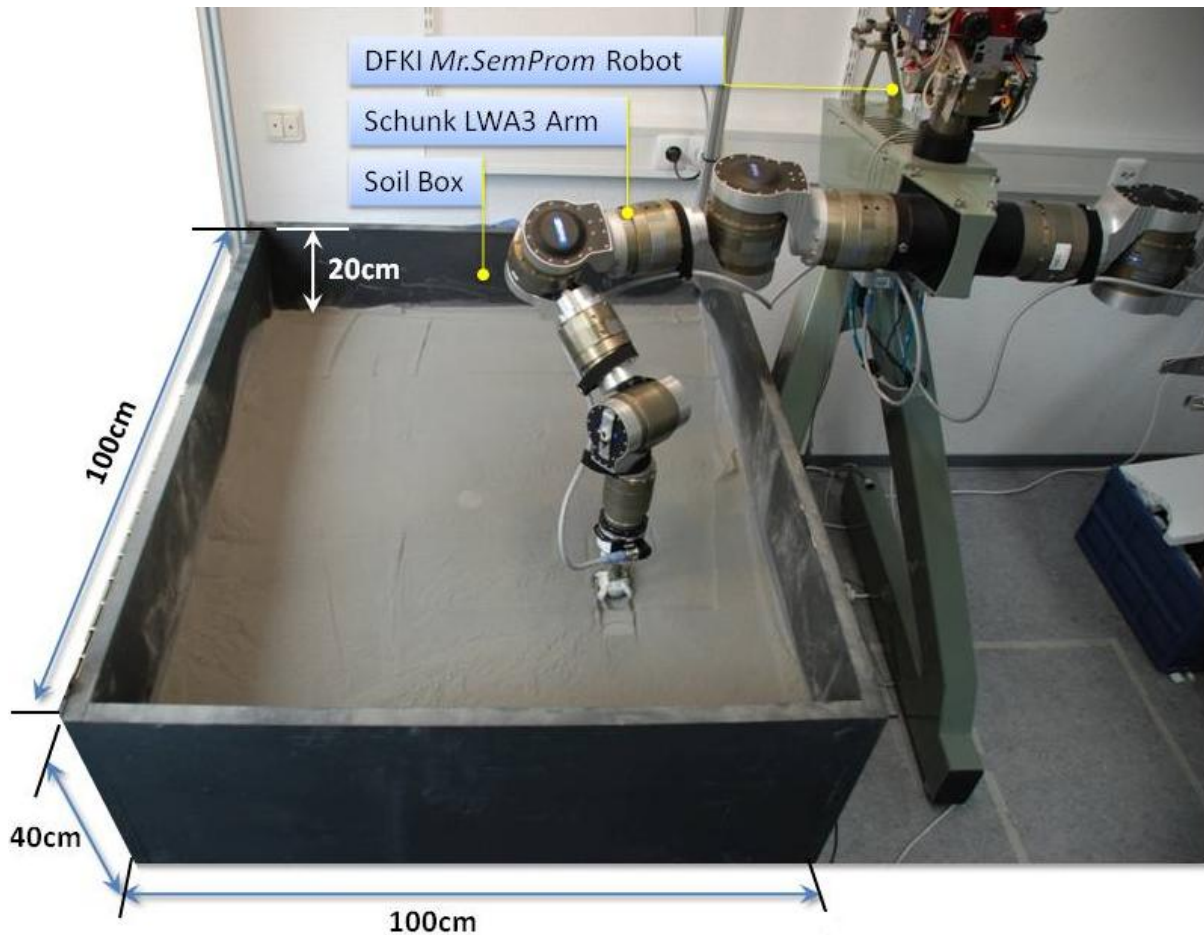


Abbildung 65: Details und Größen des Bodendynamik-Testbeds

Bibliothek zur Ansteuerung des SemProm Roboters Die Bibliothek verwendet drei weitere Bibliotheken für die Kommunikation mit dem Roboter, die Berechnung der Inversen Kinematik und eine Implementierung der POSIX Threads unter Windows. Abbildung 68 stellt die Architektur und den bisherigen Stand der Schunk-Bibliothek schematisch dar. Grundsätzlich wird über den Connect-Befehl wahlweise eine Verbindung zum dem realen Roboter oder zu einer Simulation des Roboters hergestellt. Nach erfolgreichem Verbindungsaufbau wird direkt der Kontroll-Thread gestartet, der für die Kommunikation mit dem Roboter zuständig ist. Über die Bibliothek können Punkte im kartesischen Raum oder im Gelenkraum unter Angabe einer Anfahrtszeit in Millisekunden und eines Flags für den rechten bzw. linken Roboterarm hinzugefügt werden. Die einzelnen Punkte werden immer in Gelenkwerte-Format in eine Liste eingetragen die von dem Kontroll-Thread abgearbeitet wird. Ist ein Punkt in der Liste vorhanden wird dieser mittels Gelenkwinkel-Interpolation über den gegebenen Zeitraum abgearbeitet und anschließend aus der Liste entfernt. Beim Beenden der Verbindung zu dem Roboter wird auch der Kontroll-Thread wieder beendet. Die erste Implementierung der SchunkControl Bibliothek wurde bereits sowohl in der Simulation als auch mit dem realen Robotersystem getestet. In nächsten Schritten soll die Gelenkwinkel-Interpolation optimiert werden und die Bibliothek um einige Kontrollfunktionen erweitert werden.

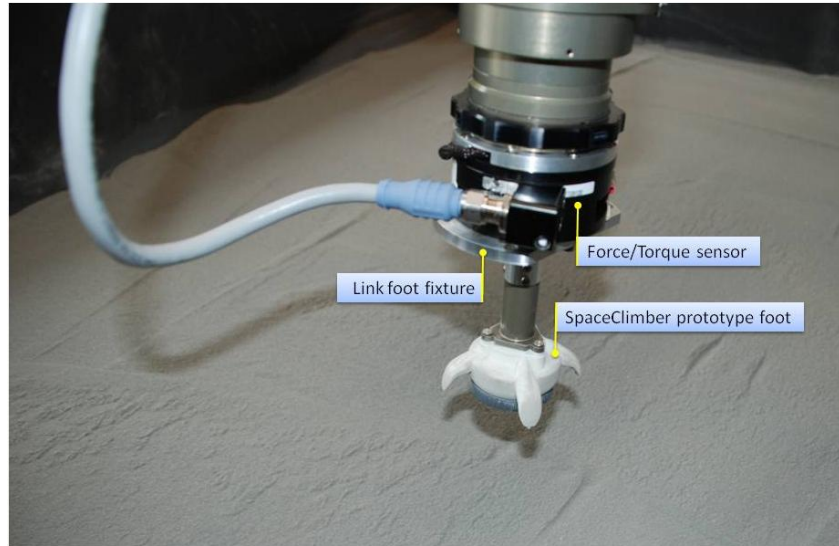


Abbildung 66: SpaceClimber Fuß verbunden mit dem Mr. SemProm-Arm im Testbed

Grafische Benutzeroberfläche zur Ansteuerung des SemProm Roboters Die grafische Benutzeroberfläche bietet unter Verwendung der SchunkControl Bibliothek die Möglichkeit sich wahlweise mit der Simulation oder dem realen System zu verbinden und danach die Arme des Mr. SemProm zu kontrollieren. Ein Bildschirmfoto der Benutzeroberfläche ist in [Abbildung 69](#) dargestellt. [Abbildung 70](#) zeigt die Simulation der beiden Roboterarme. Über die Benutzeroberfläche können zwei feste Positionen der Arme angesteuert werden (die Nullstellung der Gelenke und eine initiale Haltung der Arme). Zudem kann ein inkrementeller Wert auf die End-Effektor Position gegeben werden und eine Verbindung zu der Kraftmessdose zum Visualisieren und Loggen der Kraftdaten wurde ebenfalls implementiert. Das Verwalten von mehreren Punkte, die als Trajektorie von dem Roboter abgefahren werden sollen, ist bereits in der Benutzeroberfläche vorbereitet und wird in einem der nächsten Schritte fertiggestellt.

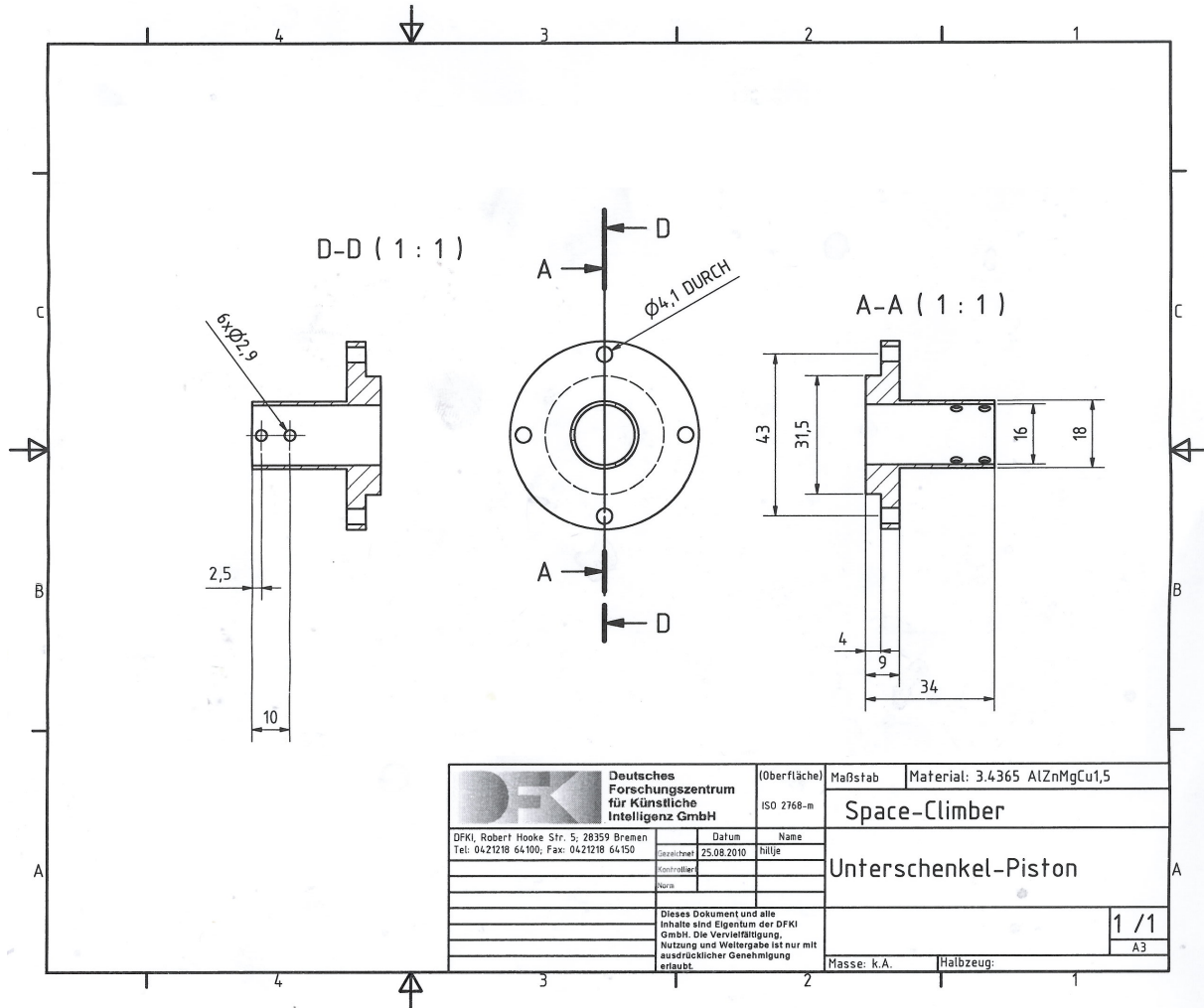


Abbildung 67: Details des Fußadapters

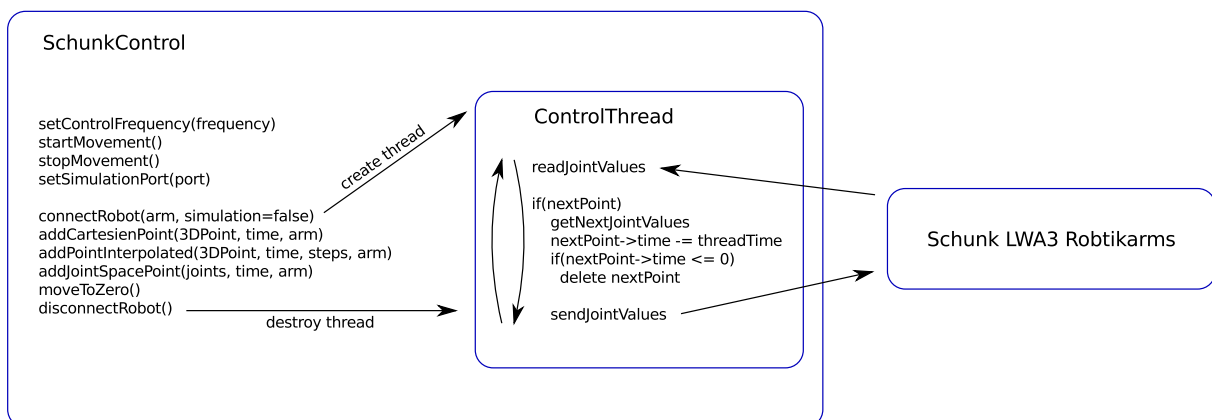


Abbildung 68: Schematische Darstellung der Schunk-Bibliothek.

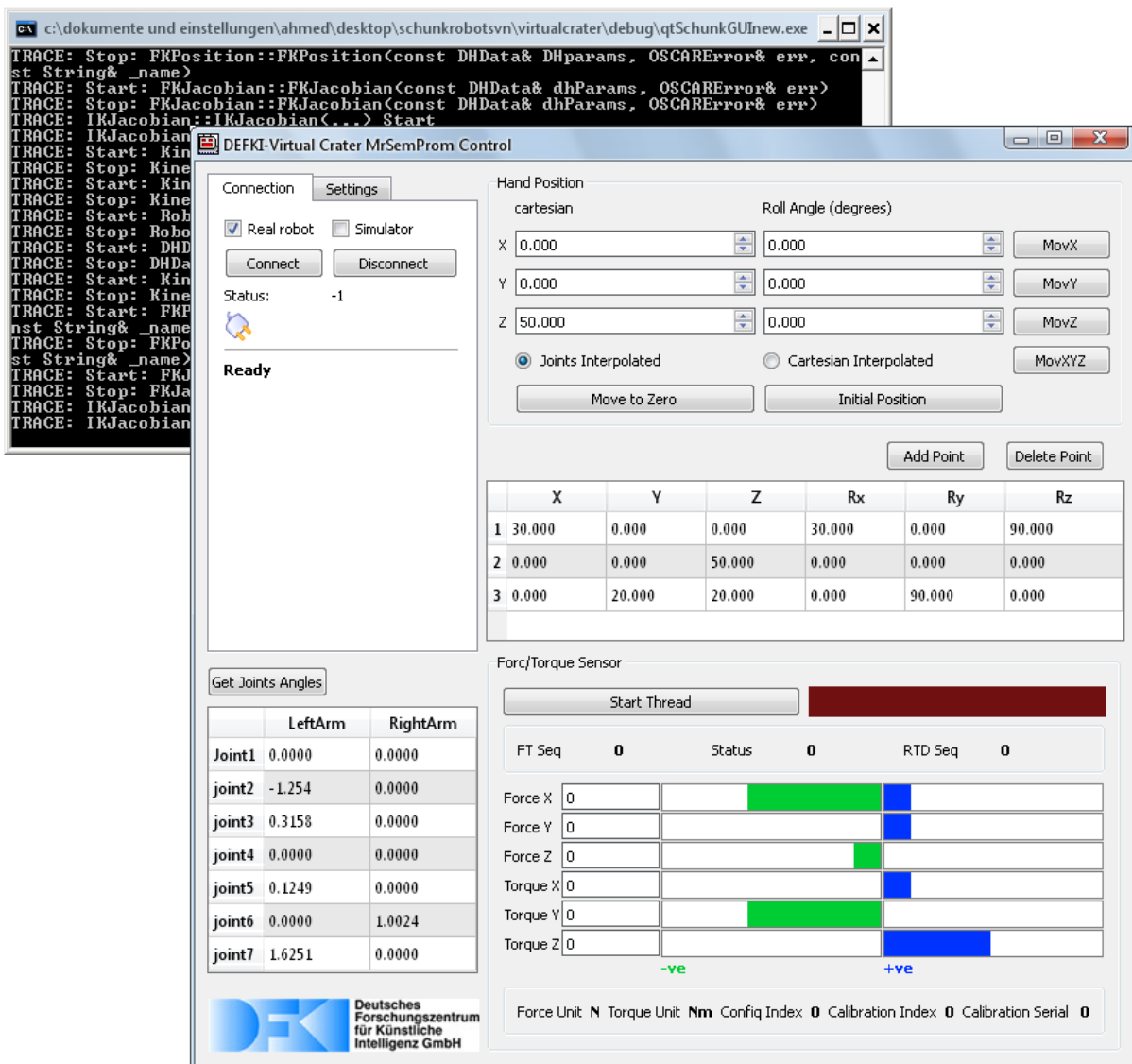


Abbildung 69: vcSemPromControlGUI

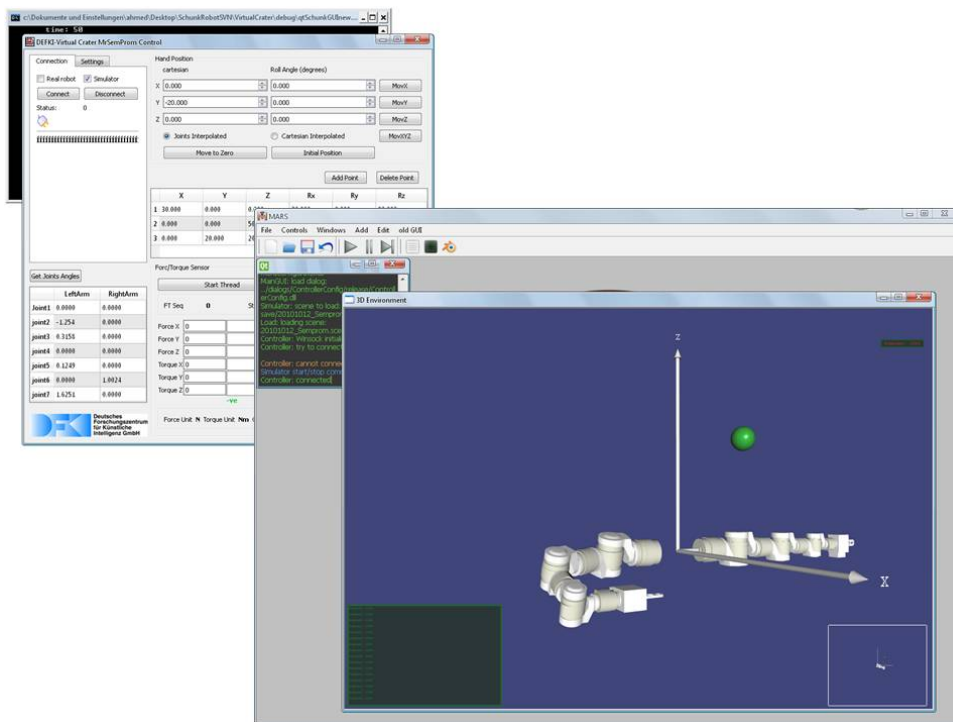


Abbildung 70: Test der entwickelten Software, mit der Mr. SemProm in der MARS-Simulationsumgebung kontrolliert wird

3.1.35 AP4420: Modellierung der Referenzexperimente im virtuellen Testbed

Der Motorteststand (AP3400) wurde vom MMI schematisch in VEROSIM[®] nach modelliert. Das Modell ist in Abb. 71 zu sehen.

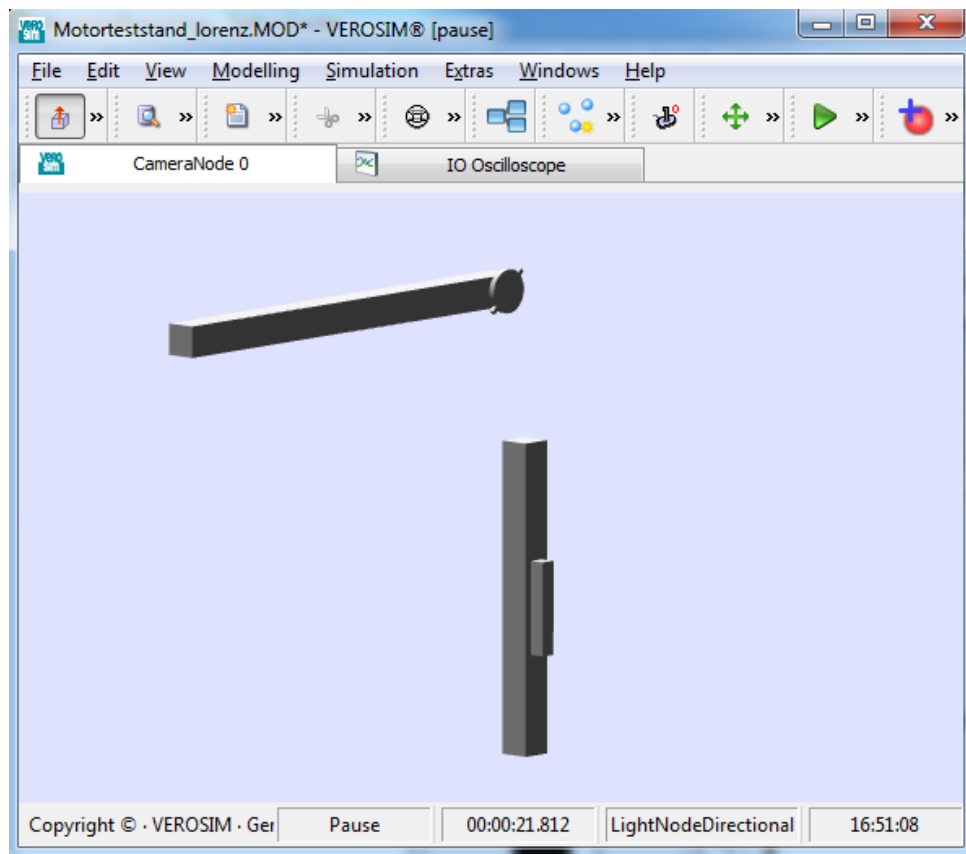


Abbildung 71: Der in AP3400 definierte Motorteststand als schematisches Modell in VEROSIM[®].

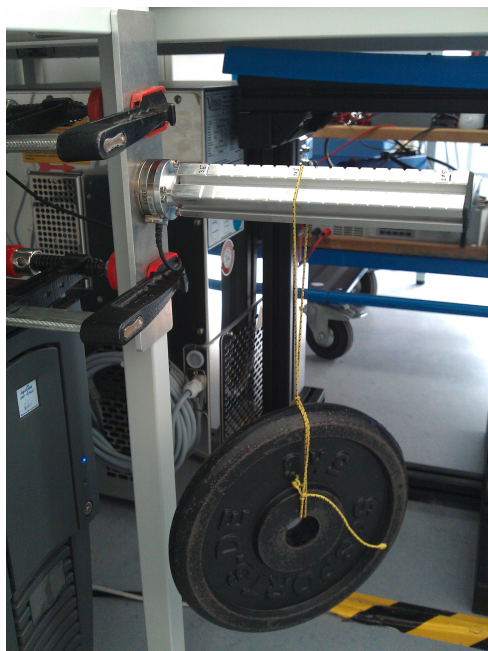
Die Modellierung der virtuellen Bodenmechanik- und Sensorexperimente wurde ebenfalls abgeschlossen.

Nachfolgend werden die einzelnen Referenzexperimente der Sensoren im virtuellen Testbed erläutert.

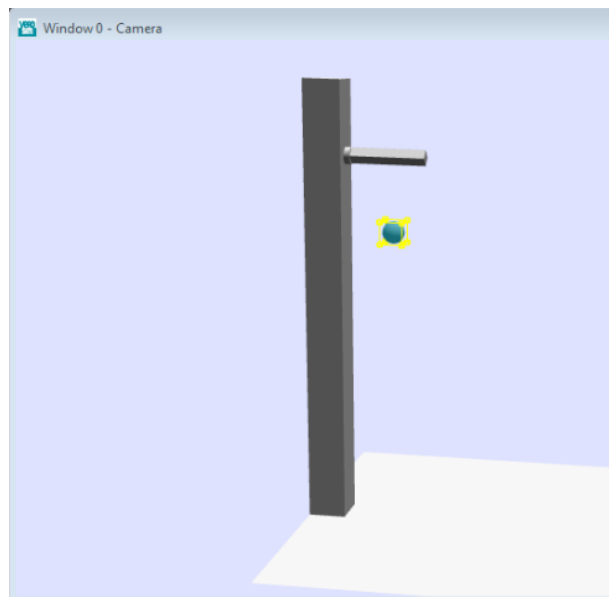
Force-Torque-Sensor

Wie im realen Aufbau ist die der simulierte Kraftmesser an einer starren Struktur befestigt. An dem Sensor ist ein Stange befestigt. Im virtuellen Aufbau hat diese Stange kein Gewicht. Im realen Experiment wurde der Sensor mit der Stange auf Null kalibriert, deswegen ist es kein Problem, dass die Stange im virtuellen Experiment kein Gewicht hat. An der Stange ist eine Kugel befestigt. Diese Kugel stellt das im realen Experiment angehängte Gewicht dar. Dieses Gewicht lässt sich in der Simulation über die Eigenschaften des Objektes ändern. In Abbildung 72 ist der Aufbau in der Simulation und in der Realität gegenüber gestellt. Eine Frage die während der Modellierung

des Experiments aufgetreten ist, ist die nach dem genauen Punkt der Messung. Wird an einer der Seiten der Kraftmessdose oder in der Mitte der Kraftmessdose gemessen. Diese Frage ist für die reine Kraftmessung nicht interessant, da sie unabhängig von dem genauen Messpunkt ist. Für die Torque-Messung ist dieser Punkt jedoch interessant, da sich mit seiner Verschiebung unter Umständen auch der Hebel ändert, mit der eine Kraft wirkt. Diese Information war aus dem Datenblatt nicht zu entnehmen. Dieser Punkt lässt sich in dem Modell leicht ändern, so dass dies bei der Durchführung der virtuelle Referenzexperimente genauer untersucht werden musste.



(a) Realer Aufbau



(b) Virtueller Aufbau

Abbildung 72: Vergleich des Experimentaufbaus für den Force-Torque-Sensor

Joint-Position-Sensor

Für dieses Experiment wurde das vorhandene Modell für ein SpaceClimber-Gelenk mit Motormodell in VEROSIM[®] erstellt. Als Input wird das PWM-Signal genutzt. Ausgewertet wird dann die Position des Gelenkes. In Abbildung 73 ist ein Screenshot des Modells in VEROSIM[®] zu sehen.

2D-Laserscanner

Es wurde ein 3D-Modell des Hokuyo URG-04LX Laserscanners in VEROSIM[®] nachgebaut. In diese Mesh wurde der simulierte Laserscanner integriert. Damit kann der Sensor in einem 3D-Robotermodell einfach eingesetzt werden. Im virtuellen Experiment wird wie im echten Experiment auch nur der zentrale Messpunkt ausgewertet. Der Laserscanner wird also nur als 1D-Laserscanner getestet. In der Abbildung 74 ist

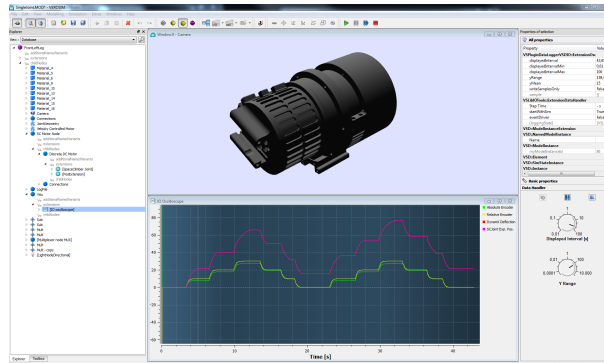


Abbildung 73: Screenshot vom virtuellen Experiment zum SpaceClimber Joint-Position-Sensor

das virtuelle Experiment zu sehen. Es ist gut der rot markierte zentrale Strahl zu sehen, der ausgewertet wird.

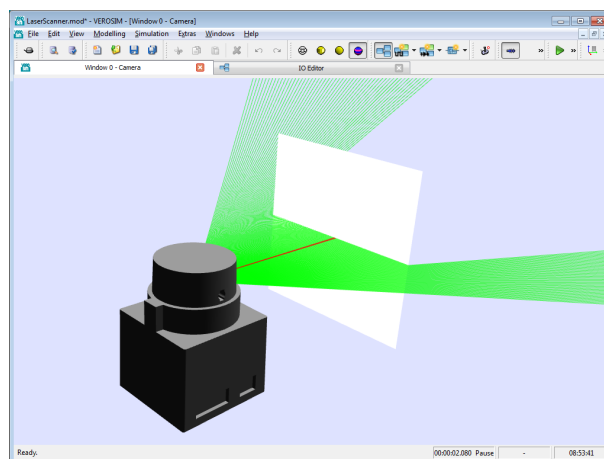


Abbildung 74: Screenshot vom virtuellen Experiment zum Hokuyo URG-04LX Laserscanner

Tiefenbild-Kamera (PMD-Kamera)

In der Abbildung 75 ist das virtuelle Experiment zum Swissranger SR4000 zu sehen. Die rot/schwarze Box links unten im Bild ist die Tiefenbildkamera. Ihr Öffnungswinkel ist durch die graue Pyramide visualisiert. Die weiße Fläche repräsentiert die Wand, auf die im echten Experiment gemessen wurde. In der linken oberen Ecke des Screenshots ist die Ausgabe der PMD-Kamera zu sehen. Die verschiedenen gemessenen Tiefen werden durch unterschiedliche Farben in dem oberen linken Bild kodiert. In dem Beispiel ist die Fläche gleichmäßig, da eine ebene Fläche durch den Sensor erfasst wird. Der schwarze Rand ergibt sich daraus, dass die weiße Fläche, welche die Wand repräsentiert nicht den ganzen Öffnungswinkel abdeckt. Hier kann die Kamera in ihrem Messbereich also kein Objekt erkennen.

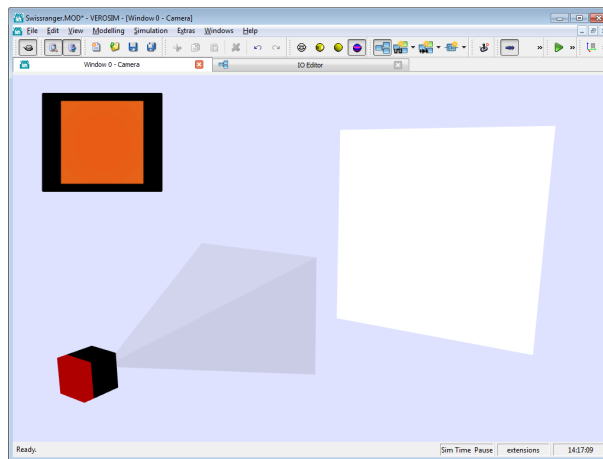


Abbildung 75: Screenshot vom virtuellen Experiment zur Swissranger Tiefenbild-Kamera

3.1.36 AP4500: Aufbau einer Simulations-/Projektionsumgebung im DFKI

Im Rahmen dieses Arbeitspaketes wurde eine stereoskopische 5-fach Projektion am DFKI in Bremen installiert. Im ursprünglichen Zeitplan wurde der Beginn dieses Arbeitspakets für Februar 2010 angesetzt. Das Paket hat eine Laufzeit von 3 Monaten.

Da sich die räumlichen Gegebenheiten am DFKI vor Beginn dieses Arbeitspaketes durch einen Umzug der Simulationsgruppe geändert hatten, konnte das Arbeitspaket nicht im vorgesehenen Zeitrahmen abgearbeitet werden. Die neuen Räumlichkeiten standen erst zum Mai 2010 zur Verfügung, so dass erst ab diesen Zeitpunkt in die konkrete Planung eingestiegen werden konnte.

Die Grafik 76 zeigt die CAVE wie sie nun im Unicom aufgebaut ist. Der Raum für die CAVE wurde extra für diese umgestaltet. So wurde eine Klimatisierung in den Raum eingebaut, es wurden Bodentanks mit Steckdosen für Strom und Netzwerk für die Beamer und Serverschränke eingebaut.

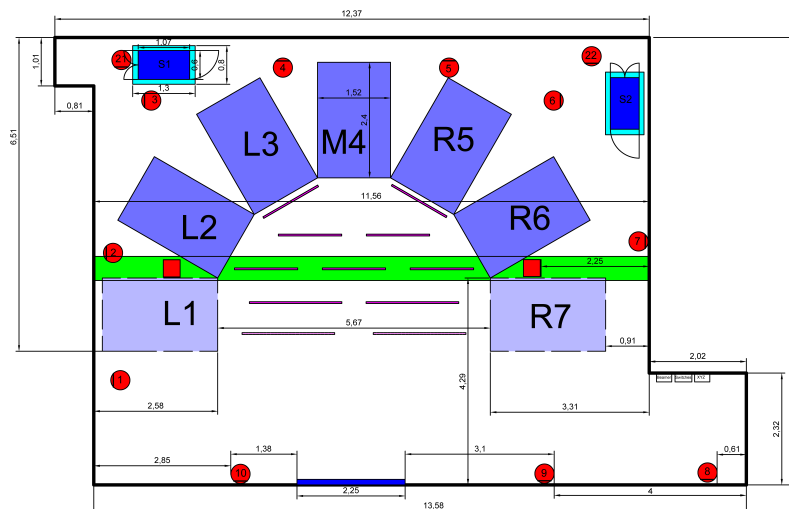


Abbildung 76: CAD-Zeichnung der CAVE im Gebäude UNICOM

Von Mai bis Mitte September 2010 wurde vom RIF in enger Absprache mit dem DFKI die genaue Umsetzung der CAVE abgestimmt. Hierbei traten Probleme auf.

Beamer Zum einen waren die Beamer, welche im Angebot standen, nicht mehr verfügbar. Der Nachfolge Beamer hatte zwei Nachteile. Zum einen hätte dieser wegen seines größeren Gehäuses die Projektionsboxen um ca. 15 cm in der Tiefe vergrößert. Dies wäre in dem Raum sehr eng geworden. Zum anderen gab es Bedenken seitens des RIF's wegen der Haltbarkeit der neuen Beamer. Als Alternative wurde ein vom Gehäuse baugleiches Modell zu dem ursprünglichen Beamer angeboten. Dieser hat vor



allem eine höhere Auflösung (1400 x 1050 anstelle von 1024 X 768) die damit eine Verdoppelung der Pixelzahl darstellt, was sich positiv auf die Qualität der Projektion auswirken sollte. Der Nachteil ist ein höherer Preis. Das Problem des Preises wurde durch eine Anpassung des verwendeten Tracking-Systems erreicht (siehe [3.1.36](#)).

Tracking-System Um mit der CAVE bzw. der darin simulierten oder visualisierten Umgebung interagieren zu können sollte ein 6 DOF Tracking-System der Firma InterSense installiert werden. Dieses System besteht aus zwei Komponenten. Dem eigentlichen Tracker der mit Hilfe von Beschleunigungssensoren und Gyroskopen seine Eigenbewegung misst und einer Ultraschall dusche, welche unter der Decke montiert wird. Die Ultraschall dusche besteht aus einzelnen Emittlern. Der Tracker hat entsprechende Mikrofone, so dass er über die Laufzeit der Signale der Dusche seine absolute Position im Raum bestimmen kann. Diese wird genutzt, um den Drift der Sensoren in dem Tracker zu korrigieren. Damit dies gut funktioniert, müssen die Emitter bestimmte Abstände aufweisen.

Aufgrund der geringen Deckenhöhe in dem Raum von 3,06 Metern und einem Unterzug der direkt durch den zu trackenden Bereich geht mit einer lichten Höhe von 2,66 Meter können die Vorgaben des Herstellers des Tracking-Systems (mindestens 70 cm Abstand zwischen Tracker und Dusche) nur schwer eingehalten werden, sofern der Kopf einer Person getrackt werden soll. Da auch die Auswirkung des Unterzuges nicht eingeschätzt werden konnte, wurde zu Testzwecken ein System ausgeliehen.

Bevor das System testweise aufgebaut wurde, wurde die Umgebung als 3D-Modell in Blender nachgebaut. Hier konnten verschiedene Konstellationen und die Abdeckungen untersucht werden. Als beste Konstellation wurde dabei die in der Abbildung [77](#) bewertet. Die Abbildung [78](#) zeigt die zu erwartende Abdeckung durch das System. Hierbei sind die beiden Elemente direkt an der Projektionsfläche um 45 degree geneigt. Da mit dem geliehenen System die in der Abbild [77](#) gezeigte Konfiguration nicht umzusetzen war, wurden bei einem Test die zwei kritischen Fragen geklärt:

- Wirkt sich der Unterzug störend aus, wenn die Dusche direkt unter der Decke befestigt wird?
- Ist der Abstand zur Ultraschall dusche im Bereich unter dem Unterzug noch groß genug?

Der Test verlief sehr gut, so dass der geplante Aufbau umgesetzt werden konnte. Ein wichtiger Punkt dabei ist, dass der gesamte Raum vor der Projektionsfläche mit nicht mehr als 36 der Ultraschallemitter abgedeckt werden kann. Wären mehr dieser Emitter nötig gewesen, wäre ein anderes Modell des Tracking-Systems nötig gewesen. Dieses hätte jedoch wesentlich mehr Kosten verursacht.

Verdunkelung Damit die Projektion nicht durch von außen einfallendes Licht und damit das Erlebnis des Betrachter gestört wird, müssen die Fenster licht-dicht verschlossen sein. Zudem mussten auch die Bereiche in denen sich die Beamer befinden von den

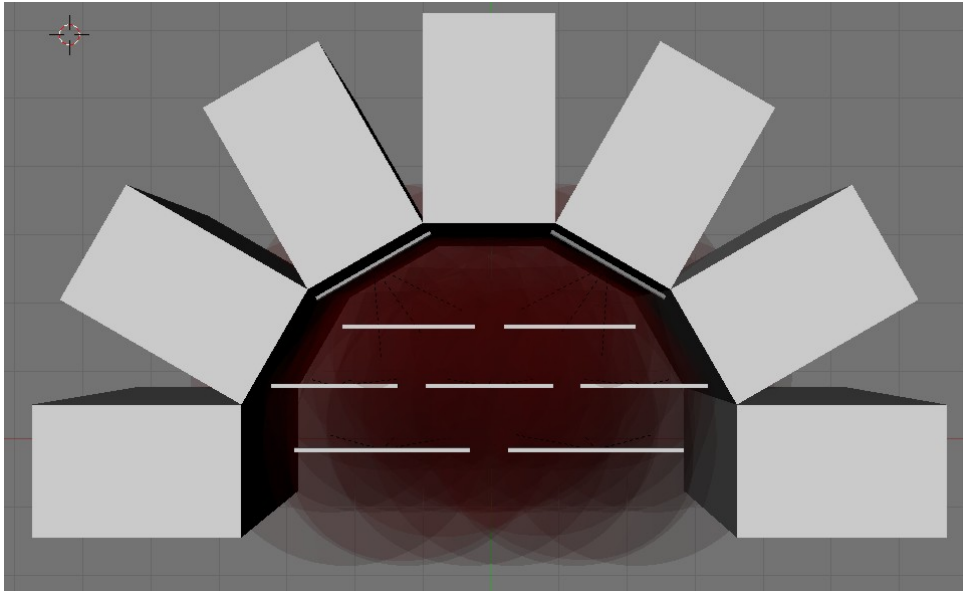


Abbildung 77: Gerenderte Ansicht der Projektionsumgebung: Das geplante Tracking-System

Bereichen in denen sich die Betrachter befinden durch licht-dichte Vorhänge getrennt werden, damit es auch hier nicht zu einer Störung des visuellen immersiven Eindrucks kommt. Zusätzlich mussten noch Vorhänge installiert werden, welche es ermöglichen den Bereich rechts und links neben der CAVE von dem Besucherraum zu trennen. Dies soll den Eindruck der Projektionsflächen noch verstärken und ermöglicht es in den Bereichen hinter dem Vorhang Stauraum zu schaffen.

Ausstattung der CAVE Die CAVE wurde mit 2 48HE Serverschränken ausgestattet. In diesen Schränken befinden sich insgesamt 16 PCs. 10 PCs davon berechnen die 3D-Szene für je eine Leinwand bzw. Auge. Zwei PCs sind als redundante Render-PCs ausgelegt. Zwei Master-PCs sind vorhanden. Diese berechnen die Physik und steuern die Render-PCs. Ein PC ist als Server ausgelegt und hält die Modelldaten für alle PCs bereit und bildet die Verbindung zwischen dem hauseigenen Netzwerk und dem CAVE internen Netzwerk. Zusätzlich ist noch ein Backup-Server vorhanden auf den Sicherungen der PCs und der Daten auf dem Server abgelegt werden können.

Des Weiteren sind beide Server mit einem Gigabit-Switch, einer IP-Steckdose, einem KVM-Switch und einem IP-Reporter (KVM over IP) ausgestattet.

Die Render PCs sind jeweils mit einem Core i7 Hauptprozessor und einer nVidia GTX 480 Grafikkarte ausgestattet. Da es Probleme mit dem Einschaltstrom der PCs gab, kam es dazu dass der Sicherungsautomat beim Einschalten ausgelöst hat, daher wurden die verbauten Netzteile in den PCs gegen eine „höhere“ Serie ausgetauscht, welche einen Anlaufstrombegrenzer integriert haben. Zusätzlich wurden die Sicherungsautomaten gegen welche mit einer trägeren Auslösecharakteristik ausgetauscht.

Als Sound-System wurde ein 5.1 Surround-System der Marke Teufel installiert.

Die CAVE ist seit Mitte Oktober 2010 im Unicom aufgebaut und auch betriebsbereit.

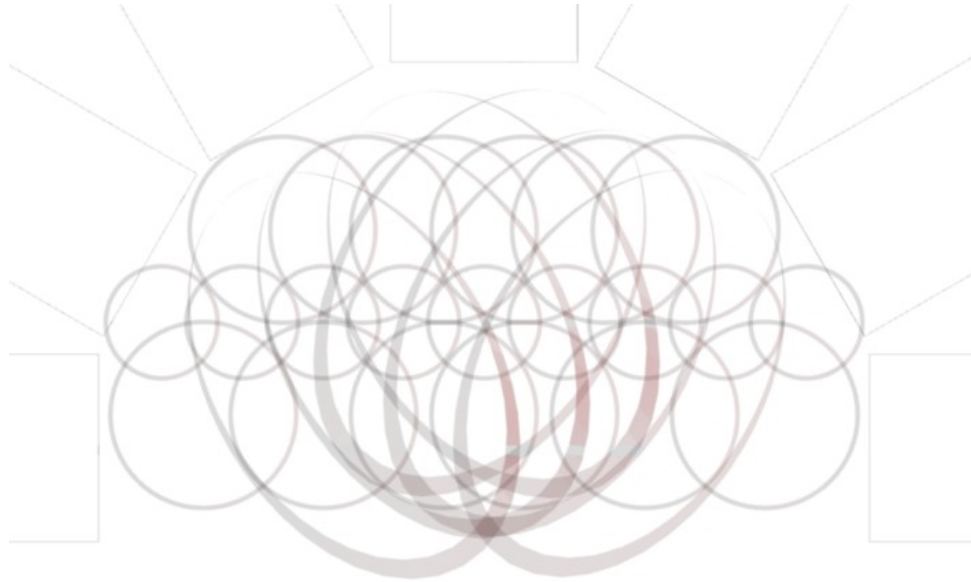


Abbildung 78: Die Abdeckung des Tracking-Systems in ca. 2 Meter Höhe

Es wurde in vierstem Quartal 2011 die Verkleidung oberhalb der Projektionsflächen fertiggestellt. Ebenso wurden alle vorgesehenen Vorhänge angebracht, so dass nun der technische Bereich von dem Besucherbereich abgetrennt ist. Hierdurch wird ein dunkler Besucherraum und eine teilweise Abschottung der Geräusche der Projektoren und Rechner erreicht.

Außerdem wurden die Spiegel gegen Oberflächenspiegel getauscht. Die Halterung der Spiegel wurde überarbeitet, so dass diese nun nicht mehr durch hängen und somit keine Verzerrung der Projektion verursachen.



3.1.37 AP4600: Einarbeitung in das und Schulung auf dem Simulationssystem

Es wurde eine Schulung für das Simulationssystem VEROSIM[®] durchgeführt. Zudem wurde parallel zum im Oktober 2010 begonnenen Aufbau der CAVE in Bremen den Mitarbeitern des DFKI der Umgang der CAVE näher gebracht und Einzelheiten erklärt. Zur Verbesserung der Kommunikation zwischen den Partnern wurde auf Basis eines Microsoft-Sharepoint-Servers ein Internet-Forum eingerichtet, über das Fragen und Probleme diskutiert werden können. Das Forum wurde intensiv genutzt und hat die Effizienz der Kommunikation deutlich gesteigert. Über den Sharepoint Server bestehen neben der reinen Forums-Funktion weitere Möglichkeiten der Zusammenarbeit, wie z.B. das gemeinsame Bearbeiten von Dokumenten. Der Virtual Crater - Sharepoint Server ist unter der Adresse <http://mmi-share.mmi.rwth-aachen.de/websites/VirtualCrater> zu erreichen.

Dieses Arbeitspaket wurde parallel zum Aufbau und während der Einrichtung der Projektionsumgebung bzw. während der Nacharbeiten durchgeführt. Mitarbeiter des DFKI wurden sehr eng bei dem Aufbau und der Einrichtung der Umgebung eingebunden, so dass ein weiterer Schulungstermin nicht mehr nötig war.

Zur Einarbeitung wurden mehrere Modelle in die Simulationsumgebung gebracht und bearbeitet. Hiermit wurde dem Personal am DFKI ein Einblick in die Arbeit mit und an der Projektionsumgebung gegeben.

3.1.38 AP5000: Transfer, Systemtests, Verifikation

3.1.39 AP5100: Systemtest und Anpassungen der Benutzerschnittstellen des Simulationssystems

Das Trackingsystem und der Datenhandschuh wurden entsprechend den Anforderungen angebunden. Durch die Softwareentwicklungen am DFKI wurden bis Ende des Projektes jedoch weiterhin kleine Arbeiten an VEROSIM[®] durchgeführt, wie Fehlerkorrekturen und Anpassungen. Da die ursprünglichen Anforderungen an das System im Bereich Benutzerschnittstellen jedoch bereits erfüllt waren, konnte dieses AP in Q1-2011 abgeschlossen werden.

3.1.40 AP5200: Systemtest und Anpassungen der Schnittstellen zwischen Steuerungshardware und Simulationssystem

Dieses Arbeitspaket beschäftigte sich mit den Schnittstellen zwischen der Roboter-Hardware und der Simulation-Software bzw. der Software, die den Roboter steuert und dem Simulations-System. Hierzu wurden mehrere Experimente durchgeführt. So wurde zuerst der SCARABAEUS-Roboter in der CAVE von dem Roboter-Betriebssystem MONSTER und der dazu gehörigen GUI HOBBIT durch eine erste einfache Mondlandschaft gesteuert.



Der Partner RIF/MMI hat die DFKI-Roboter-Steuerungskomponenten HOBBIT und MONSTER erfolgreich an VEROSIM[®] angebunden, auch die entsprechenden Komponenten des SpaceClimber Roboters. Es existiert auch ein Daten-Rückkanal von VEROSIM[®] an HOBBIT. Durch die am DFKI neu implementierte Möglichkeit, HOBBIT über TCP/IP fernzusteuern, konnte die Anbindung von HOBBIT an VEROSIM[®] nun so erweitert werden, dass Objekte in der virtuellen Umgebung als Pfad-Zielpunkte für den gesteuerten Roboter dienen können.

HOBBIT - VEROSIM[®] Verbindung: Wie oben beschrieben wurde in HOBBIT ein TCP-Server implementiert. Dieser nimmt Kommandos von anderen Programmen entgegen. Über diese Kommandos können im wesentlichen die gleichen Werte geschickt werden, wie sie sonst über die grafische Benutzeroberfläche einzugeben sind.



3.1.41 AP5300: Anpassung der Simulationsmodelle an reale Ergebnisse

Motorteststand Bisherige Analysen haben gezeigt, dass der externe Stromverbrauch (der Gleichstrom der von der Spannungsquelle bezogen wird) des SpaceClimber Gelenkaktuators sowohl von der Geschwindigkeit als auch vom extern wirkenden Drehmoment abhängt. Diese Abhängigkeit wurde weiter untersucht, um ein akkurates Simulationsmodell des Stromverbrauchs zu entwickeln. Das Modell wurde als „Black Box“ mit Hilfe der experimentellen Systemidentifikation entwickelt. Um genug Daten für die Systemidentifikation zu sammeln wurde eine Reihe von Experimenten durchgeführt. Eine kurze Beschreibung dieser Experimente ist in Tabelle 10 zusammengefasst. Die Experimente wurden an der hausinternen „Motortestbench“ durchgeführt. Der Versuchsaufbau ist in Abbildung 79 zu sehen.

Exp.	V_{DC}	T_L	PWM
1.	24V	variable	variable
2.	24V	variable	variable
3.	24V	variable	variable
4.	24V	0	variable
5.	24V	variable	variable
6.	24V	steps	constant
7.	48V	variable	variable
8.	48V	variable	variable
9.	30V	0	variable

Tabelle 10: Motortestbench Experiment Beschreibung. SpaceClimber Gelenk Versorgungsspannung (V_{DC}), anliegendes externes Drehmoment (T_L) und PWM Eingangssignal.

Um das Gelenk und die Hysteresebremse mit den gewünschten Signalen zu kontrollieren wurde die bestehende Motortestbench LabVIEW-Software erweitert (siehe Abbildung 80). Diese Erweiterung ermöglicht der Software sich direkt über TCP/IP mit dem Gelenk zu verbinden, das gewünschte PWM-Signal zu setzen und die gelenkinternen Sensoren auszulesen.

Die Software speichert alle gemessenen Daten mit einem Samplingintervall von $T_s = 10$ ms, während die „SpaceClimber joint GUI“-Software gleichzeitig alle gelenkinternen Daten zur späteren Konsistenzprüfung speichert. Abbildung 81 zeigt einige Daten während der Durchführung eines Experiments.

Zur Erstellung des Modells zum Stromverbrauch wird ein iterativer „Extended Recursive Least Squares“ Systemidentifikationsalgorithmus eingesetzt. Der Algorithmus erhält als Eingaben das PWM-Signal, das anliegende Drehmoment und den daraus resultierenden Stromverbrauch aus den realen Experimenten und erzeugt eine Übertragungsfunktion gewünschter Ordnung. Als vorläufiges Ergebnis erzeugte der Algorithmus folgende Übertragungsfunktion als „Black Box“-Modell:

$$i = \frac{-0.3525s^2 + 22.46s + 601.5}{s^3 + 142.5s^2 + 5.319e004s + 1.868e006} pwm + \frac{29.5s^2 - 4885s + 6.19e005}{s^3 + 142.5s^2 + 5.319e004s + 1.868e006} T_L \quad (17)$$

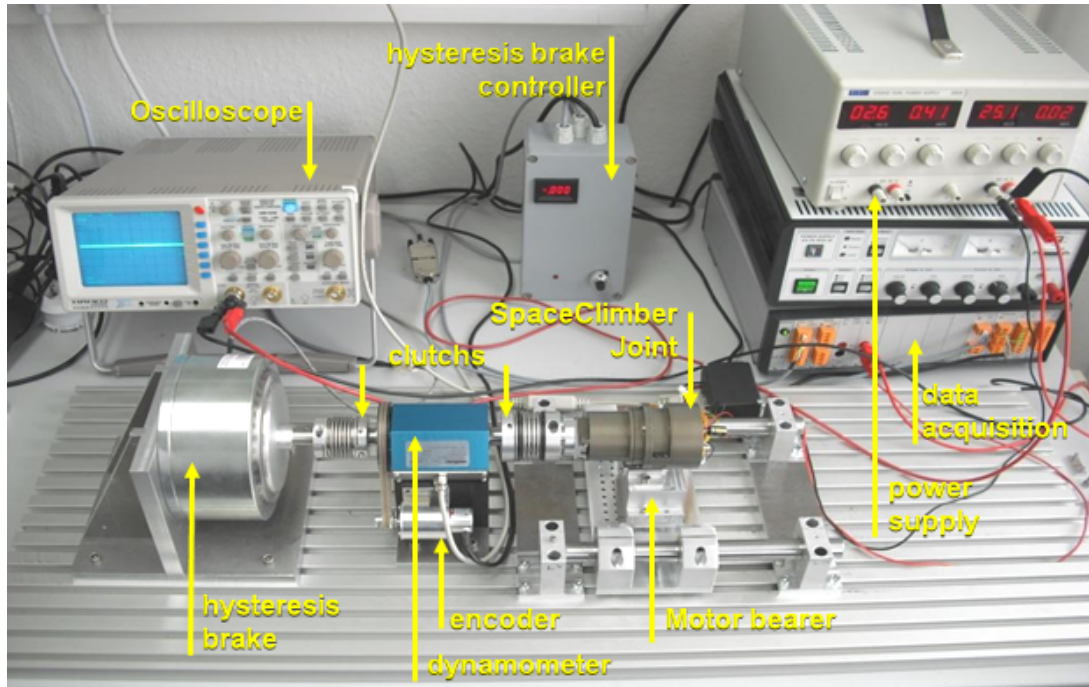


Abbildung 79: Aufbau des „Motortestbench“ Experiments. Das SpaceClimber Gelenk ist am Datenerfassungsgerät (data acquisition) angeschlossen.

Dieses Modell wird benutzt um den Stromverbrauch eines anderen Experiments aus dem PWM-Signal und dem Drehmoment vorherzusagen. Abbildung 82 zeigt den vom Modell vorhergesagten Stromverbrauch im Vergleich zu dem im Experiment Gemessenen.

BLDC Motormodell Zunächst wurde ein abstraktes Motormodell des SpaceClimbers entwickelt, welches manche physikalischen Eigenschaften des Motors wie zum Beispiel Position, Geschwindigkeit und Stromverbrauch in Echtzeit simuliert. Dieses Modell lässt sich für gewöhnlich durch Systemidentifikation aus gewissen Experimenten mit dem realen Motor ableiten. Alternativ lässt sich das abstrakte Modell auch aus einem detaillierten Brushless DC (BLDC) Motormodell ableiten. Dies ist vor allem von Vorteil, wenn die Durchführung einer experimentellen Systemidentifikation nicht möglich ist, weil zum Beispiel ein reale Motor noch nicht existiert. Um dies zu ermöglichen, wurde ein detailliertes BLDC Motormodell in Matlab/simulink entwickelt (Abb. 83). Das detaillierte BLDC Modell besteht aus einem BLDC Kern, einem Invertierer und einem PWM Generator.

Der BLDC Kern besteht aus einem State-Space Modell, das auf einem dreiphasen Motormodell beruht, und eines Berechnungsteils, der das trapezförmig Signal, die Position und das Drehmoment berechnet (Abb. 84 und Abb. 85).

Der Invertierer generiert die Eingangsspannung für die Zustandsgleichung des State-Space Modells anhand der Position und des Chopping Signals. Das Chopping sorgt dafür, dass der Strom in der energetisierten Phase entsprechend des eingehenden

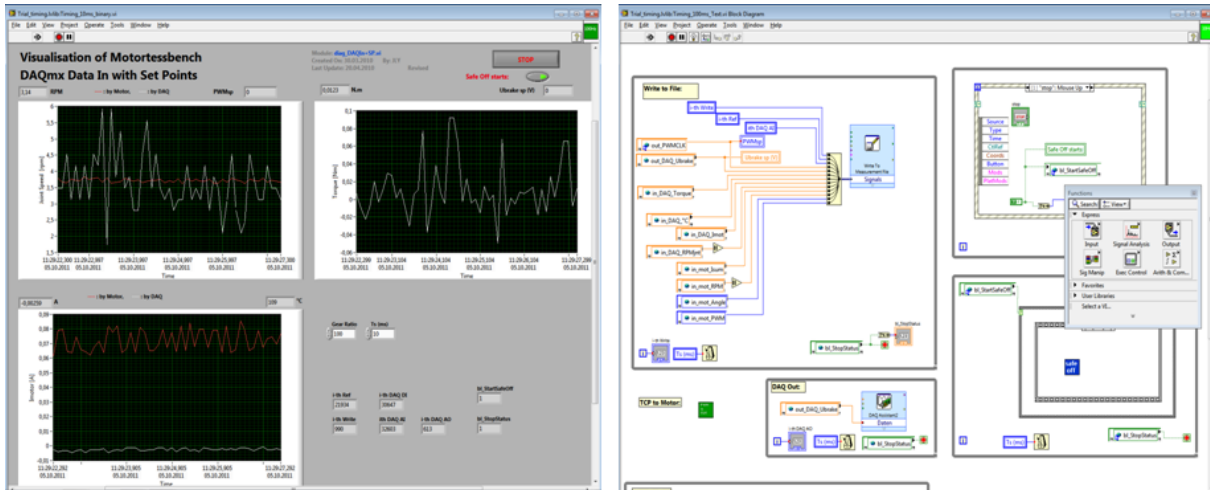


Abbildung 80: Motortestbench LabVIEW-Software Benutzerschnittstelle und internes Blockdiagramm.

PWM-Signals ein- und ausgeschaltet wird.

Abbildung 86 und 87 zeigen den Dreiphasenstrom, die elektromotorische Gegenkraft, die Geschwindigkeit und das Drehmoment der Simulation. Dabei ist R der Abschlusswiderstand (Phase-zu-Phase), L die Anschlussinduktivität (ebenfalls Phase-zu-Phase), J das Trägheitsmoment des Rotors, K_t die Drehmomentkonstante und K_f die Reibungskonstante des Motors.

Motormodelle In Tests zur Echtzeitfähigkeit des SpaceClimber Modells haben sich die 24 Motormodelle in den Gelenken als signifikant verlangsamenderer Faktor für die Simulation herausgestellt. Insbesondere die Implementierung der Positions- und Geschwindigkeitskontroller in VEROSIM[®], die in der Entwicklungsphase auf Grund ihrer hohen Flexibilität gewählt wurde, hat sich als zu langsam herausgestellt. Aus Performancegründen wurden die besagten Kontroller nun in C++-Code in das Motorplugin integriert. Dadurch verzichtet man auf die Möglichkeit das Kontrollerdesign flexibel in der Simulation zu ändern und gewinnt dadurch erheblich an Ausführungsgeschwindigkeit. Eine Anpassung der Kontrollerparameter ist nach wie vor auch zur Laufzeit noch möglich.

Um auch ohne nicht echtzeitfähige genaue Motormodelle auf einen Stromverbrauch der Motoren schließen zu können, wurde eine echtzeitfähige Funktion ermittelt, die ein Mapping von Drehmoment und Motorgeschwindigkeit auf den Stromverbrauch darstellt. Die dabei verwendete Funktion ist $f(x, y) = axy + bx + ay + d$. Die Parameter a , b , c und d wurden mittels Parameteroptimierung bestimmt. Das Ergebnis ist in Abbildung 88 dargestellt.

Neben der vordefinierten Funktion wurde ebenfalls die Implementierung eines neuronalen Netzes verwendet, um eine Stromabbildung zu generieren. Die Ergebnisse dazu sind in Abbildung 89 dargestellt.

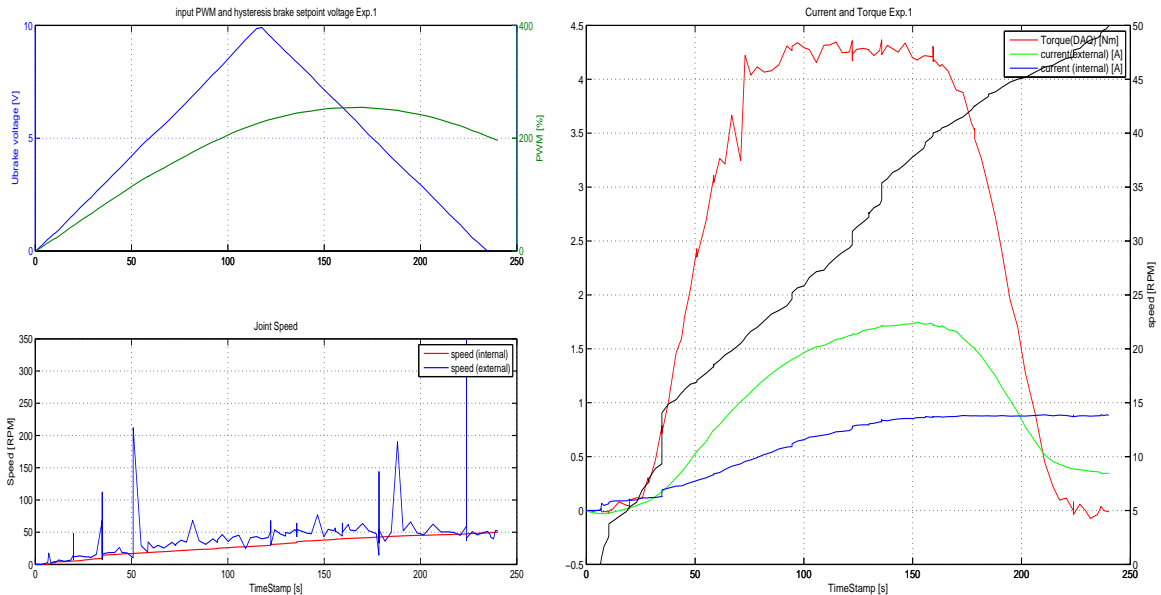


Abbildung 81: Einige Daten während der Durchführung eines Experiments auf der „Motortestbench“ mit dem SpaceClimber Gelenk.

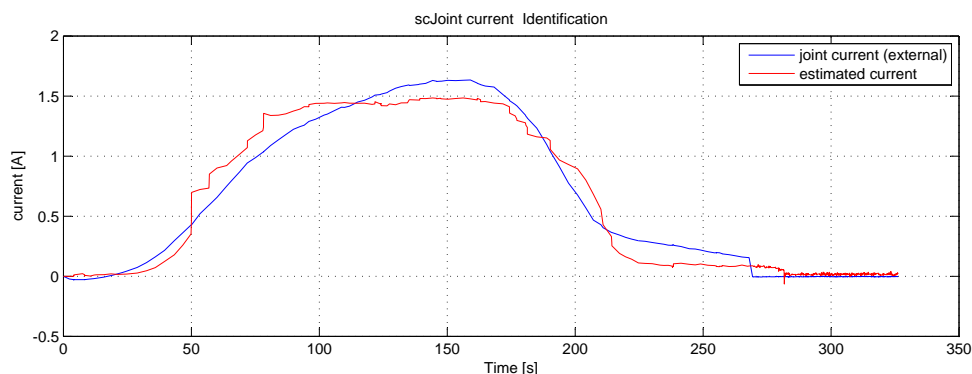


Abbildung 82: Vorläufiges Ergebnis des Stromverbrauchmodells des SpaceClimber Gelenks aus der Systemidentifikation.

Referenzexperimente zur Bodenmechanik Zur Durchführung der virtuellen Referenzexperimente zu den lateralen Bodenkräften (siehe AP5420) wurden die entsprechenden Parameter des PTP Modells mit Hilfe des Optimierungs- und Automatisierungs-Plugin an die Daten des realen Referenzexperiments angepasst. Dazu wurde das Plugin um die Fähigkeit erweitert nicht nur Fließkommazahlen sondern auch andere Datentypen wie zum Beispiel Vektoren und Koordinatentransformationen (VSM::Frame) zu optimieren. Auch das PropertyChanger-Plugin welches diese Parameter auf die entsprechenden VEROSIM®-Knoten schreibt wurde entsprechend erweitert.

Um die lateralen Kräfte der Bodenmechanik weiter zu untersuchen, wurde eine Reihe von Experimenten durchgeführt, bei denen im Unterschied zu den bereits durchgeführten Experimenten versucht wurde die wirkende Normalkraft konstant zu halten. Dazu wurde eine aktive Kraftregelung in der Steuersoftware des MrSemProm-Arms implementiert. Zusätzlich wurde ein Federadapter zur passiven Kraftregelung zwischen die

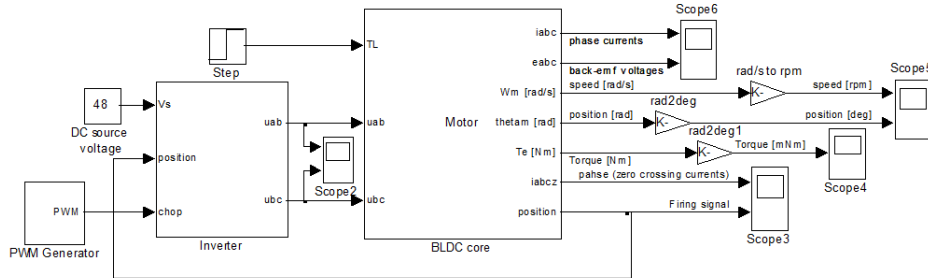


Abbildung 83: Der Aufbau des detaillierten BLDC Motormodells.

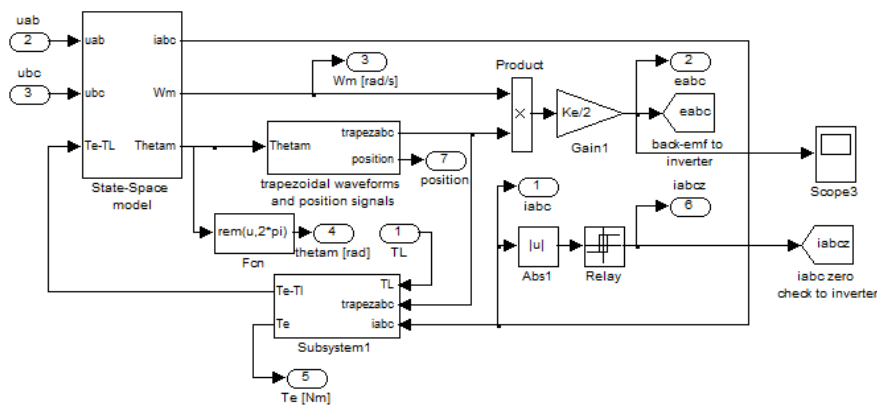


Abbildung 84: BLDC Kern des detaillierten Motormodells.

Kraftmessdose und den SpaceClimber-Fuß angebracht. In diesem Experiment wurde der SpaceClimber-Fuß mit integrierten Sensoren benutzt. Abbildung 90 zeigt links den Arm-Adapter-Fuß Aufbau und rechts Details des Federadapters.

In diesem Experiment wurde, nachdem der Fuß senkrecht über dem Boden positioniert wurde, die Kraftregelung eingeschaltet mit einem Sollwert von 50 N. Dadurch bewegt dich der Arm nach unten und drückt den Fuß in den Boden. Nach etwa 5 Sekunden wird der Arm mit einer Geschwindigkeit von 5 mm/s seitlich um 150 mm bewegt. Dabei versucht die Kraftregelung die Normalkraft aufrecht zu erhalten. In Abbildung 91 ist eine Skizze des Experiments zu sehen.

In allen Experimenten werden die sieben Gelenkwinkel, die Daten des Kraft-/Drehmomentsensors, die kartesischen Koordinaten des Tooltips und die internen Sensoren des SpaceClimber-Fußes (Beschleunigung, IMU, Kraft- und Drucksensoren) mit einem Zeitstempel versehen abgespeichert. Abbildung 92 zeigt die Resultate von fünf durchgeführten Experimenten.

Optimierung Neuronales Netz an Referenzdaten Die Implementierung der genetischen neuronalen Netze wurde um Positionsinformationen der Neuronen erweitert. Zudem werden innerhalb der Optimierung neue Neuronen und Verbindung über Regeln er-

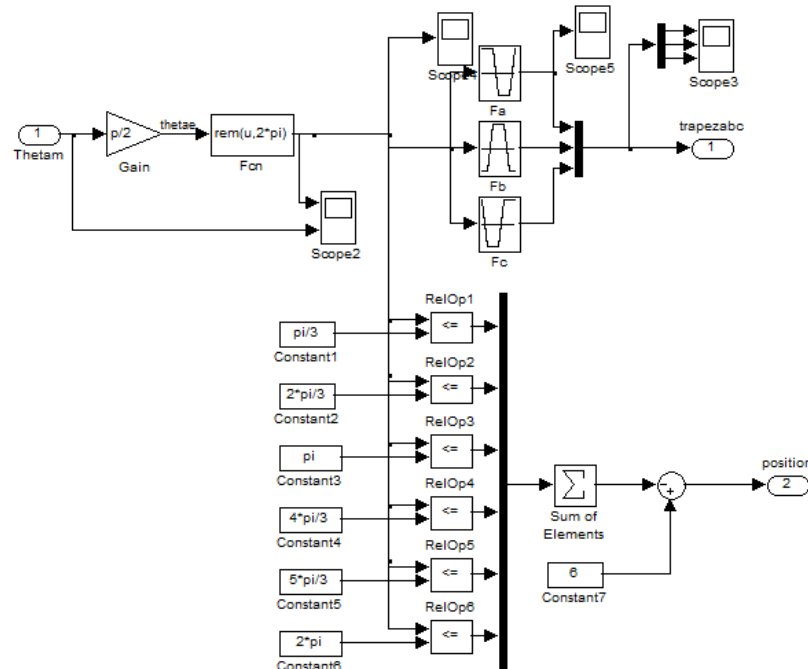


Abbildung 85: Berechnung des trapezförmigen Signals, der Position und des Drehmoments.

stellt, die die Positionen berücksichtigen. So ist die Wahrscheinlichkeit, dass nah beieinander liegende Neuronen bei der Erstellung von Verbindungen verknüpft werden sehr viel höher als bei weit auseinander liegenden Neuronen. Zudem wurde die Möglichkeit implementiert festpositionierte Neuronen vorzugeben die als Ein- bzw. Ausgabeneuronen verwendet werden. Dies hat zur Folge, dass nicht alle Neuronen die Eingaben des Netzes bekommen und Strukturen gebildet werden müssen um die gewünschte Abbildung zu ermöglichen.

Die Neuerungen wurden auf den Testfunktionen, den Normalkraft-Experimenten und den neu durchgeführten Experimenten zur Messung der lateralen Kräfte durchgeführt. [Abbildung 93](#) zeigt die Ergebnisse der neuronalen Netze mit der weiterentwickelten Strukturoptimierung angewendet auf die Testfunktionen. In [Abbildung 94](#) ist das beste Ergebnis von 20 Optimierungsdurchläufen auf den Experimenten zur Bestimmung der Normalkräfte dargestellt. Die Abweichung der Netzwerkausgabe zu den Experimentdaten konnte durch die Erweiterungen im Vergleich zu den Ergebnissen aus dem letzten Quartal halbiert werden.

Aus den lateralen Referenzexperimenten wurden sechs Experimente ausgewählt um ein Netzwerk zu entwickeln, welches eine Abbildung aus der Eindringtiefe und der Normalkraft auf einen Reibungskoeffizienten darstellt. In [Abbildung 96](#) sind alle sechs Experimente mit den Netzwerkeingaben Tiefe und Normalkraft sowie dem Reibungskoeffizienten aus den Experimenten und dem Reibungskoeffizienten aus dem Netz dargestellt. [Abbildung 96](#) stellt alle Messpunkte mit den Netzausgaben gegenüber die zum Entwickeln des Netzes verwendet wurden, während [Abbildung 97](#) das Netzwerk mit allen aufgenommenen Experiment-Messpunkten vergleicht.

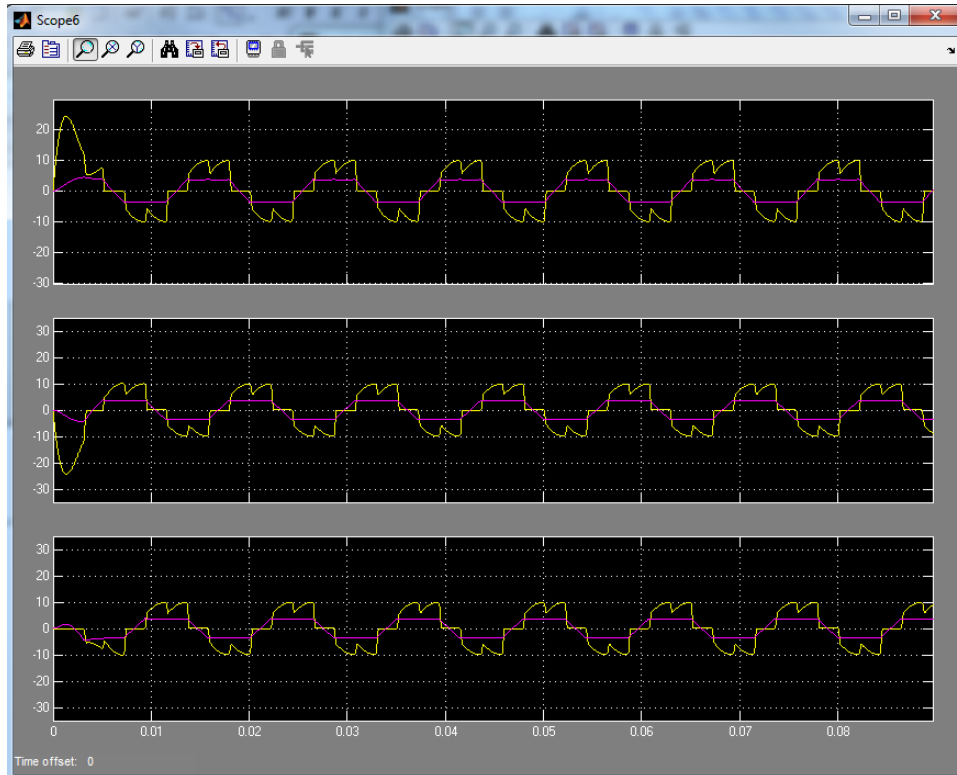


Abbildung 86: Simulation des Dreiphasenstroms und der elektromotorischen Gegenkraft mit folgenden Motorparametern: $R= 0.3433 \Omega$, $L= 605 \mu H$, $J= 4.9e - 6 \text{ kg} \cdot \text{m}^2$, $Kt= 0.075$, $Kf= 1.3e - 3$

Innerhalb dieses Arbeitspaketes wurden weitere Experimente zum Messen der Normalkräfte beim Eindringen in sandigen Boden mit dem Basalt als Substrat durchgeführt (siehe Abbildung 98). Die neuen Experimentdaten zeigen einen ähnlichen Varianzbereich wie der feinkörnigere Sand. Es hat sich in einzelnen Experimenten allerdings gezeigt, dass die Geschwindigkeit beim Eindringen einen deutlich größeren Einfluss auf den Widerstand des Bodens hat. Dieser Effekt konnte mit dem vorhandenen Testaufbau, auf Grund der fehlenden Präzision bei höheren Geschwindigkeiten, nicht ausreichend untersucht und belegt werden. Einen weiteren Unterschied stellt die Kurvenform dar, die deutlich mehr variiert als bei dem Datensatz mit dem feinkörnigerem Boden-substrat. Abbildung 99 verdeutlicht diesen Unterschied. In der Abbildung liegen die vier Kurven in der 1. Markierung nahe beieinander während sie sich innerhalb der 2. Markierung über beinahe den gesamten Varianzbereich verteilen.

Die folgenden Abbildungen zeigen das Ergebnis für ein erlerntes neuronales Netz zur Abbildung der Experimentdaten. Viele der Experimentdaten konnten sehr gut angenähert werden. In einigen Fällen konnte das Netz allerdings nicht zwischen den unterschiedlichen Kurvenformen mit nur einem Varianzparameter unterscheiden. Im Gegensatz zu dem ersten Experiment wurde hier nur die Hälfte der Experimente verwendet um das Netz zu erlernen. Die Standardabweichung auf den zum Lernen verwendeten Experimentdaten beträgt 3.72N und gemessen für alle Experimentdaten liegt sie bei 3,99N. Von den folgenden Abbildungen zeigt die erste Abbildung Kurven aus dem gesamten Varianzbereich und stellt somit die Form der gelernten Funktionen dar. Die

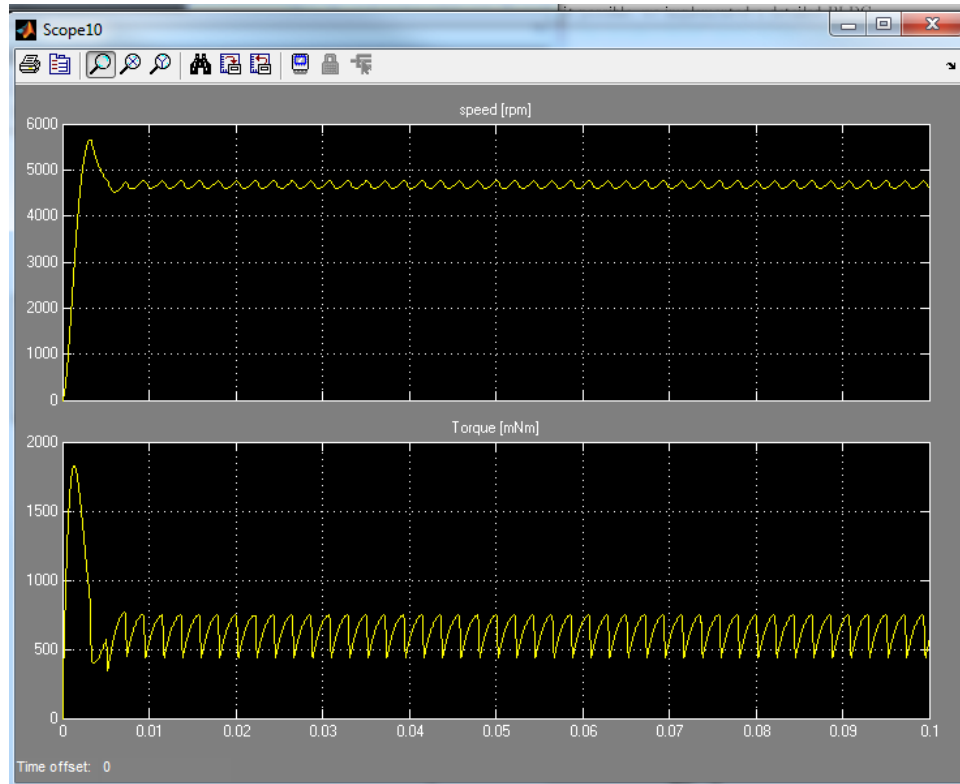


Abbildung 87: Simulation der Motorgeschwindigkeit und des Drehmoments mit den Motorparametern $R=0.3433 \Omega$, $L=605 \mu H$, $J=4.9e-6 \text{ kg} \cdot \text{m}^2$, $K_t=0.075$, $K_f=1.3e-3$

weiteren Abbildungen zeigen die Ausgabe des neuronalen Netzes gegenüber den gemessenen Daten, wobei die gepunkteten Kurven die realen Daten sind und die durchgezogene Linie die dazugehörige Netzausgabe ist.

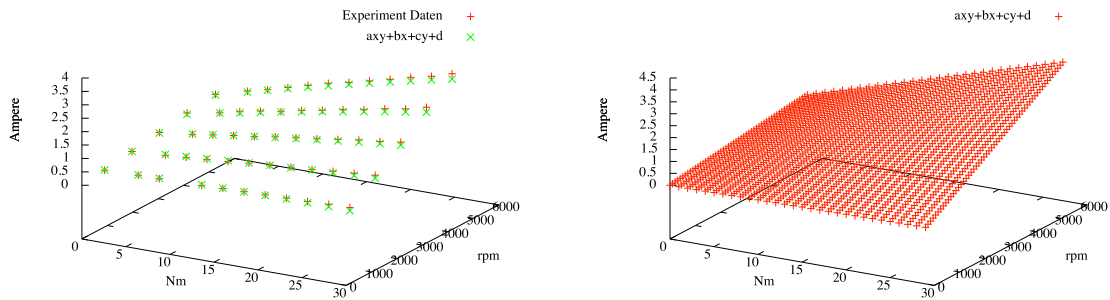


Abbildung 88: Der linke Graph vergleicht die Stromwerte, die in dem Seilwindenexperiment gemessen wurden, mit der Ausgabe der angenäherten Funktion. Der rechte Graph stellt die Funktion gesampelt über den gesamten Wertebereich dar.

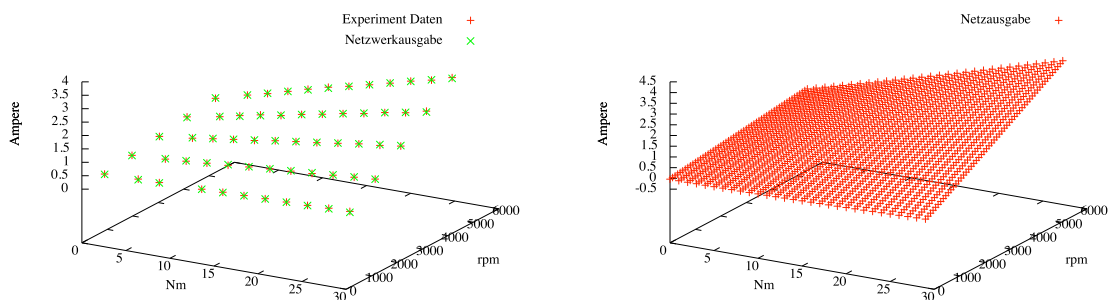


Abbildung 89: Der linke Graph vergleicht die Stromwerte, die in dem Seilwindenexperiment gemessen wurden, mit der Ausgabe eines gelernten neuronalen Netzes. Der rechte Graph stellt die Ausgabe des Netzes gesampelt über den gesamten Wertebereich dar.

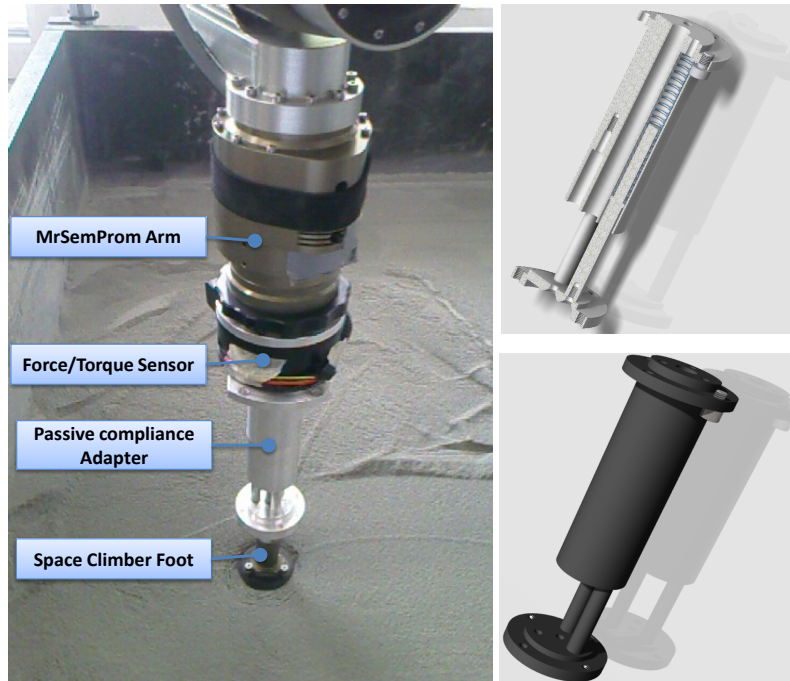


Abbildung 90: Zwischen MrSemProm-Arm und SpaceClimber-Fuß wurde ein passiver Federadapter eingebaut (links). Der Federadapter besteht aus drei Zylindern, die jeweils gefedert sind (rechts).

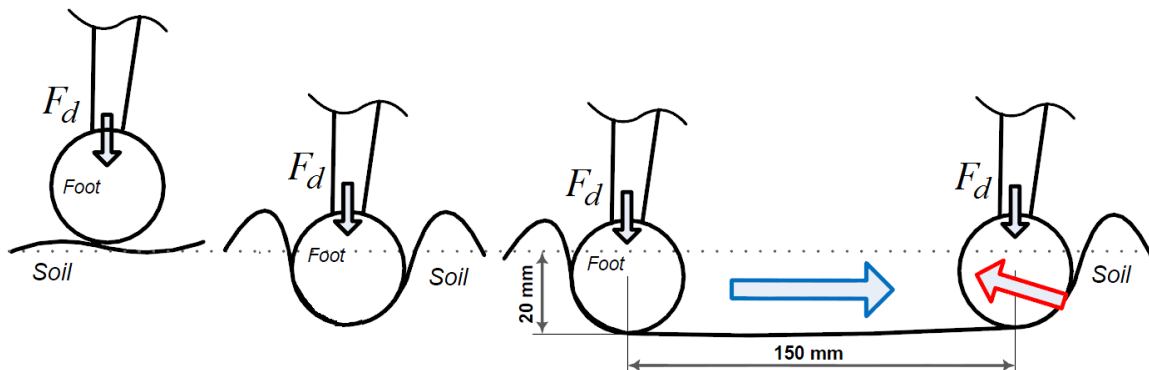


Abbildung 91: Skizze des Experiments zu den lateralen Kräften.

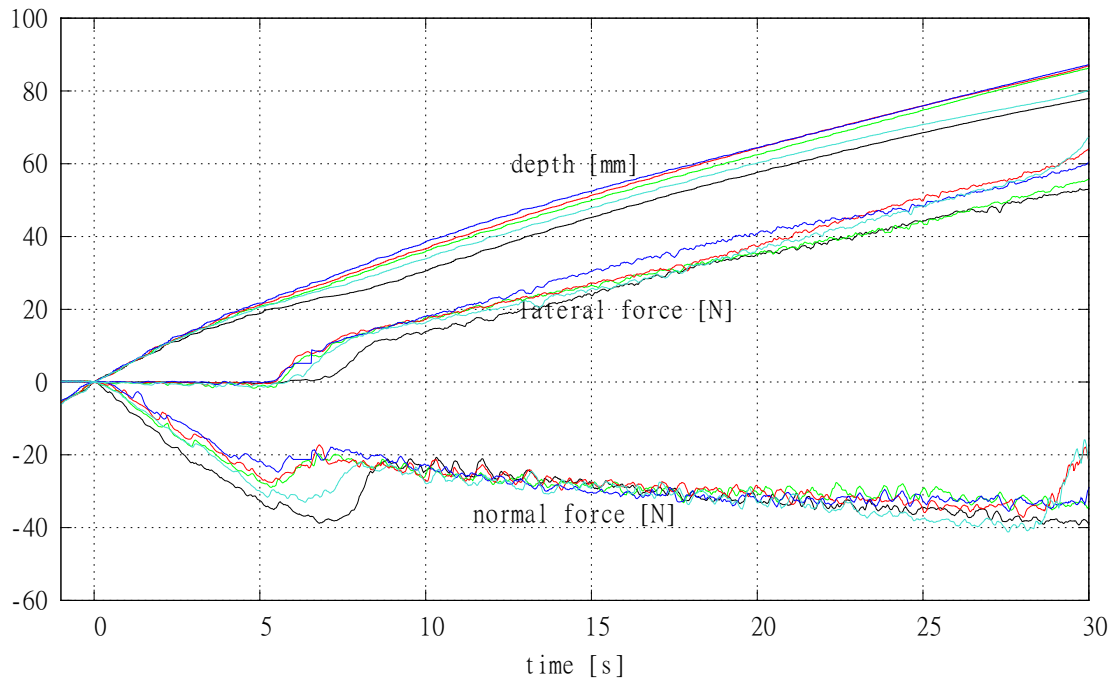


Abbildung 92: Resultate der lateralen Bodenexperimente mit Kraftregelung.

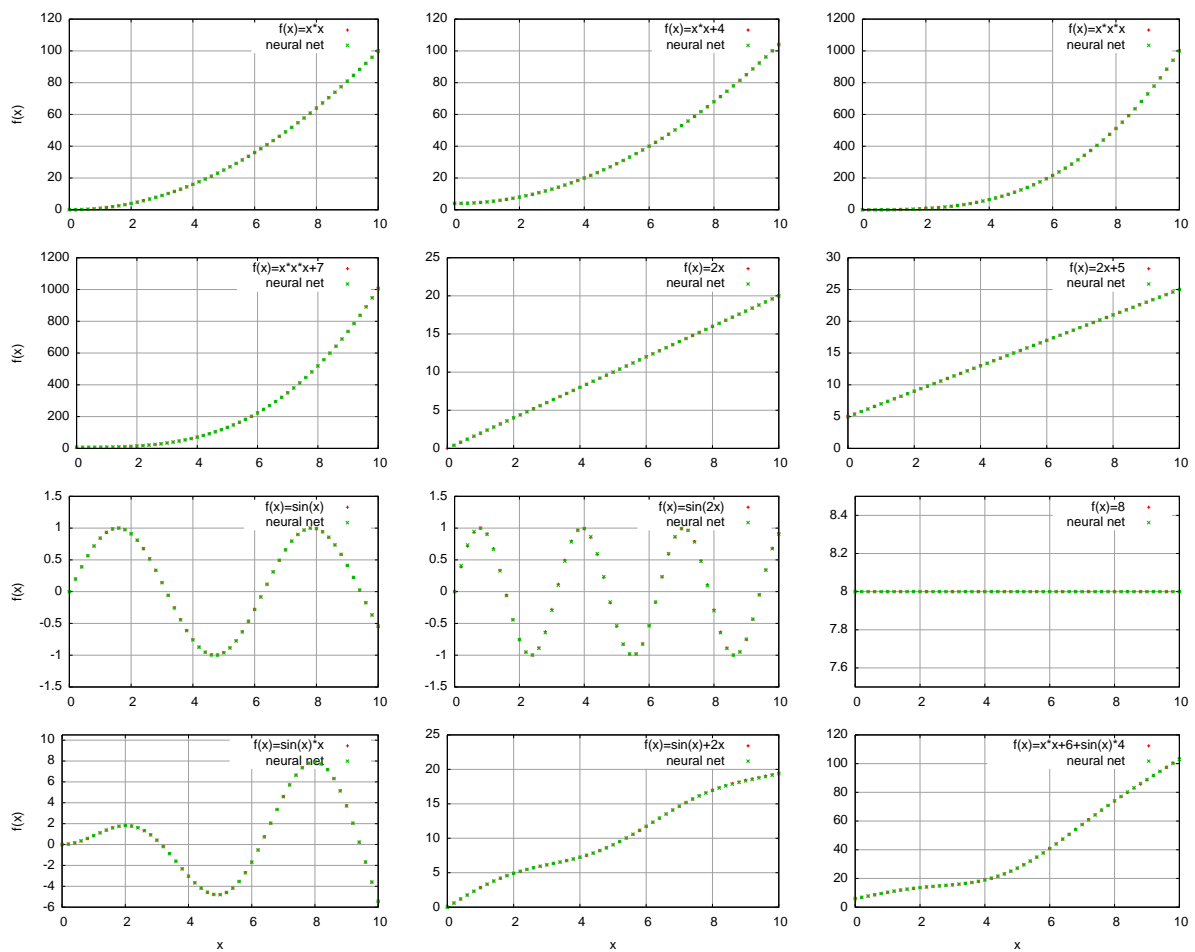


Abbildung 93: Vergleich der optimierten neuronalen Netze mit den Testfunktionen.

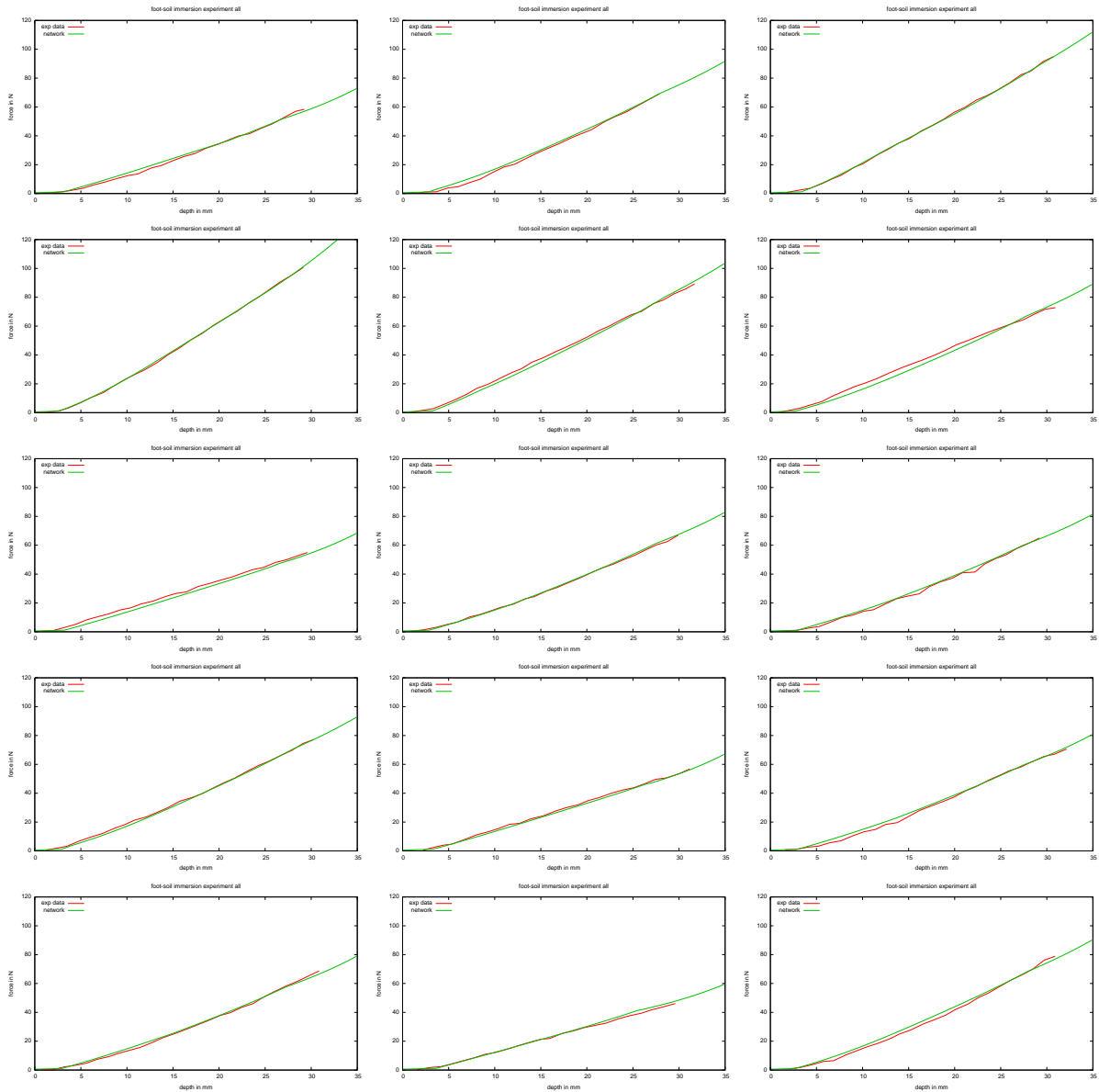


Abbildung 94: Vergleich der Netzwerkausgabe mit den realen Experimentdaten.

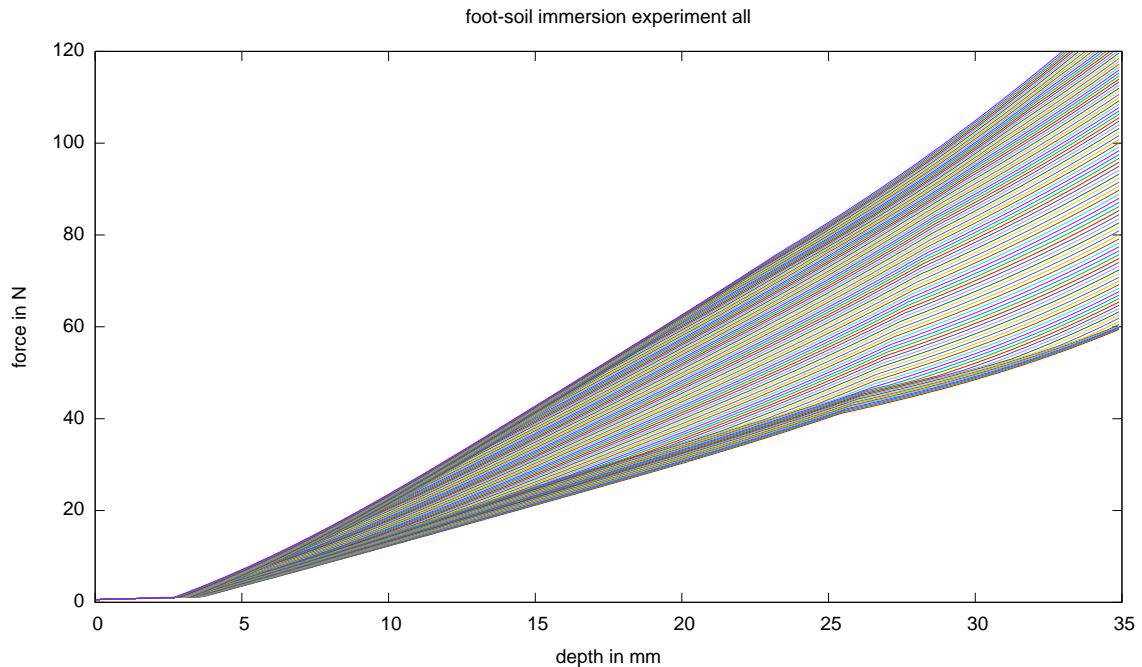


Abbildung 95: Generalisierung des Netzes für den gesamten Varianzbereich.

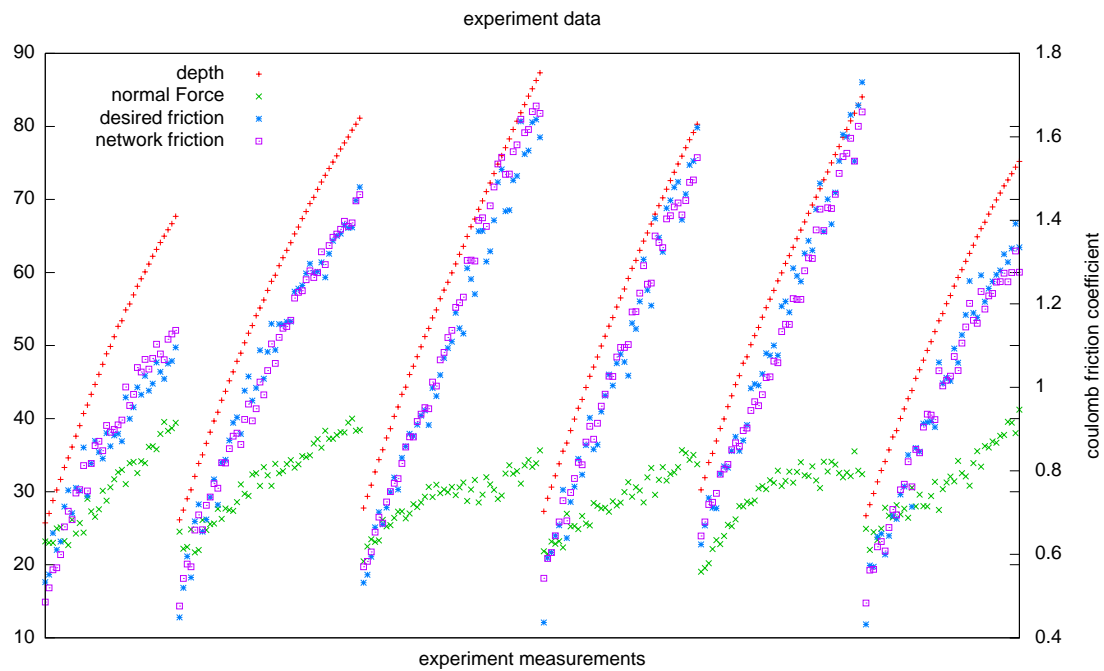


Abbildung 96: Der Graph vergleicht die für das Entwickeln des Netzes verwendeten Experimentdaten mit der Ausgabe eines ausgelernten Netzes. Die sechs Abschnitte stellen jeweils die Messdaten eines Experimentes dar. Da dem Netzwerk keine Unterscheidungsmöglichkeit, ähnlich des Varianzparameters, zur Verfügung stand, stellt das erlernte Netzwerk ein Mittel aller Experimentdaten dar. Die durchschnittliche prozentuale Abweichung der Netzwerkausgabe zu den Experimentdaten pro Messung beträgt 3.67%.

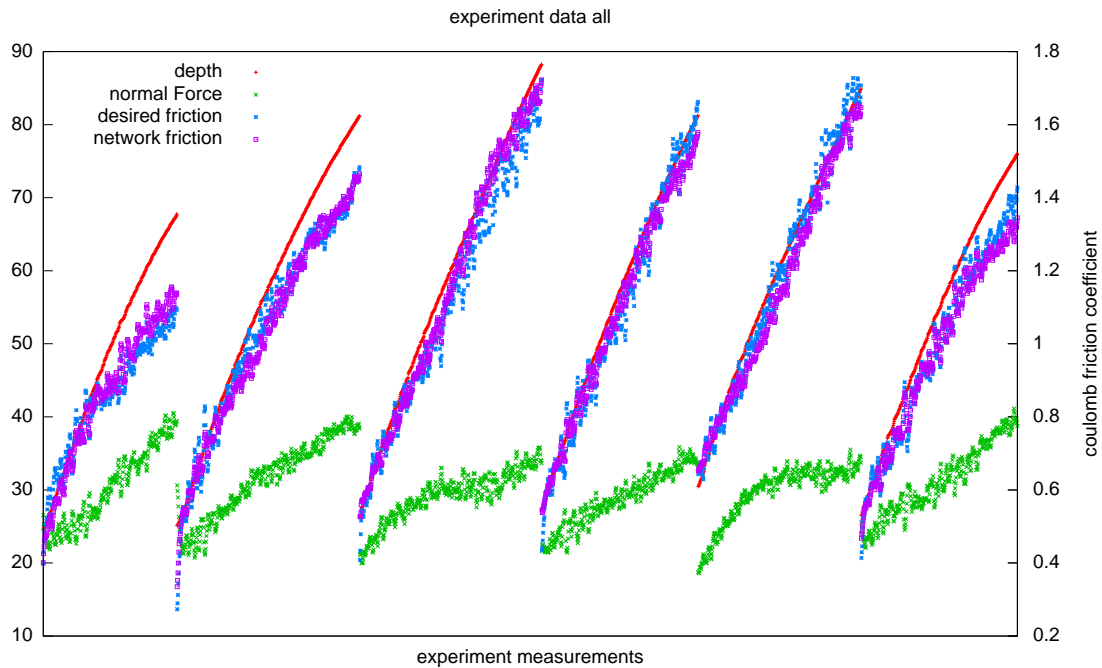


Abbildung 97: Der Graph vergleicht Experimentmessungen die nicht für das Erlernen des Netzes verwendet wurden mit der Netzwerkausgabe. Die Messungen sind aus den selben sechs Experimenten entnommen wie aus Abbildung 96. Im Vergleich zu der prozentualen Abweichung gegenüber den trainierten Messungen von 3.67% beträgt die Abweichung der nicht verwendeten Messungen 3.69%.

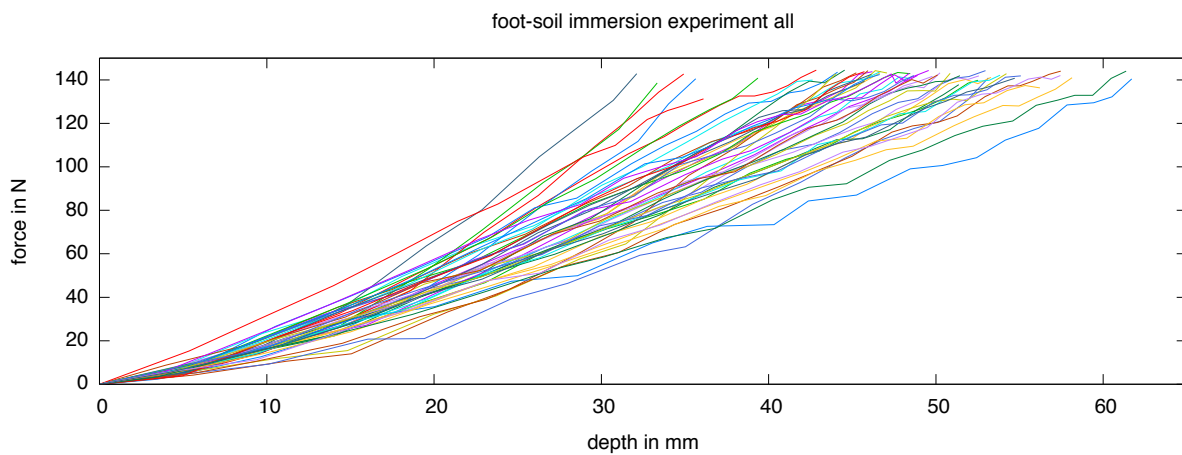


Abbildung 98: Dieser Graph enthält die Daten aller 48 durchgeführten Experimente zum Messen der Normalkräfte beim Eindringen in den Basaltboden.

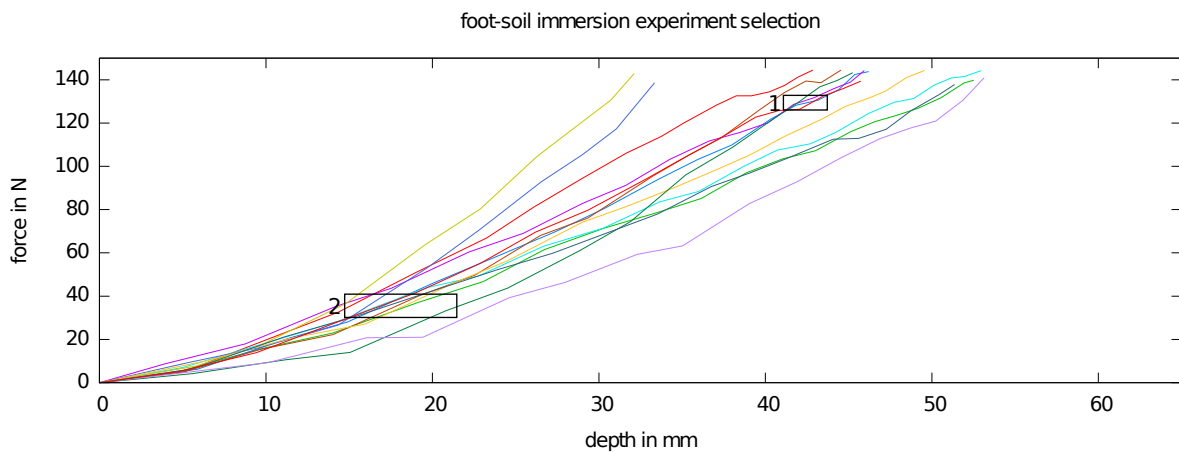


Abbildung 99: Dieser Graph veranschaulicht die unterschiedlichen Kurvenformen der gemessenen Daten. Die vier Kurven innerhalb der 1. Markierung liegen relativ nah beieinander, während sie sich in der 2. Markierung über den gesamten Varianzbereich verteilen.

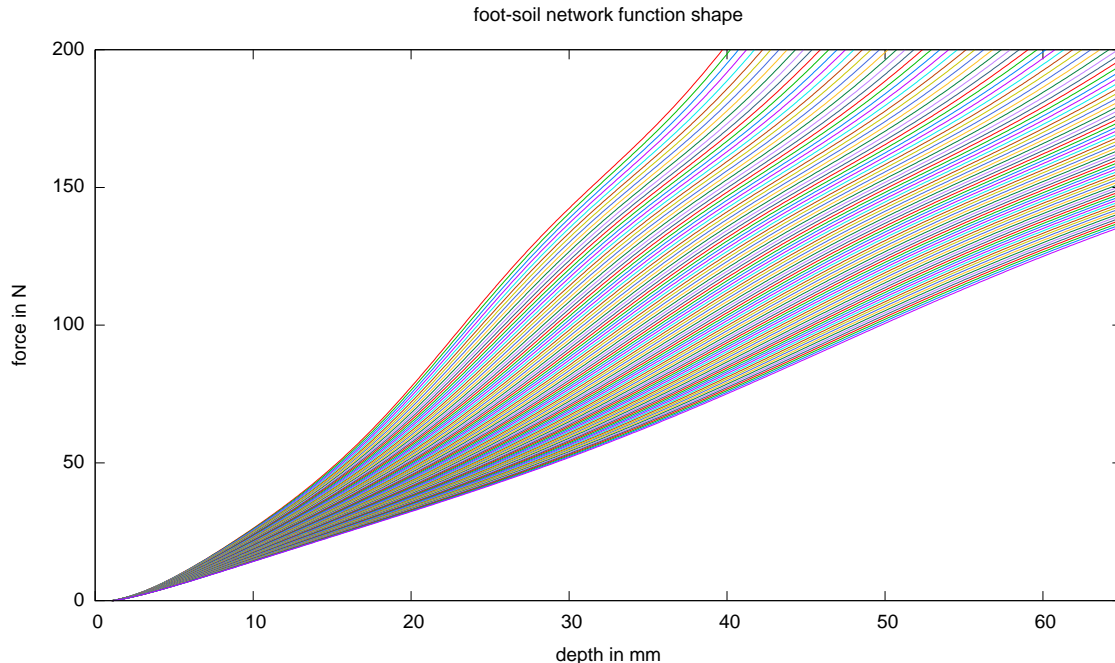


Abbildung 100: Gesamte Netzausgabe geplottet in 100 Kurven.

Für die Untersuchung der lateralen Kräfte in dem Basaltsubstrat wurden 56 Datensätze aufgenommen. Die Daten wurden in Sekundenblöcke unterteilt und innerhalb dieser Blöcke gemittelt. Dadurch haben sich 1015 Datenpunkte ergeben die eine Tiefe, Normalkraft und laterale Kraft beinhalten. Diese Daten sind in [Abbildung 113](#) dargestellt. Besonders hervorzuheben ist der Bereich bei einer Eindringtiefe von 43 mm bis 50 mm. Dort wurden Normalkräfte von 10 N bis 70 N gemessen, während sich die seitlichen Kräfte nur in einer Spanne von 12 N bis 38 N befinden. Daher leitet sich die Annahme ab, dass der lastabhängige seitliche Widerstand relativ gering ist und der Widerstand hauptsächlich über den Bulldozingwiderstand entsteht. Da sich der Bulldozingwiderstand aus der Eindringtiefe ergibt wurde ein neuronales Netz entwickelt, welches eine Abbildung von der Eindringtiefe auf den Bulldozingwiderstand darstellt. Die resultierende Netzausgabe ist in [Abbildung 114](#) dargestellt. In der Simulation kann die aktuelle Eindringtiefe in einen maximalen seitlichen Widerstand umgerechnet werden und in Kombination mit der momentanen Last kann ein Reibungskoeffizient als Simulationsparameter errechnet und verwendet werden.

Nach der Integration in die Simulation hat sich gezeigt, dass die Bodenhaftung des SpaceClimbers relativ gering ist und gerade in Schrägen der Roboter deutlich weniger Haftung hat als es den Erfahrungen mit dem realen System entspricht. Dies führt zu der Annahme, dass das Experimentsetup, in dem die Daten für die Bodenmechanik aufgenommen wurden, nicht ausreichend Informationen liefert um die realen Bedingungen des gesamten Roboters abzubilden. Zum Einen steht die unregelmäßige Steuerung des Manipulatorarmes in Verdacht die zu messenden Kräfte zu reduzieren und zum Anderen konnten dynamische Kontaktbedingungen (Impulskräfte) nicht ausreichend

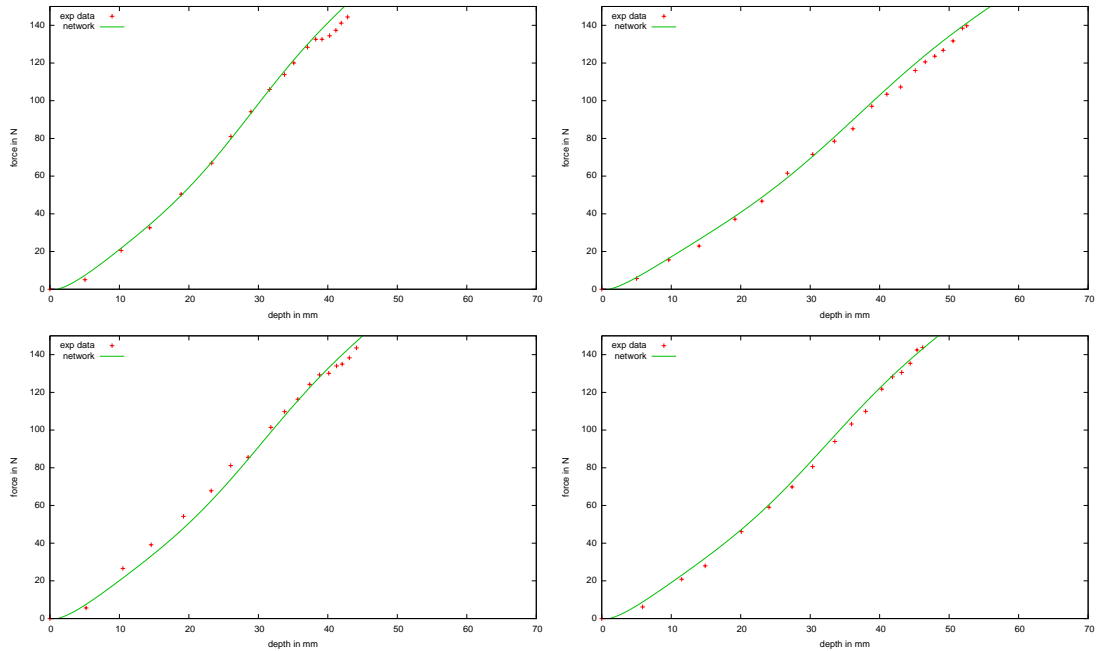


Abbildung 101: Vergleich der Netzwerkausgabe mit realen Experimentdaten 1-4. Gepunktete Kurven sind die realen Daten und die durchgezogene Kurve ist die Netzwerkausgabe.

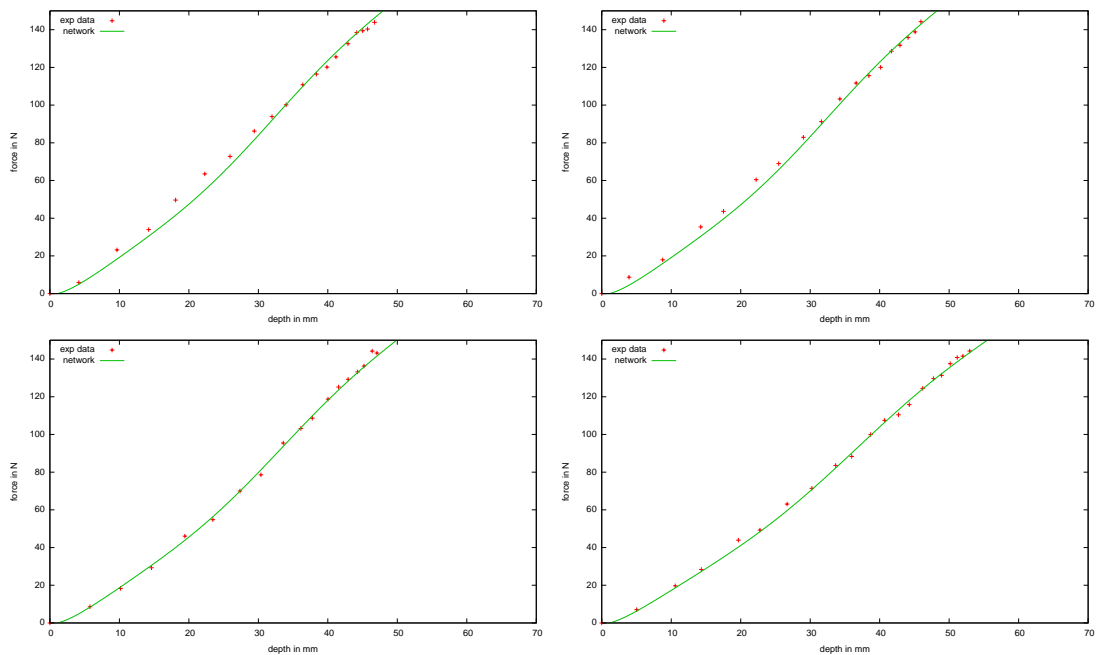


Abbildung 102: Vergleich der Netzwerkausgabe mit realen Experimentdaten 5-8. Gepunktete Kurven sind die realen Daten und die durchgezogene Kurve ist die Netzwerkausgabe.

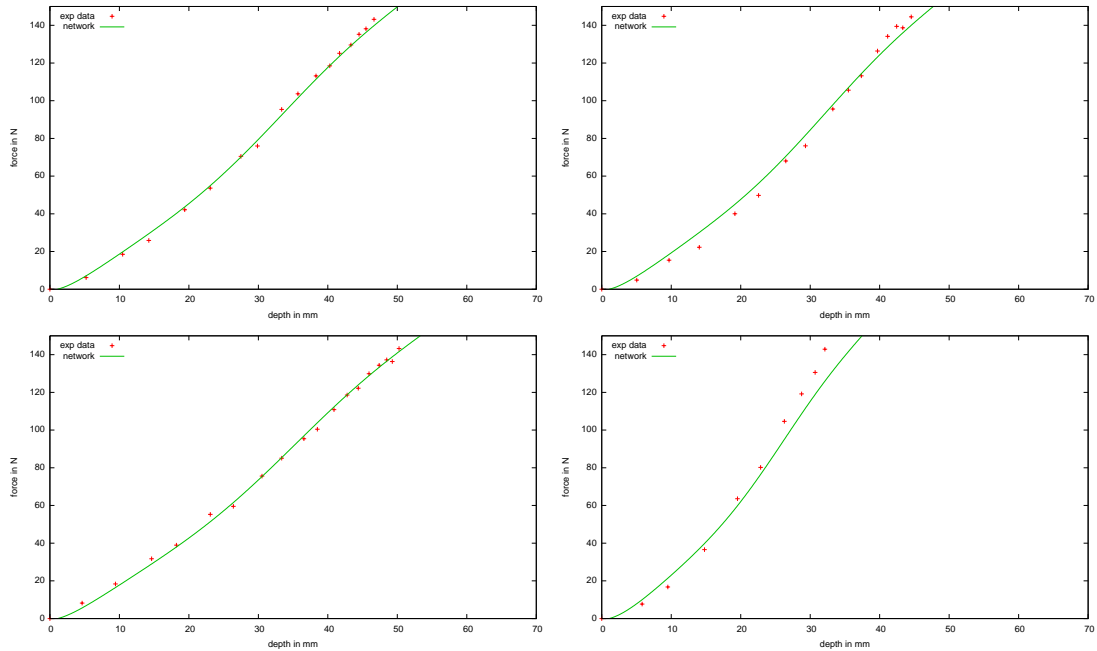


Abbildung 103: Vergleich der Netzwerkausgabe mit realen Experimentdaten 9-12. Gepunktete Kurven sind die realen Daten und die durchgezogene Kurve ist die Netzwerkausgabe.

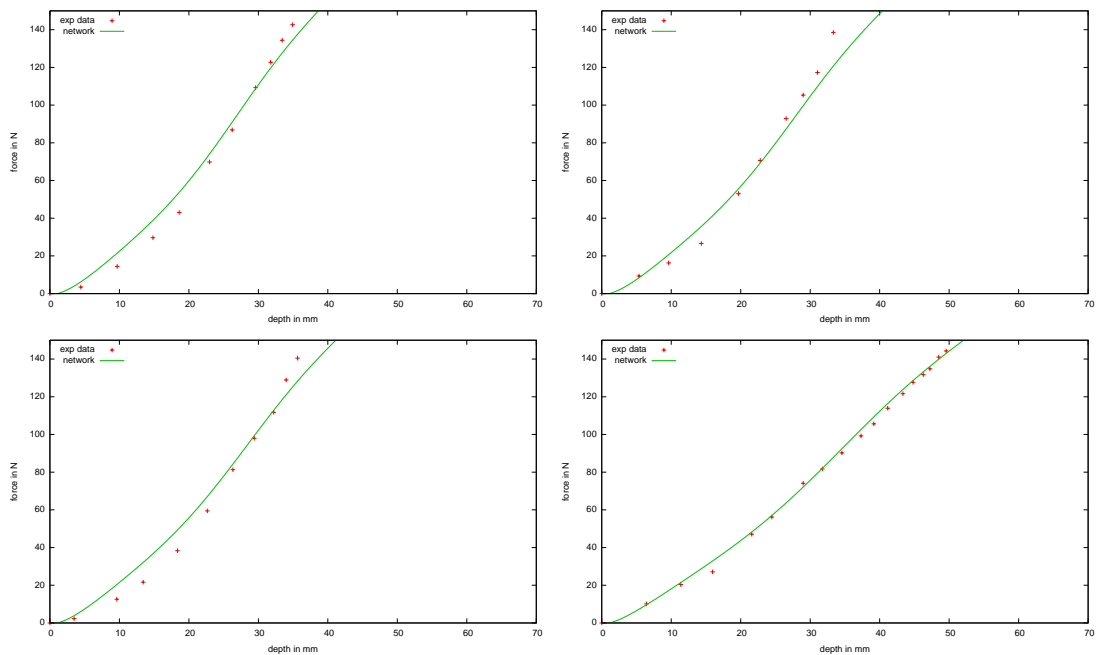


Abbildung 104: Vergleich der Netzwerkausgabe mit realen Experimentdaten 13-16. Gepunktete Kurven sind die realen Daten und die durchgezogene Kurve ist die Netzwerkausgabe.

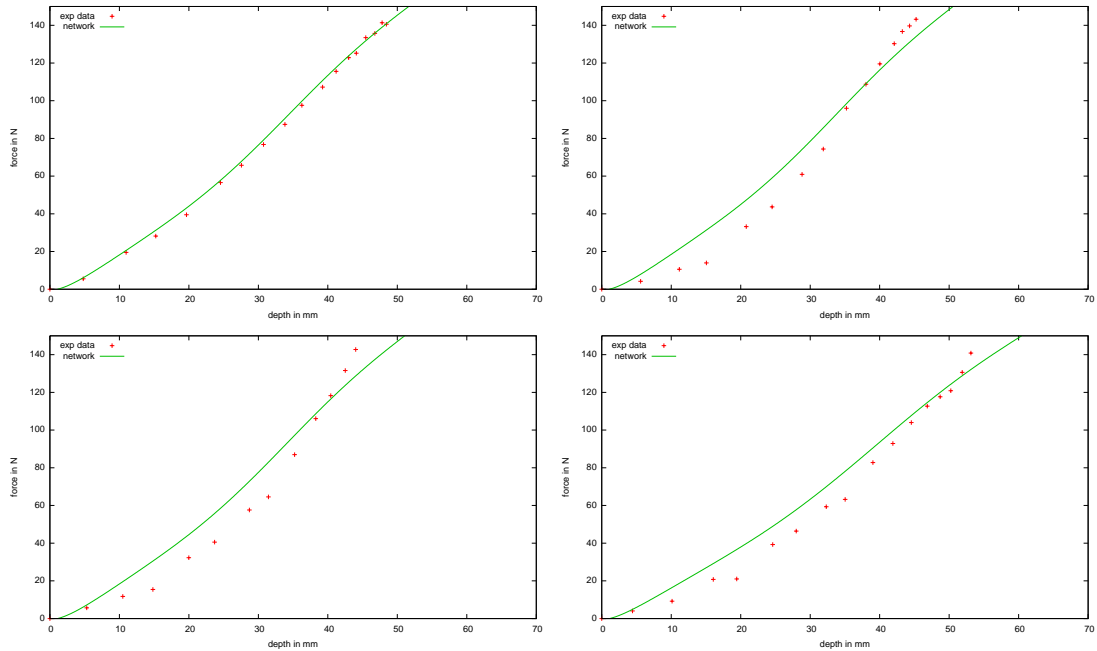


Abbildung 105: Vergleich der Netzwerkausgabe mit realen Experimentdaten 17-20. Gepunktete Kurven sind die realen Daten und die durchgezogene Kurve ist die Netzwerkausgabe.

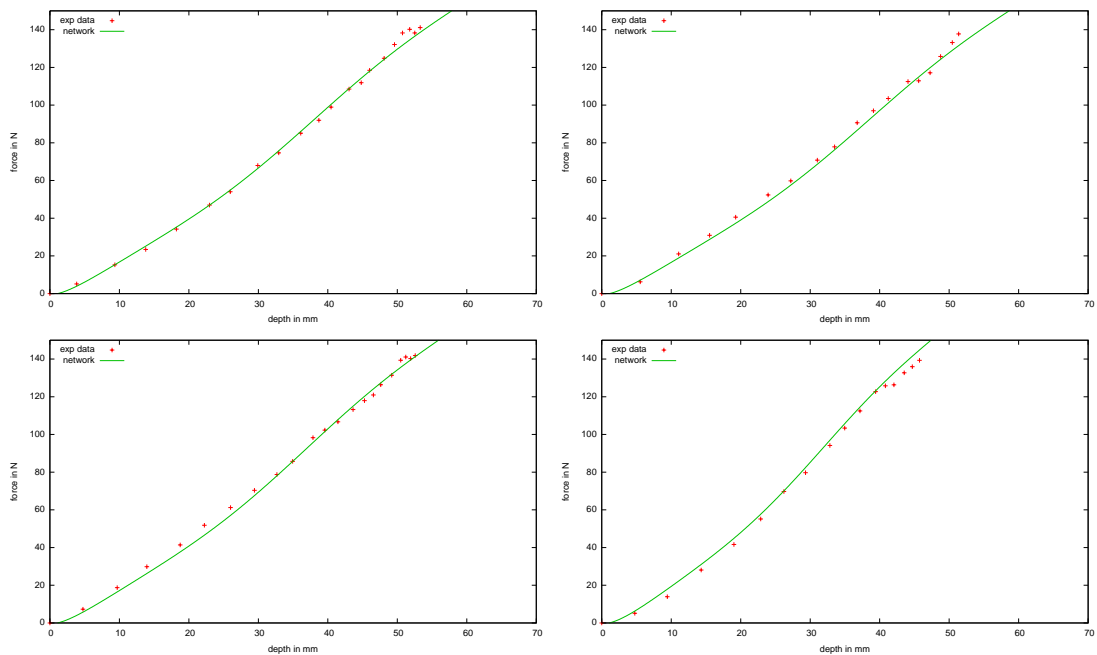


Abbildung 106: Vergleich der Netzwerkausgabe mit realen Experimentdaten 21-24. Gepunktete Kurven sind die realen Daten und die durchgezogene Kurve ist die Netzwerkausgabe.

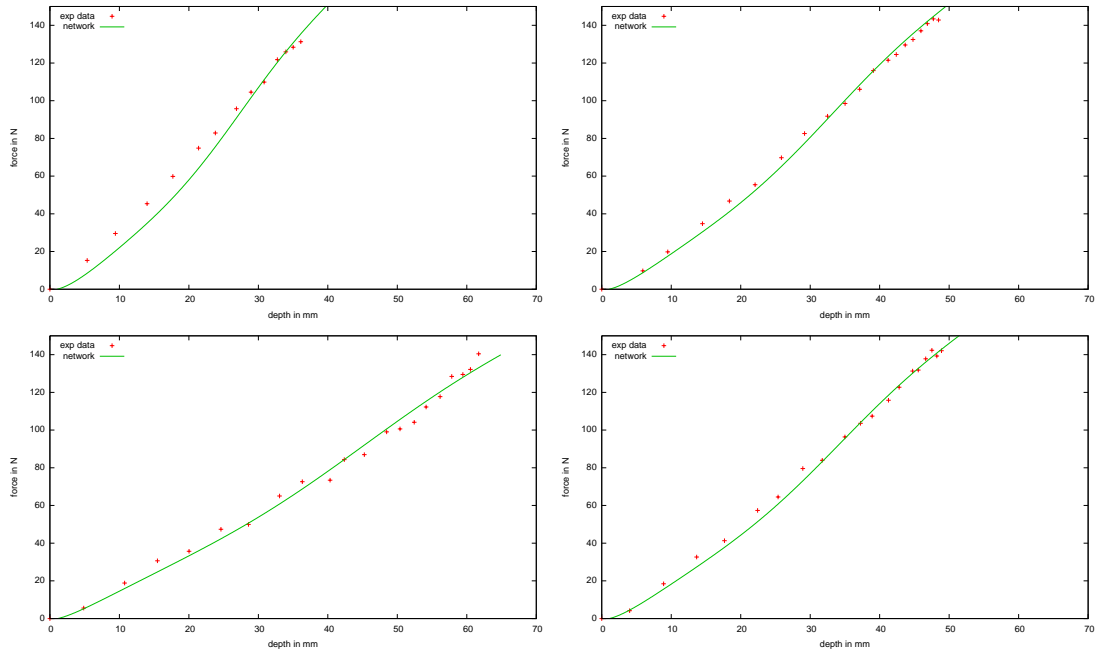


Abbildung 107: Vergleich der Netzwerkausgabe mit realen Experimentdaten 25-28. Gepunktete Kurven sind die realen Daten und die durchgezogene Kurve ist die Netzwerkausgabe.

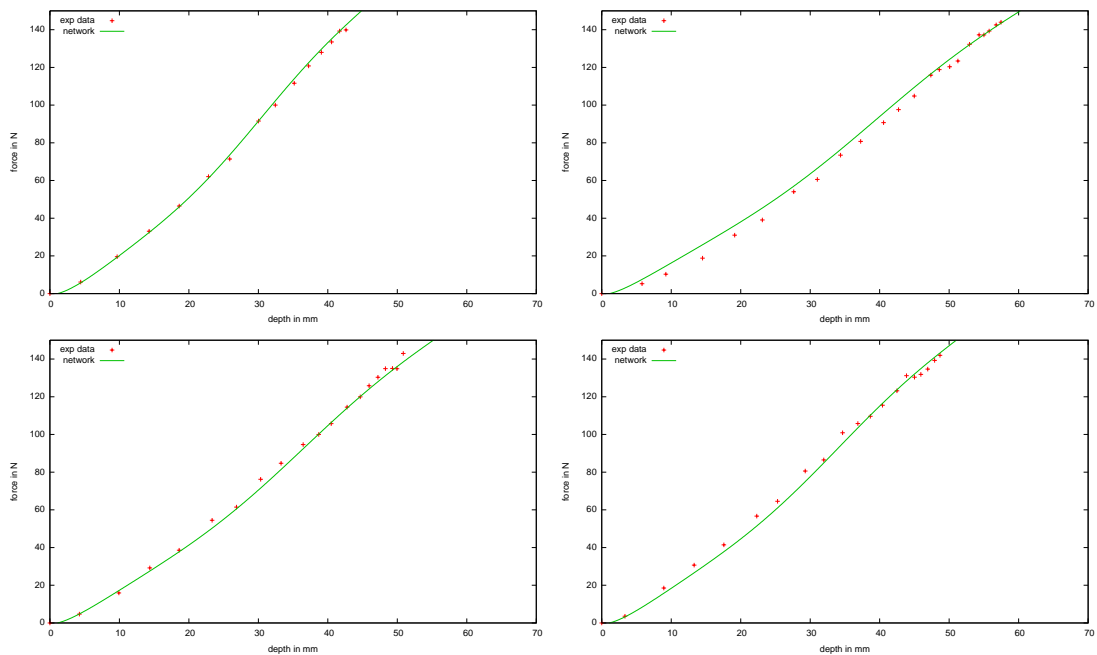


Abbildung 108: Vergleich der Netzwerkausgabe mit realen Experimentdaten 29-32. Gepunktete Kurven sind die realen Daten und die durchgezogene Kurve ist die Netzwerkausgabe.

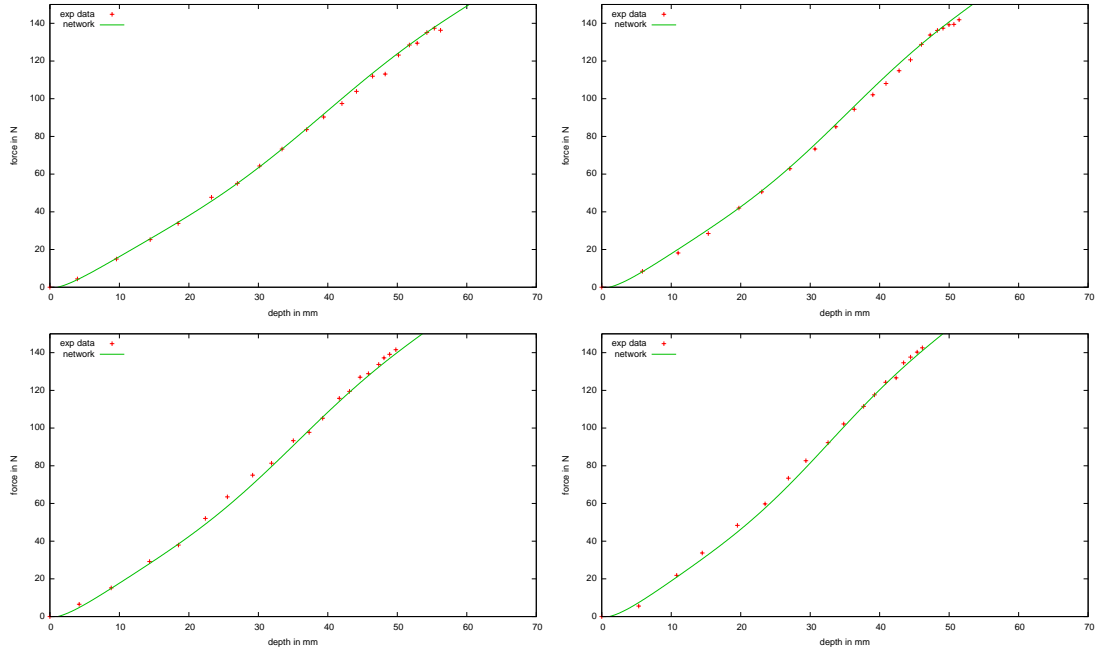


Abbildung 109: Vergleich der Netzwerkausgabe mit realen Experimentdaten 33-36. Gepunktete Kurven sind die realen Daten und die durchgezogene Kurve ist die Netzwerkausgabe.

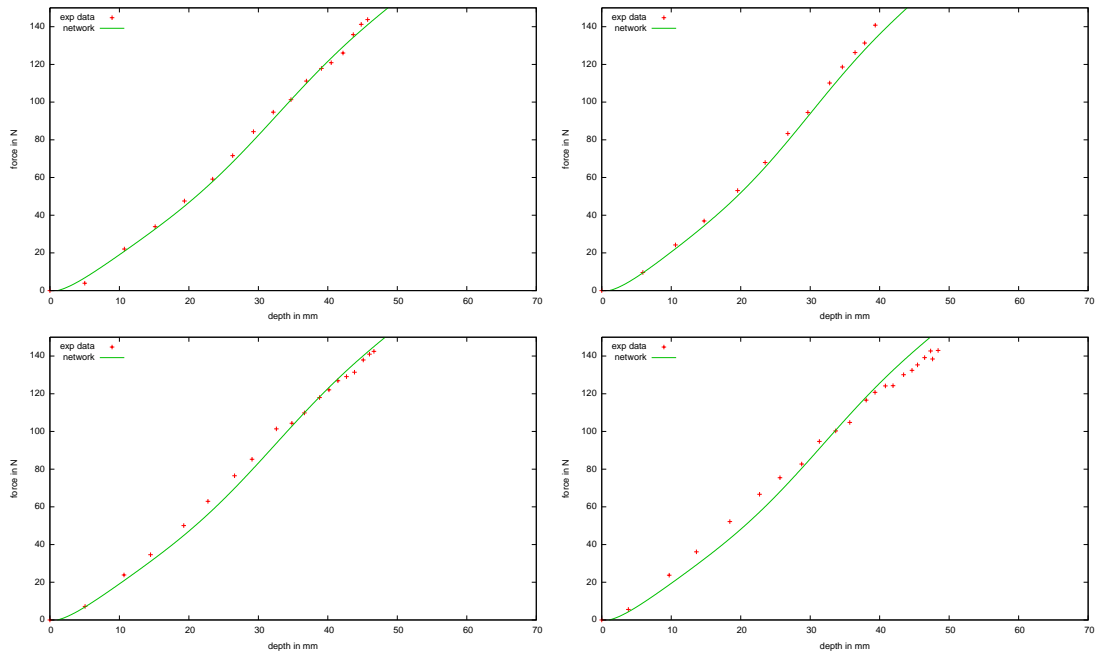


Abbildung 110: Vergleich der Netzwerkausgabe mit realen Experimentdaten 37-40. Gepunktete Kurven sind die realen Daten und die durchgezogene Kurve ist die Netzwerkausgabe.

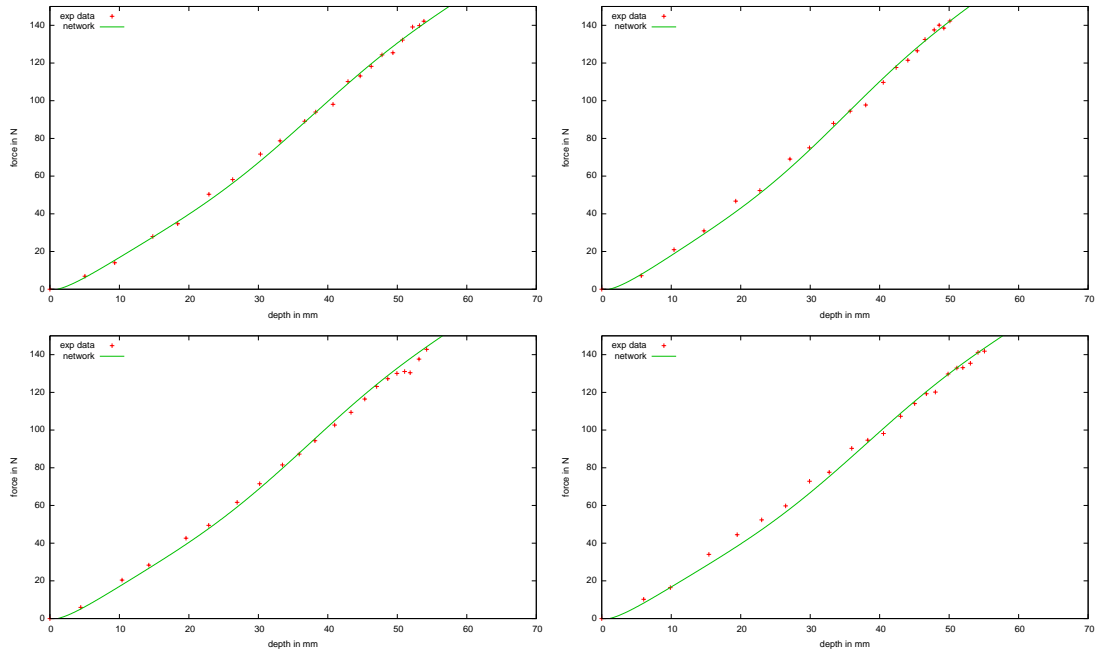


Abbildung 111: Vergleich der Netzwerkausgabe mit realen Experimentdaten 41-44. Gepunktete Kurven sind die realen Daten und die durchgezogene Kurve ist die Netzwerkausgabe.

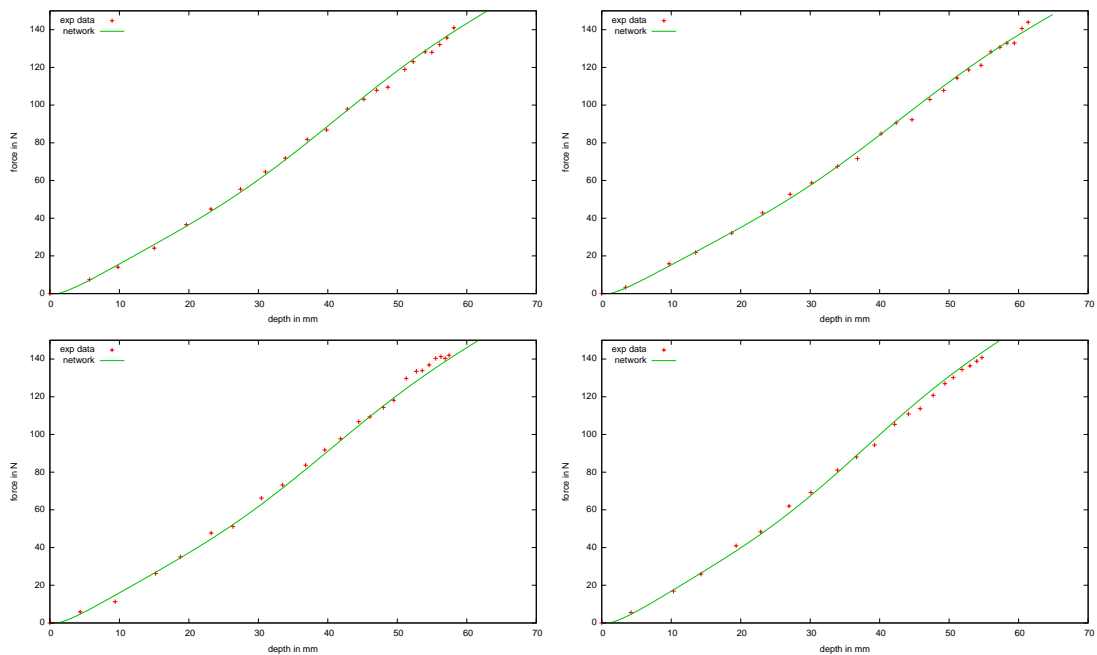


Abbildung 112: Vergleich der Netzwerkausgabe mit realen Experimentdaten 45-48. Gepunktete Kurven sind die realen Daten und die durchgezogene Kurve ist die Netzwerkausgabe.

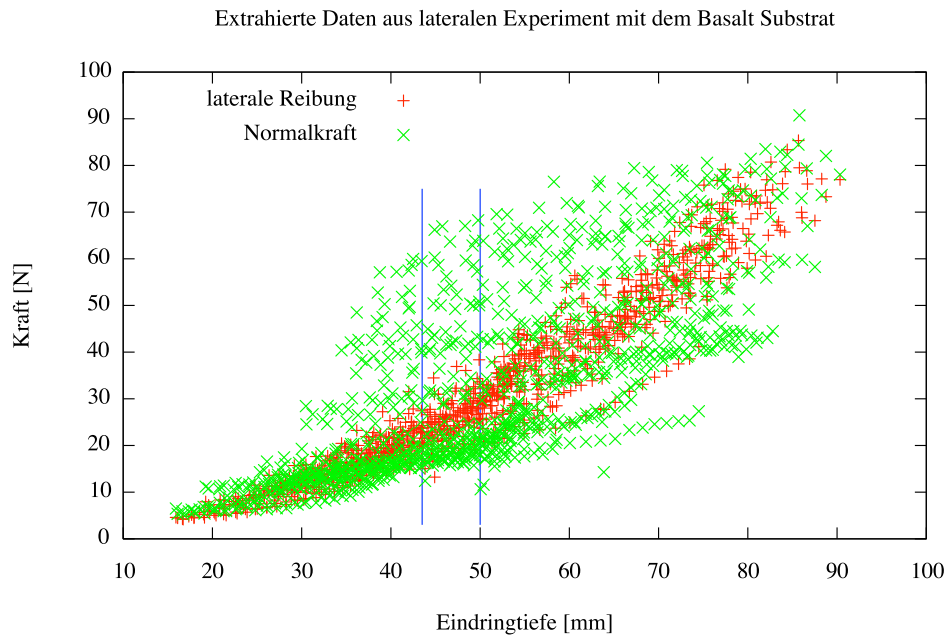


Abbildung 113: Der Graph zeigt die Daten die aus den lateralen Experimenten mit dem Basalt Substrat extrahiert und gemittelt wurden.

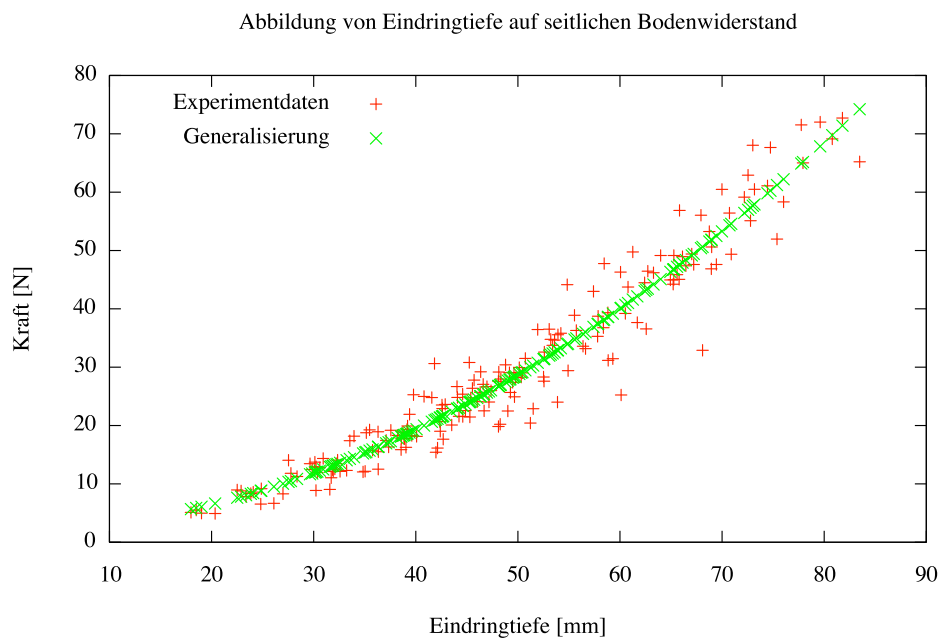


Abbildung 114: Der Graph zeigt die Abbildung der Eindringtiefe auf einen seitlichen Widerstand. In rot sind die aus dem Experiment ermittelten Daten dargestellt und in grün eine Generalisierung über ein neuronales Netz.



untersucht werden. Es konnte lediglich die Beobachtung gemacht werden, dass für verschiedene Geschwindigkeiten beim Eindringen in das Basaltsubstrat, bei gleicher Tiefe, deutlich unterschiedliche Gegenkräfte gemessen wurden.

Um zumindest die Annahme, dass die gemessenen seitlichen Kräfte zu gering sind, zu bestätigen, wurde folgende theoretische Rechnung durchgeführt:

Wir nehmen an, dass der SpaceClimber auf einer Schräge mit 25° steht. Das Gewicht von 25,5 kg ist auf alle Beine gleich verteilt. Daraus ergibt sich eine Last pro Fuss von 4,25 kg (41,693 N). Davon wirken 37,79 N senkrecht auf die Kontaktfläche und 17,62 N seitlich. Bei einer Kontaktkraft von 37,79 N ergibt sich laut Experimentdaten eine Eindringtiefe von 12 mm - 26 mm. Für diese Eindringtiefe wurden maximal 10 N seitlicher Widerstand gemessen (siehe Abbildung 113). Bei benötigten 17,62 N würde der SpaceClimber in einer Schräge von 25° nicht genug Haftung haben und somit abrutschen. In realen Experimenten aus dem SpaceClimber Projekt konnte aber eine Schräge von 25° Grad bestiegen werden.

3.1.42 AP5400: Durchführung der Referenzexperimente

3.1.43 AP5410: Durchführung im realen Testbed

Sensoren

In diesem Abschnitt sind alle Sensor-Experimente zusammengefasst.

Hokuyo URG-04LX Im Rahmen dieser Experimente wurde das Rauschverhalten des Laserscanners Hokuyo URG-04LX in Abhängigkeit von der Untergrundfarbe und der Entfernung zum Ziel ermittelt. Als Firmware wurde die Version 3.3.00 eingesetzt und als Kommunikationsprotokoll SCIP2.0. Es wurden je 1000 Scans aus 21 unterschiedlichen Entfernungen mit je acht verschiedenen Untergrundfarben durchgeführt. Bei diesen 168000 Einzelscans wurde die gerade Entfernung zum Ziel (Step 384) gemessen. Abbildung 115 zeigt den Aufbau des Experiments.

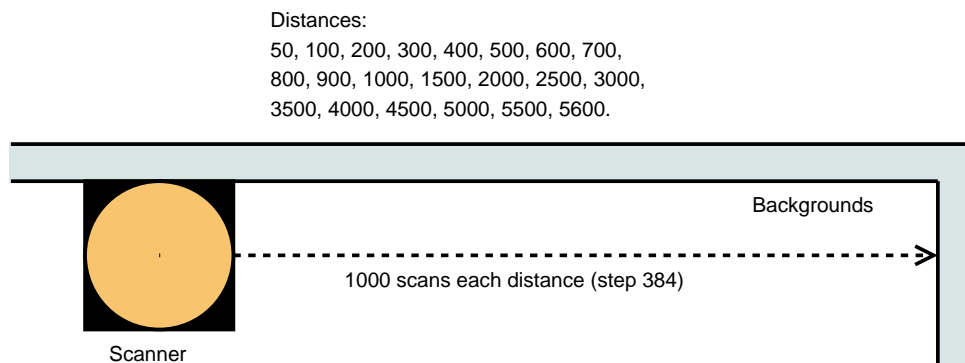


Abbildung 115: Aufbau des Scanner-Experimentes

Durch das für die Durchführung der Experimente implementierte Programm wurden die gemessenen Entfernungen der Einzelscans, die durchschnittlichen Entfernungen für jede Kombination aus Farbe und Entfernung (Tabelle 11) und die Abweichungen zum Erwartungswert protokolliert und berechnet.

Durchschnittliche Entfernungen In der nachfolgende Tabelle mit den durchschnittlichen Entfernungen ist die kleinste gemessene Entfernung je Entfernungsschritt rot und die größte grün gekennzeichnet. Hierbei wurden schwarz und weiß nicht berücksichtigt. Ein Wert von 0 bedeutet, dass nur Fehlerwerte (Werte < 20) empfangen wurde, was generell bedeutet, dass das reflektierte Licht eine zu geringe Intensität hatte. Die nachfolgende Tabelle mit den durchschnittlichen Entfernungen zeigt, dass weiße Objekte generell näher am Scanner zu liegen scheinen und schwarze weiter entfernt. Bezüglich den farbigen Untergründen kann gesagt werden, dass die Farbe Grün geringere und die Farbe Orange größere Entfernungswerte zur Folge hat.



distances	white	black	red	purple	blue	green	yellow	orange
50	53.959	47.6021	55.3844	54.645	48.909	51.166	55.092	56.037
100	95.607	101.303	101.541	99.578	100.54	101.273	101.778	101.237
200	193.932	199.523	200.084	199.206	196.442	194.757	202.432	203.998
300	298.948	297.896	308.081	306.265	302.054	299.124	309.018	310.245
400	408.573	398.192	410.13	409.157	405.541	401.435	410.729	411.935
500	500.735	503.992	510.2	512.18	506.455	503.895	512.458	513.824
600	603.538	607.268	610.22	609.1	604.73	603.028	609.477	611.383
700	705.507	709.647	706.902	705.019	703.013	700.597	704.567	708.143
800	791.383	810.543	805.942	805.34	803.926	802.147	789.601	807.761
900	888.968	911.567	906.857	905.454	903.56	902.832	905.899	906.701
1000	989.882	995.782	1001.32	1003.56	1001.1	1001.56	1001.8	1003.84
1500	1438.96	1509.61	1509.84	1508.64	1510.4	1511.46	1510.58	1508.28
2000	1939.49	2018.01	2010.38	2008.14	2004.51	2000.08	2008.08	2010.66
2500	2439.52	2521.79	2512.97	2511.32	2513.88	2515.95	2512.76	2515.79
3000	2875.31	3005.91	3013.38	3016.24	3019.09	3019.72	3014.06	3011.03
3500	3450.41	3523.54	3519.34	3523.18	3524.95	3525.76	3520.45	3522.35
4000	3956.74	4028.78	4026.48	4026.12	4025.31	4025.85	4026.54	4025.27
4500	2859.64	0	5164.5	0	0	5166	5178.5	5132.33
5000	0	0	0	0	0	0	0	0
5500	5495.04	5513	5514.8	5513.73	5515.46	5512.48	5516.2	5514.88
5600	0	5511.93	0	0	0	0	0	0

Tabelle 11: Durchschnittliche Entfernungen über 1000 Scans

Abweichungen Bei dieser Auswertung wurden von den gemessenen Entfernungen die jeweiligen Sollwerte subtrahiert. Auf diese Weise sollte annähernd eine Normalverteilung um 0 auftreten, womit das Rauschverhalten des Scanners beschrieben werden kann. Die meisten Abweichungen lagen zwischen -10 und 30. Generell liefert der Scanner also um 10 mm längere Entfernungen (Abbildung 116(c)) als die tatsächliche Distanz beträgt. Kleinere Spitzen traten zwischen Abweichungen von -513 bis -468 mm (Abbildung 116(a)) und -105 bis -75 mm (Abbildung 116(b)) auf. Insgesamt traten zwischen -5497 und 707 mm 433 unterschiedliche Abweichungen auf, wobei die meisten nur einmalig auftraten. Der Scanner lieferte während der Experimente 24697 Fehlerwerte (Abbildung 116(d)), hauptsächlich da die Intensität des reflektierten Lichtes nicht ausreichte (Fehlerwert 1). Dieser Fehler tritt verhältnismäßig oft bei schwarzen Untergründen auf, was ein bekanntes Problem des Scanners ist.

Winkelsensor SpaceClimber Gelenk Im Rahmen von Referenzexperimenten hat sich die hohe Genauigkeit der in den SpaceClimber-Gelenken eingesetzten Sensoren bestätigt.

In den Experimenten wurde mit einem SpaceClimber-Gelenk ohne Last mehrfach von beiden Richtungen eine bzw. mehrere Positionen angefahren und in jeder Position wurden 100 Messwerte aufgezeichnet. Ziel dieser Experimente war es das Rauschverhalten des auf dem Hall-Effekts basierenden Sensor im SpaceClimber-Gelenk zu untersuchen.

Beim Inkrementalsensor werden 360° auf 6000 Schritte verteilt. Dies entspricht einer

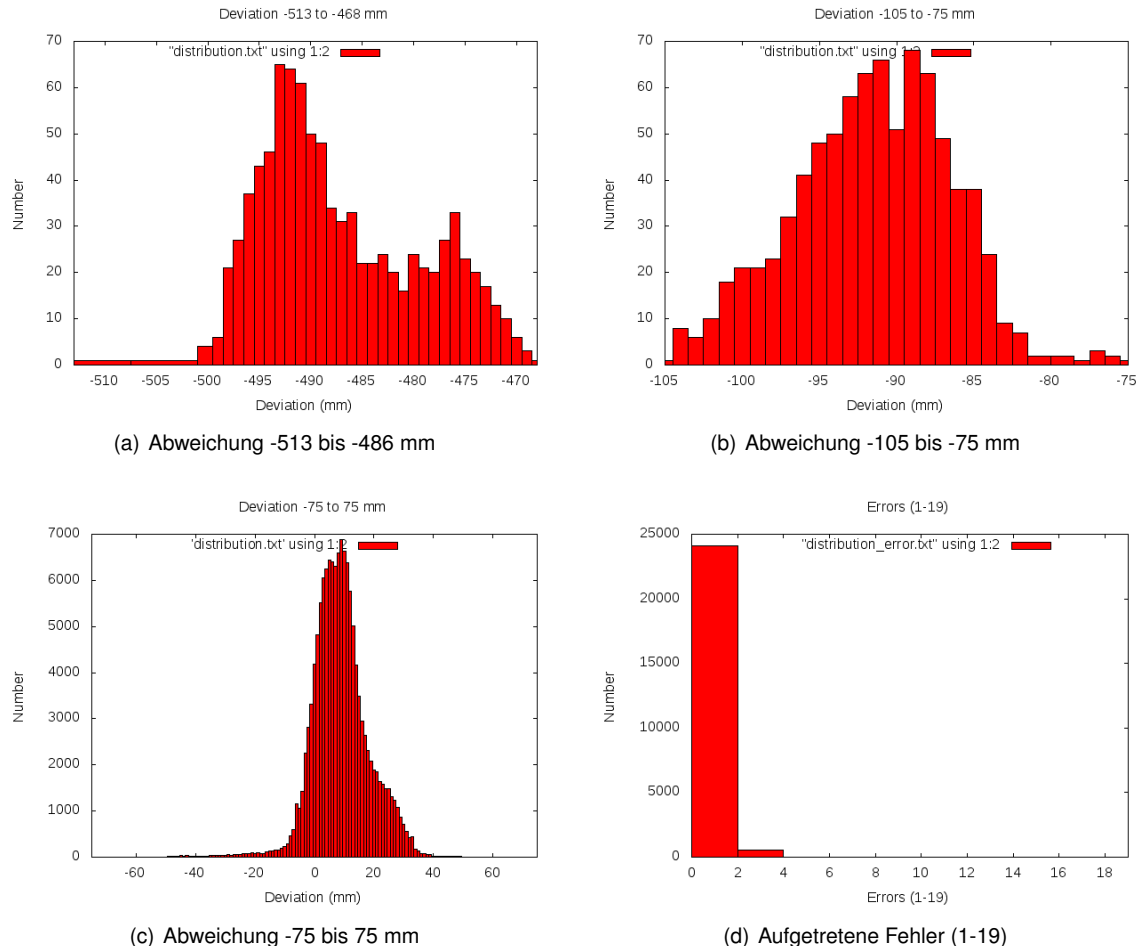


Abbildung 116: Abweichung der Messwerte und Fehlmessungen über alle Scans

Auflösung von $0.06^\circ/\text{Schritt}$. Die einzelnen Positionen können hierbei mit einer Genauigkeit von $\pm 0.06^\circ$ angesteuert werden.

Beim Absolutsensor handelt es sich um einen 12bit Sensor, dessen 4096 Schritte auf 720° verteilt sind, was einer Auflösung von $0.176^\circ/\text{Schritt}$ entspricht. Die Positionen können um $\pm 0.176^\circ$ erreicht werden.

Bei beiden Sensoren tritt das Rauschen in Abhängigkeit zum Drehmoment auf. So nimmt die Häufigkeit innerhalb der oben erwähnten Grenzen bei steigendem Drehmoment zu. Ohne Belastung der Gelenke tritt kein Rauschen auf. Aktuell wird beim SpaceClimber auf höherer Ebene kein Rauschen mehr angezeigt und entsprechen ohne Rauschen gearbeitet.

Aus diesem Grund und der insgesamt sehr hohen Genauigkeit der Sensoren kann das Rauschen der Gelenke in der Simulation ebenfalls vernachlässigt werden, da der Einfluss auf die Verwendbarkeit der Simulationsergebnisse nicht relevant ist.



GPS-Sensor Das Referenzexperiment zum GPS-Sensor wurde nicht durchgeführt, da der SpaceClimber nicht über einen GPS-Empfänger verfügt. Zudem ist es aufgrund seines Einsatzgebietes „Weltraum“ unwahrscheinlich, dass er in Zukunft mit einem solchen Sensor ausgestattet wird.

Es wurden jedoch Überlegungen im Rahmen der Referenzexperimente angestellt, wie ein GPS-Sensor oder ein ähnliches Ortungsgerät simuliert werden kann.

Zug-Druck-Kraftsensor Es wurde ein Experiment zum Rauschverhalten des Kraft- und Drehmomentsensors Mini45 (Modell SI-580-20) der Firma ATI durchgeführt. Dieser Sensor ist im SpaceClimber jeweils zwischen dem Körper und den Beinen angebracht und ist deswegen für die Simulation interessant. In der Definition zu diesem Experiment wurde noch von dem Modell SI-145-5 gesprochen. Der Unterschied bei den beiden verschiedenen Modellen ist ihr Messbereich. Zum Zeitpunkt als die Definition geschrieben wurde, war noch der IS-145-5 für den SpaceClimber vorgesehen.

In der Abbildung 117 ist der Aufbau für das Experiment zu sehen. Der Sensor wurde unbelastet vermessen. Im Anschluss wurde der Sensor mit verschiedenen Gewichten belastet und im Anschluss nochmal unbelastet vermessen. Dies wurde sowohl für die X-Achse und Y-Achse vorgenommen. Die Z-Achse wurde nur mit Gewichten belastet. Hier wurde keine Messung zum Drehmoment gemacht. Es wurde für jede Situation eine Datei mit den Werten über 30 Sekunden geschrieben. In Abbildung 118 ist die Messung für die Kraft auf der X-Achse und das Drehmoment auf der Y-Achse mit 2,544 kg an einem 105 mm langen Hebel zu sehen.

Diese Messungen werden nun an unseren Projektpartner weiter gegeben, so dass dort das Experiment im virtuellen nachgebaut werden kann, um dann den simulierten Sensor an den realen anzupassen.

PMD-Kamera Es wurde eine Kamera der Firma SwissRanger die SR4000 untersucht, welche in Abbildung 119 zu sehen ist. Da der SpaceClimber zum Zeitpunkt der Experimentdurchführung über keine solche Kamera verfügte wurde eine Kamera gewählt, die möglicherweise auf dem SpaceClimber eingesetzt werden könnte. Die SR4000 besitzt einen PMD-Sensor (Photomischdetektor) welcher Tiefenbilder mit einer Auflösung von 176 x 144 Pixel und einer Bildwiederholrate von bis zu 50 Bilder aufweist. Die Bildwiederholrate hängt von den Einstellungen der Kamera ab. Die Kamera hat einen Messbereich von 0,8 bis 8,0 Meter. Werte von 8,0 bis 10,0 Meter werden von der Kamera auch erfasst, diese wurden aber nicht vom Hersteller kalibriert.

Zur Ermittlung der Werte wurde die Kamera auf eine weiße ebene Wand gerichtet. Danach wurden in verschiedenen Entfernungen (von 100 cm bis 300 cm in 50 cm Schritten) jeweils 100 Messungen aufgenommen. Beim zweiten Durchgang (run2_XXXX.dat) waren alle Filter deaktiviert. Dies wurde einmal mit einem ANF (adaptive neighborhood filter) durchgeführt (run1_XXXX.dat).

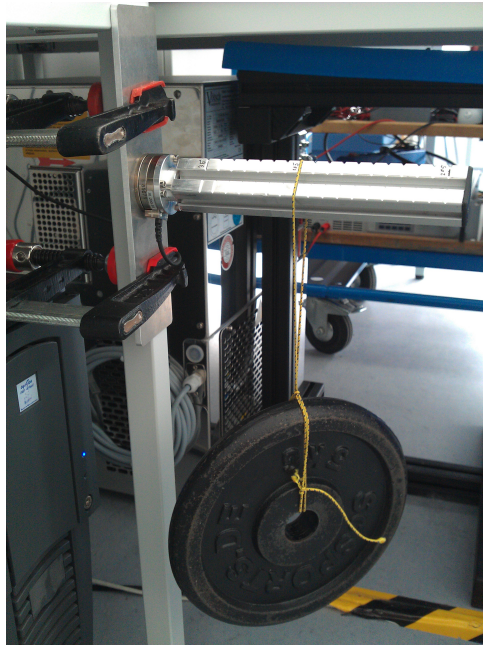


Abbildung 117: Aufbau der Experimente zum Zug-Druck-Kraftsensor.

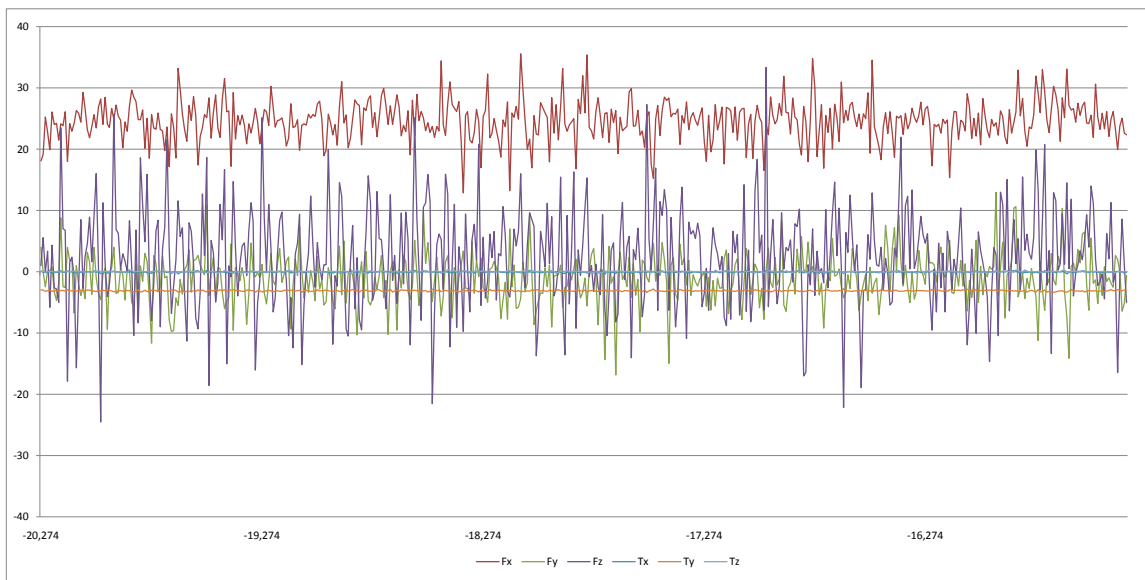


Abbildung 118: Experimente mit 2,544 Kg an einem 105 mm langen Hebel.

Motorteststand Es wurden Referenzexperimente am Motorteststand (AP3400) durchgeführt. Dabei wurde der SpaceClimber Motor unter Umgehung des Position- und Speed-Controllers mit verschiedenen PWM-Signalen direkt angesteuert. Unter verschiedenen Lasten wurden dann die Kenngrößen des Motors gemessen. In Tab. 12 sind die verschiedenen durchgeführten Experimente gelistet und in Abb. 3.1.43 ist eine Versuchsreihe exemplarisch dargestellt.



Abbildung 119: PMD-Kamera SwissRanger SR4000

Tabelle 12: Am Motorteststand durchgeführte Experimente

		PWM-Signal					
		10	50	100	150	200	250
Last	0 kg	x	x	x	x	x	x
	3.81 kg	x	x	x	x	x	x
	6.31 kg	x	x	x	x	x	x
	8.81 kg	x	x	x	x	x	x
	11.31 kg	x	x	x	x	x	x
	13.81 kg	x	x	x	x	x	x
	18.81 kg		x	x	x	x	x
	23.81 kg		x	x	x	x	x
	33.81 kg		x	x	x	x	x

Bodendynamik Eine Anzahl von Referenzexperimenten wurde in der Testumgebung eingerichtet und ausgeführt. Das Ziel dieser Referenzexperimente ist die Erstellung einer Referenzdatenbank echter Messdaten, die für die Verifikation der Simulation genutzt werden können und später verwendet werden können, um die Parameter des Bodens zu bestimmen.

Bei der Durchführung der geplanten Referenzexperimente wurde das Gesamtsystem bestehend aus verschiedenen Testobjekten, dem DFKI-Roboter-Arm "MrSemProm" und die entwickelte Steuerungs und Protokollierungssoftware verwendet. Die Referenzexperimente wurden in AP3400 mit den folgenden Parametern definiert:

- A_1 : Tiefe der Bewegung in den Boden mit einem Angriffswinkel von α .
- A_2 : die Distanz, die im Boden zurückgelegt wurde.
- A_3 : Drehung im Boden mit einem Winkel von β .

Diese Parameter werden in Abb. 3.1.43 gezeigt. Diese Parameter wurden bei der Durchführung der Experimente einzeln und in Kombination verwendet. Die verschie-

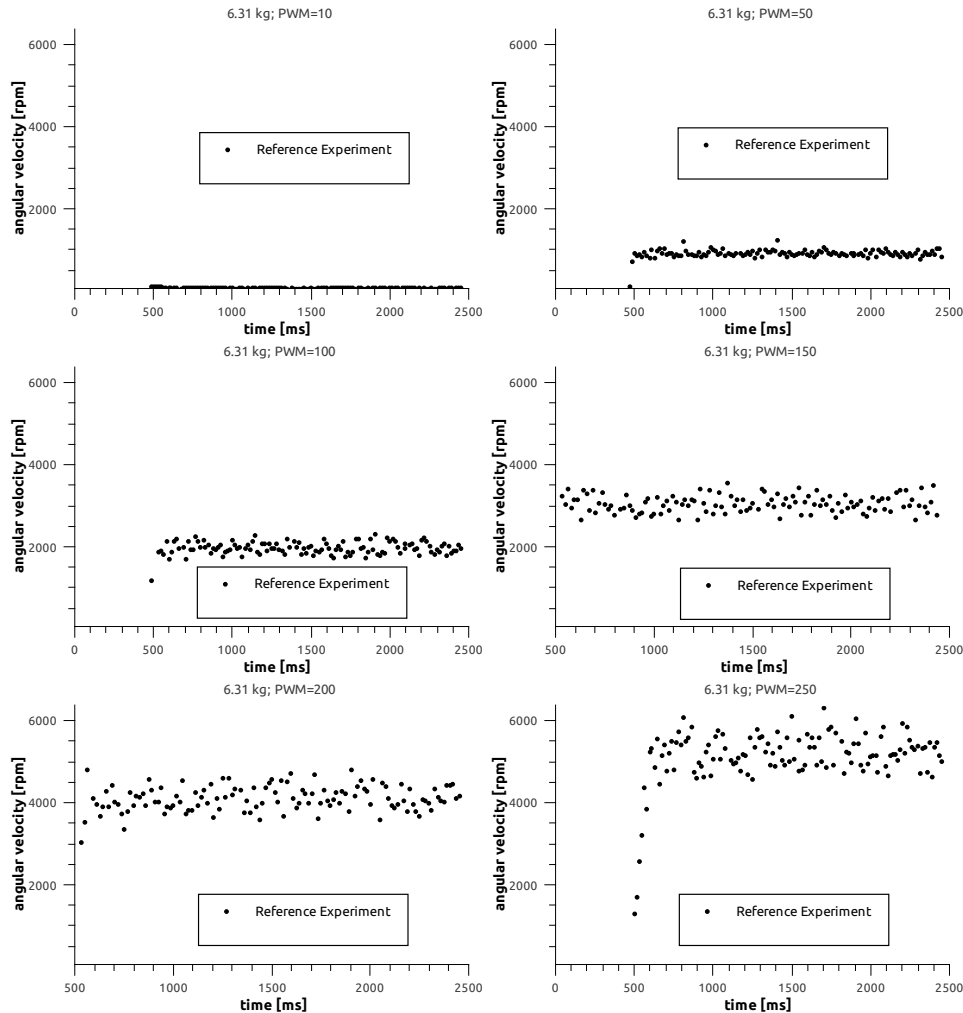


Abbildung 120: Exemplarische Daten aus dem Motorteststand Referenzexperiment. In diesem Beispiel betrug die Last 6.31 kg. Die Geschwindigkeiten wurden im Motor vor der Übersetzung durch das Getriebe gemessen.

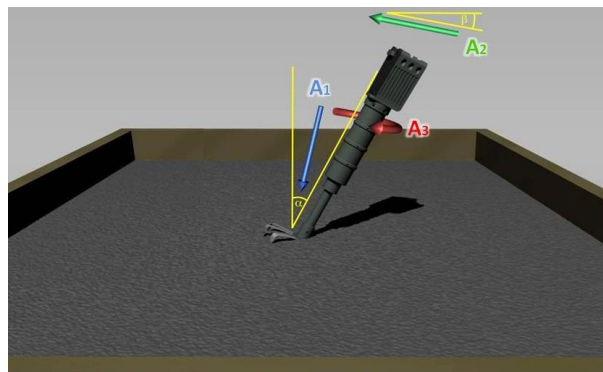


Abbildung 121: Parameter der Referenzexperimente

denen Testobjekte die in den Experimenten verwendet werden um den Boden zu berühren (gezeigt in Abb. 122) werden in Tabelle 13 beschrieben. In allen Experimenten

Name	Beschreibung
C60	Zylinder mit einem Durchmesser von $\varnothing = 60$ mm, Höhe $h = 105$ mm and Masse of $m = 893$ g
B10	Quader mit den Maßen $L \times W \times H = 80 \times 85.44 \times 44$ in [mm] und einer Masse von $m = 900$ g
D70	DLR-Platte mit einem Durchmesser von $\varnothing = 70$ mm und einer Masse von $m = 204$ g

Tabelle 13: Beschreibung der Testobjekte.

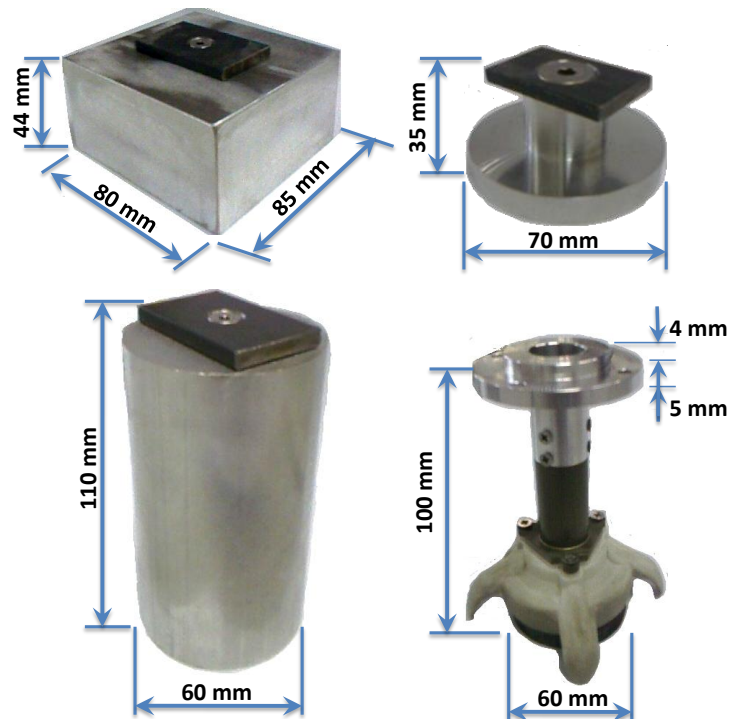


Abbildung 122: Einige der verschiedenen Testobjekte, die in den realen Bodenmechanik Referenzexperimenten verwendet werden.

die im Folgenden beschrieben werden, wurden die Experiment-Konfigurationen aus Tabelle 14 verwendet und resultierenden Messungen inklusive der Drehwinkel der sieben Gelenke (in [rad]) und des Endeffektors des Roboterarms, die Werte des Kraft-Drehmoment-Sensors (Kraft in [N] und Drehmoment in [Nm]) und die Positionen des Endeffektors in kartesischen Koordinaten wurden automatisch aufgezeichnet und erhalten einen Zeitstempel durch die entwickelte Protokollierungssoftware.

Ein Beispiel, wie die Einträge in Tabelle 14 zu verstehen sind: Das Referenzexperiment RE41 wurde durchgeführt, indem der Arm die Anweisung erhielt, den Fuß einen Schritt in einem Angriffswinkel von 0° auf den Boden zuzubewegen und dann 10mm tief in den Boden einzudringen. Dann wurde der Fuß in dieser Tiefe über eine Distanz von 100mm gezogen mit einem Bewegungs-Kontroll-Zyklus (T_c) von 50 ms und die Messwerte wurden alle (T_L) 50 ms protokolliert. Bild 123 zeigt dieses Experiment.

Experimentergebnisse

Die Abbildungen 124 bis 133 zeigen eine Teilmenge der Experimentergebnisse. In all diesen Plots sind die grünen vertikalen Linien die Protokollmarkierungen und die



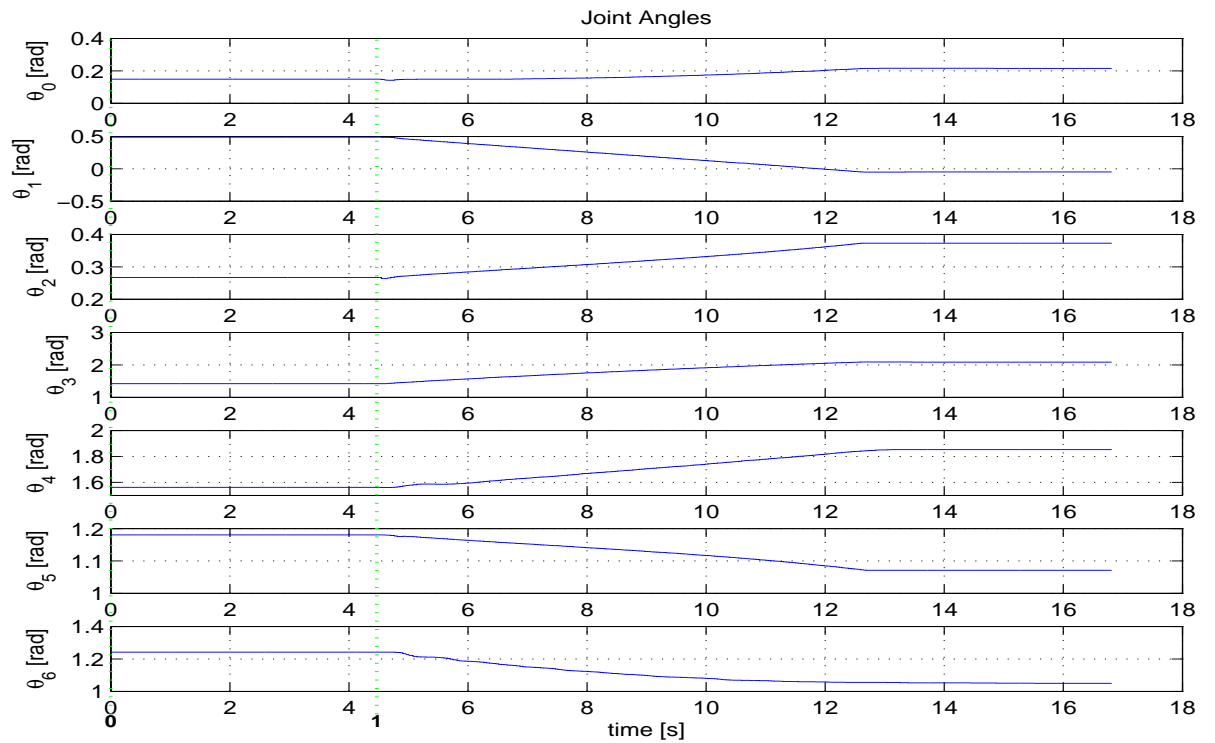
Ref.	Object	α [°]	β [°]	A_1 [mm]	A_2 [mm]	A_3 [°]	T_c [ms]	T_L [ms]
RE44	C60	0	0	10	100	0	50	850
RE24	C60	0	0	20	100	0	50	850
RE09	D70	0	0	100	300	90°	50	850
RE30	D70	0	0	20	100	90°	50	850
RE54	D70	0	0	30	100	90°	50	850
RE20	D70	0	0	20	0	10°	50	850
RE37	D70	0	0	20	200	20°	50	50
RE45	D70	0	0	20	200	20°	30	50
RE41	B10	0	0	10	100	0	50	50
RE12	B10	0	0	10	0	20°	50	50

Tabelle 14: Experiment-Parameter, T_c ist die Länge des Bewegungs-Kontroll-Zyklus und T_L ist die Länge des Messungs-Protokollierungs-Zyklus.

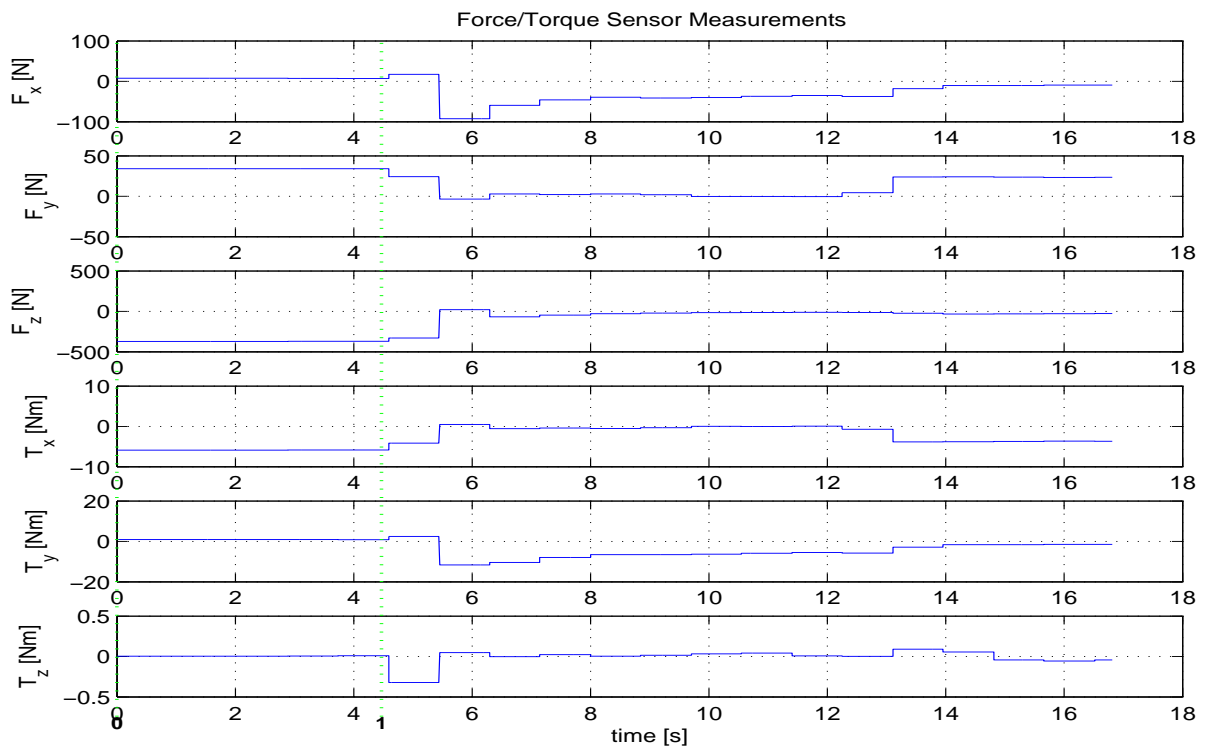


Abbildung 123: Referenzexperiment RE41 unter Verwendung des Quaders (B10) um die Boden-Interaktionskräfte zu messen.

roten sind die Wegmarkierungen der Bewegung (ihre Beschriftungen entsprechen den Punkt-IDs, die ebenfalls protokolliert werden).

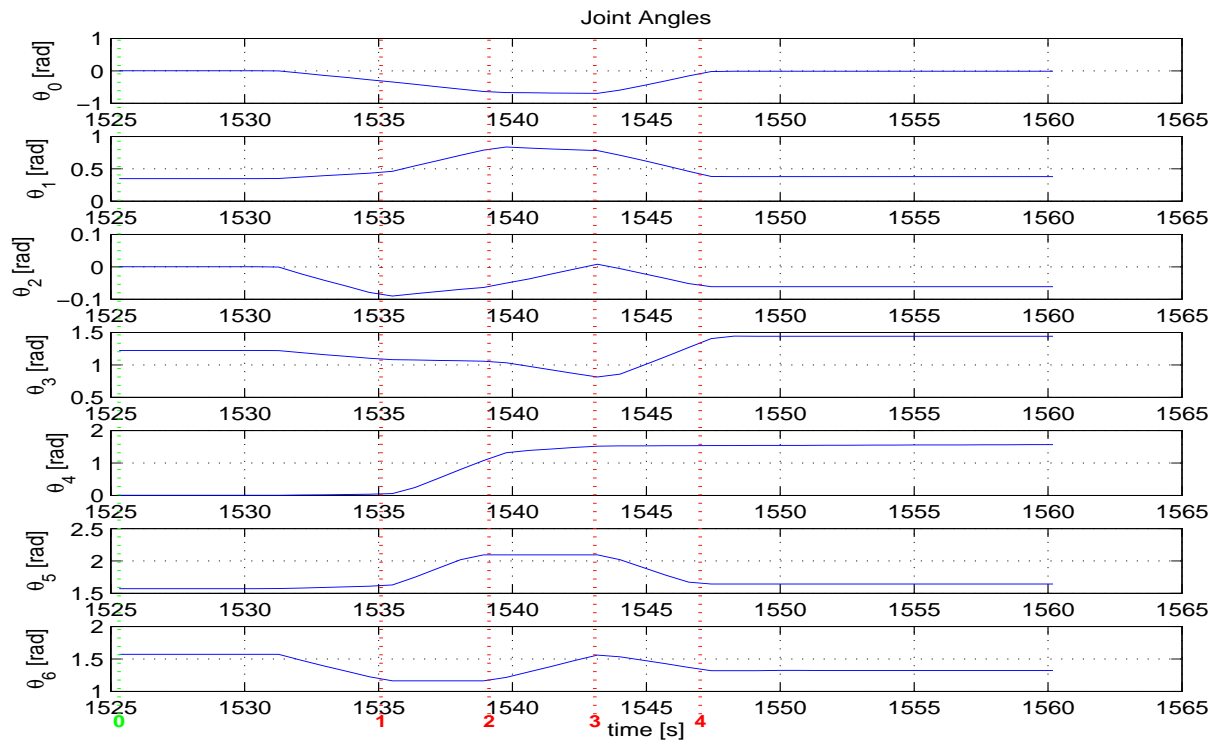


2010_11_05_12_37_44log.txt

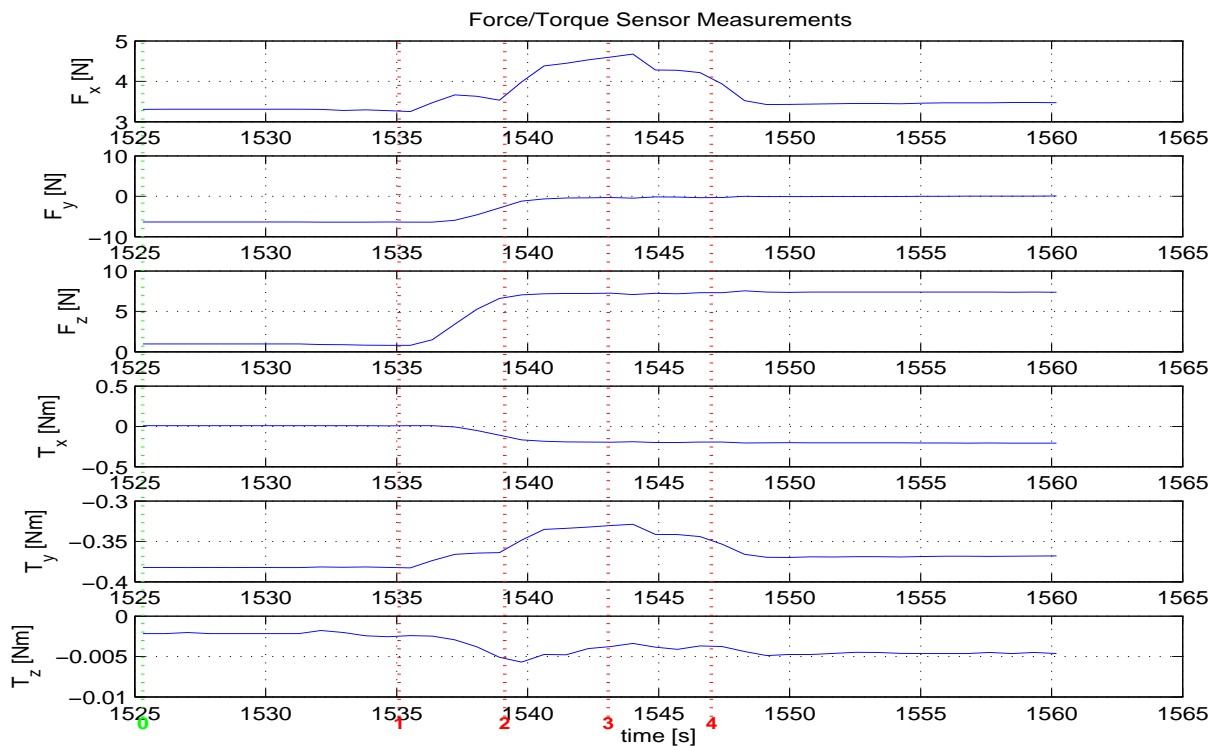


2010_11_05_12_37_44log.txt

Abbildung 124: Gelenkwinkel und Endeffektor Kraftsensor Messungen von Experiment RE44.

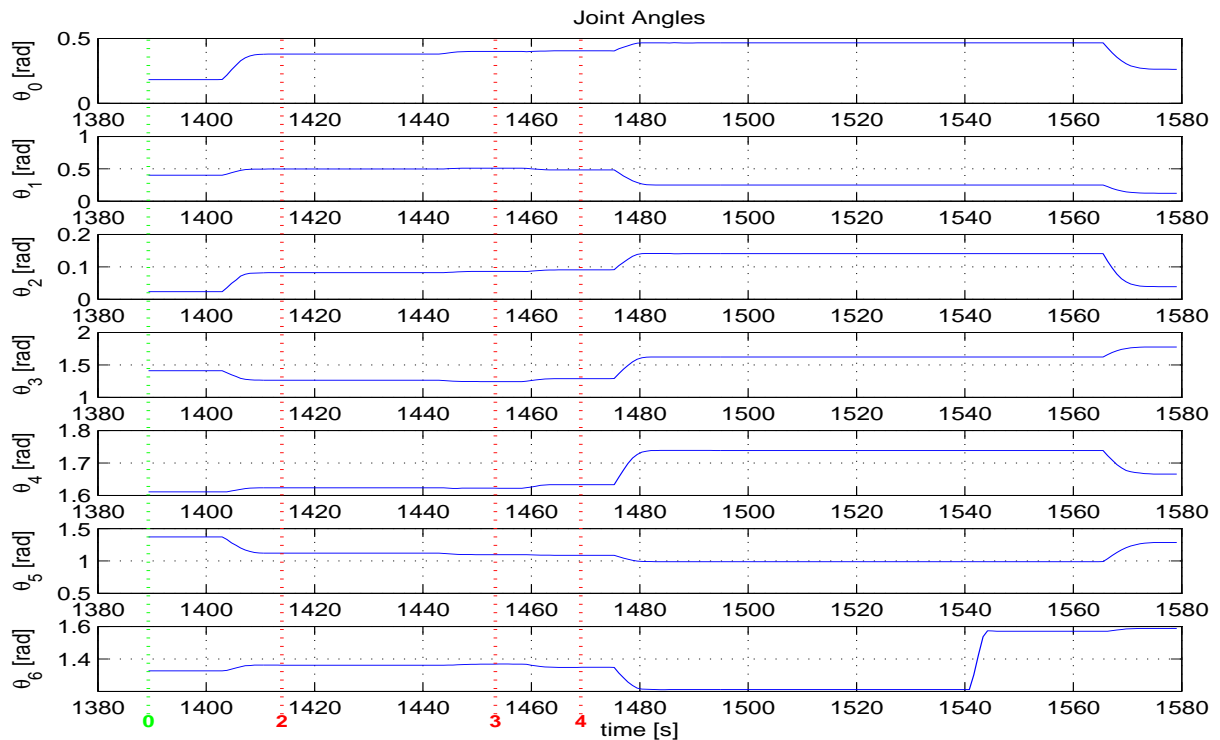


2011_01_06_14_25_24log.txt

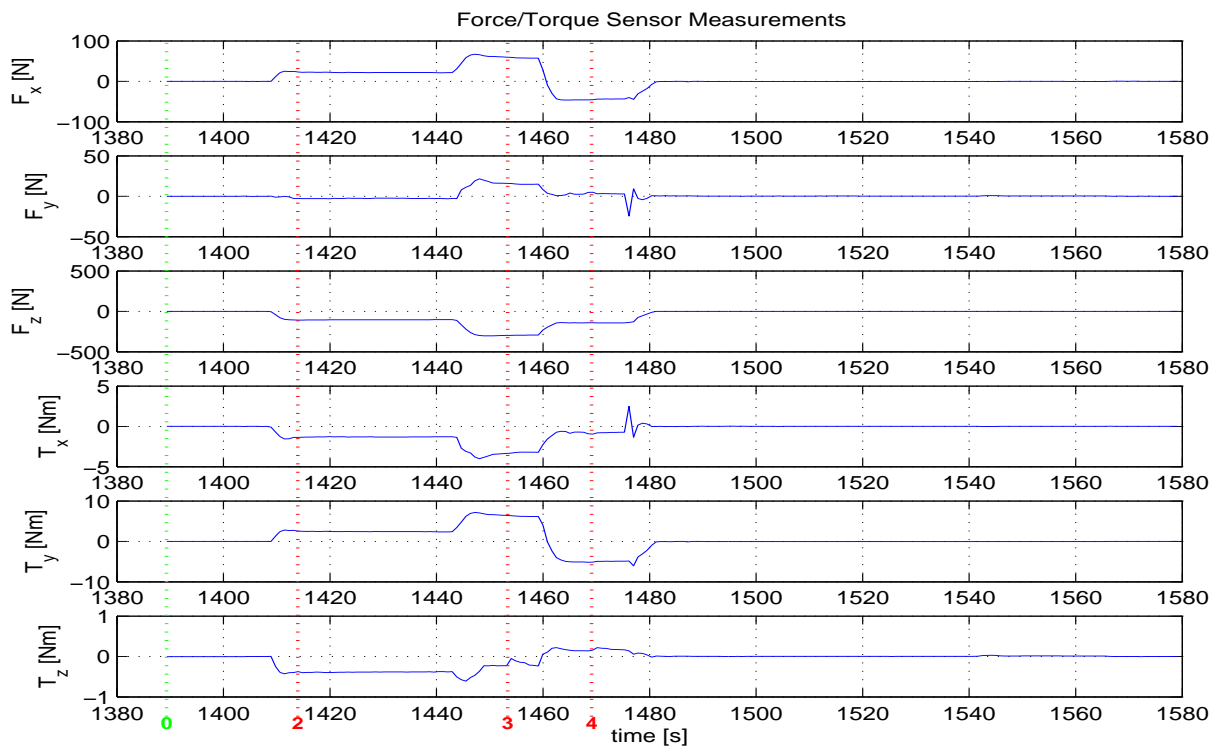


2011_01_06_14_25_24log.txt

Abbildung 125: Gelenkwinkel und Endeffektor Kraftsensor Messungen von Experiment RE24.

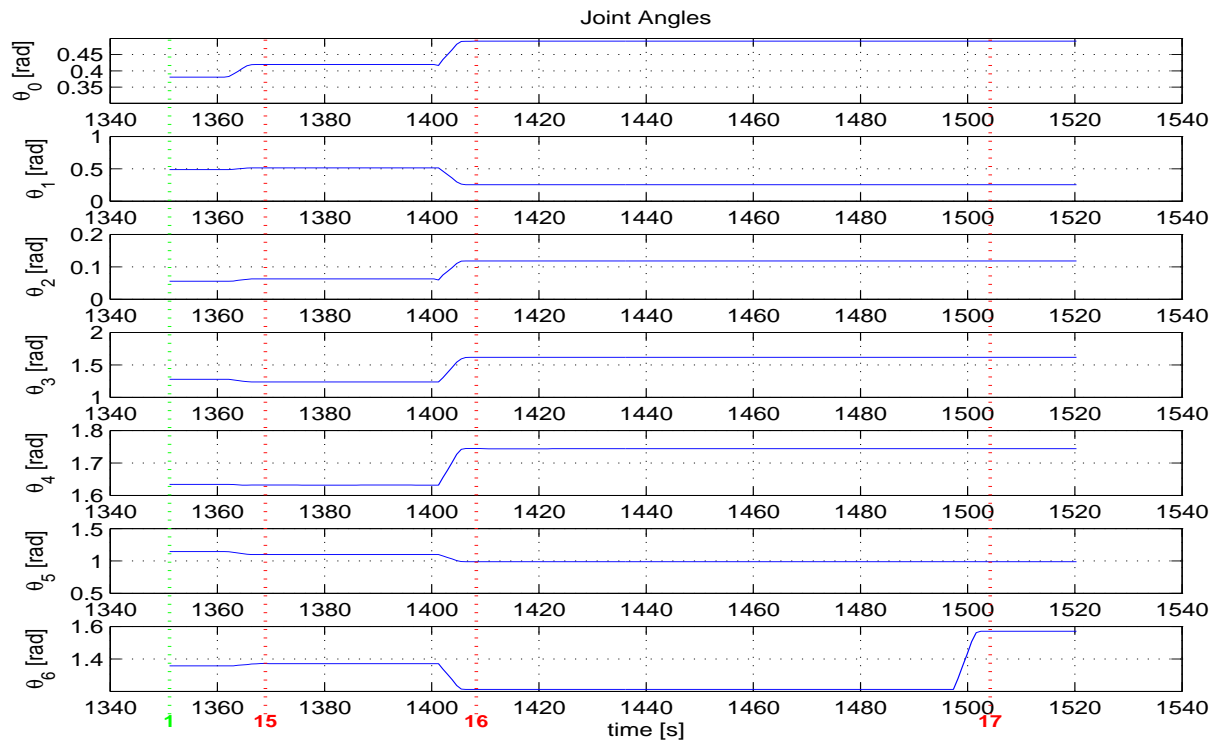


2011_01_18_15_23_09log.txt

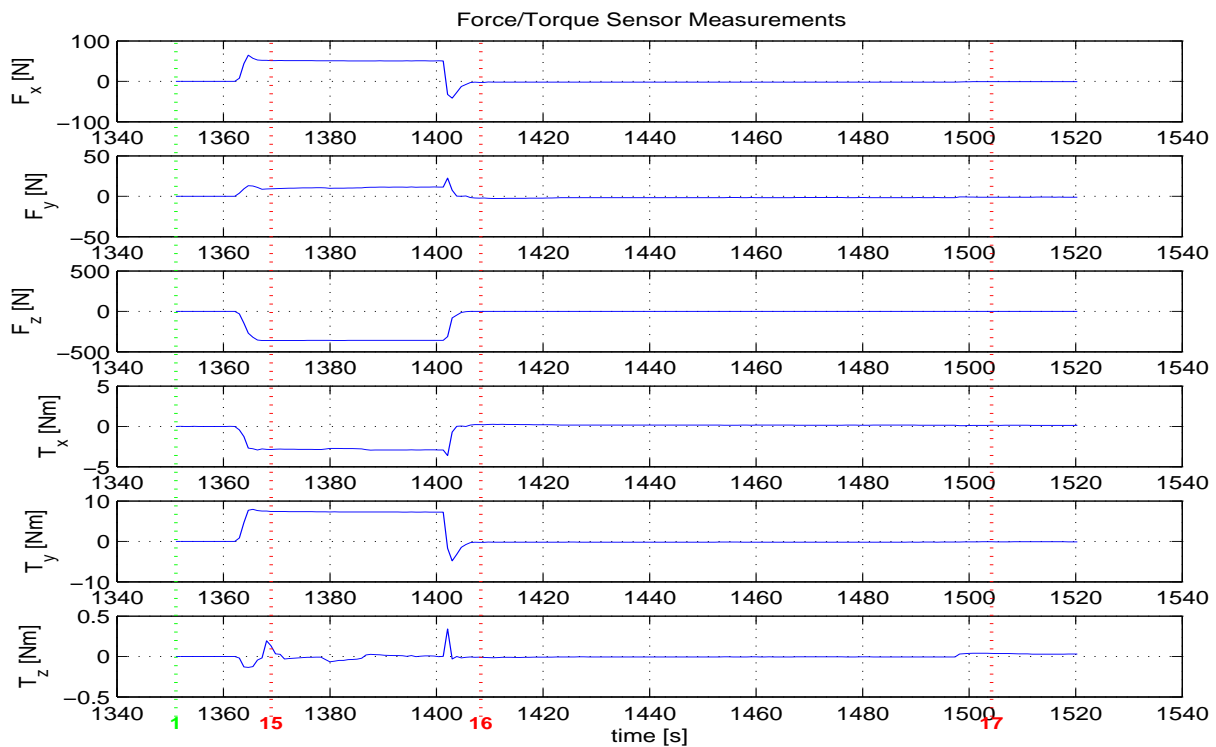


2011_01_18_15_23_09log.txt

Abbildung 126: Gelenkwinkel und Endeffektor Kraftsensor Messungen von Experiment RE09.



2011_01_20_12_22_30log.txt



2011_01_20_12_22_30log.txt

Abbildung 127: Gelenkwinkel und Endeffektor Kraftsensor Messungen von Experiment RE30.

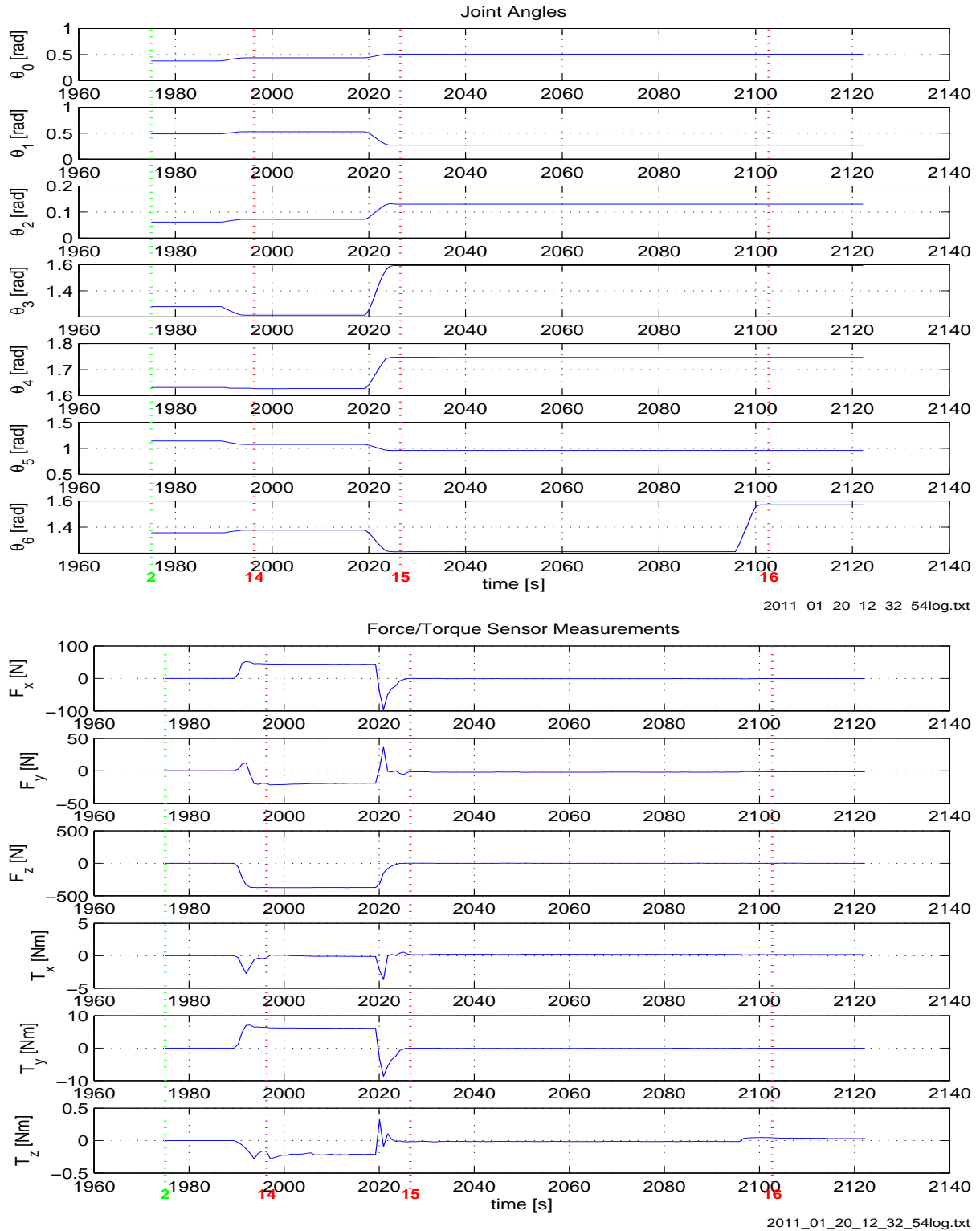
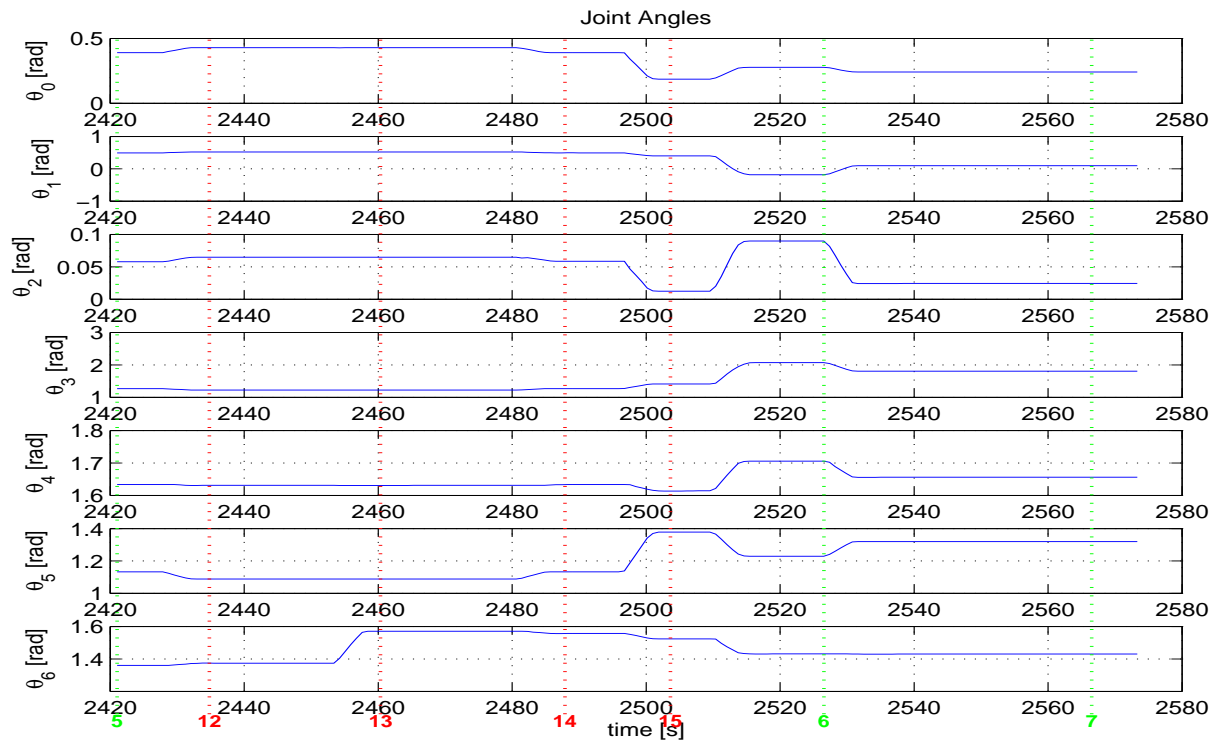
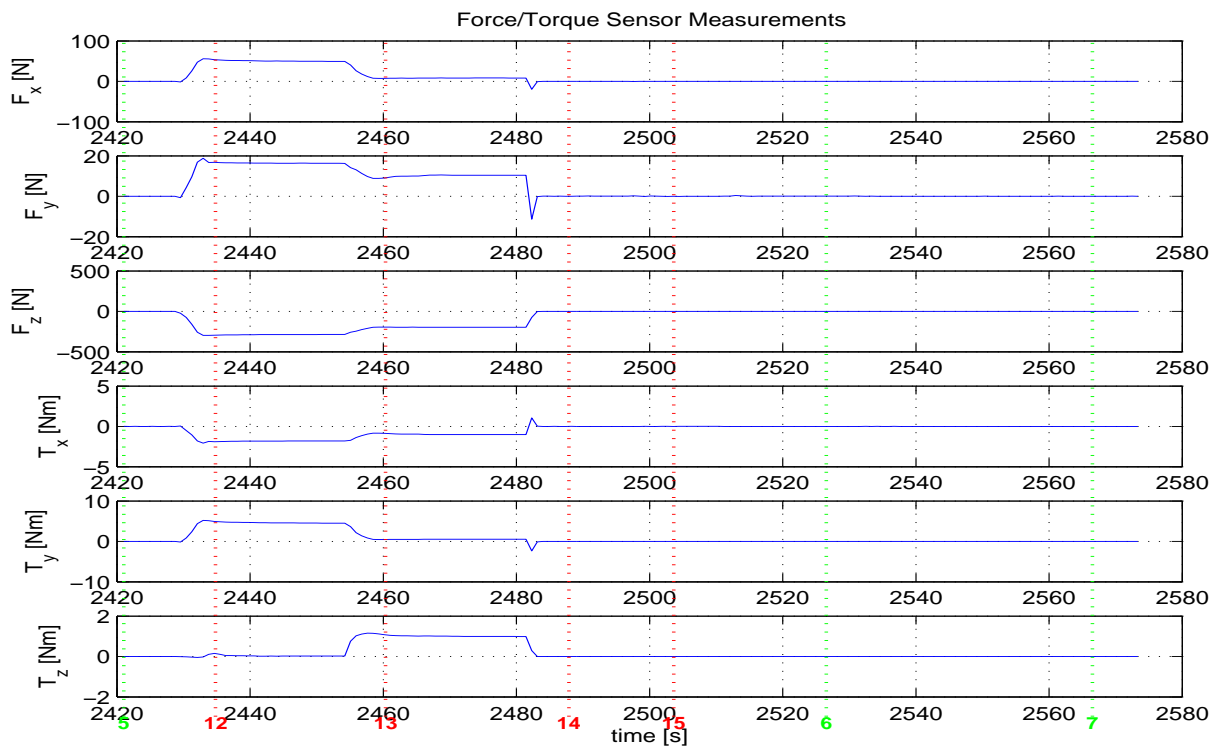


Abbildung 128: Gelenkwinkel und Endeffektor Kraftsensor Messungen von Experiment RE54.



2011_01_20_14_40_20log.txt



2011_01_20_14_40_20log.txt

Abbildung 129: Gelenkwinkel und Endeffektor Kraftsensor Messungen von Experiment RE20.



Experimente zur Bodenmechanik mit dem SpaceClimber Fuß In den Bodenexperimenten verwenden wir insbesondere auch einen original SpaceClimber Fuß. Der Fuß mitsamt der Montagevorrichtung ist in Abbildung 134 zu sehen. Die Massen der Teile sind in Tabelle 15 aufgelistet.

Fuß & Adapter	145,6 g
Sensorplatte	64,7 g
Schrauben	10,9 g
Gesamt	221,1 g

Tabelle 15: Massen des SpaceClimber Fußes und relevanter Adapter. Alle Angaben haben einen Fehler von $\pm 0,1$ g.

Es wurden drei verschiedene Arten von Experimenten durchgeführt. In der ersten Versuchsreihe wird der Fuß vom Roboterarm direkt über dem Boden im rechten Winkel zur Oberfläche positioniert. Der Arm wird dann so angesteuert, dass sich der Fuß zunächst 30 mm senkrecht nach unten und dann wieder nach oben bewegt. Dieser Vorgang wird dreimal wiederholt, wobei sich der Fuß stets mit einer konstanten Geschwindigkeit von 7,5 mm/s bewegt. In Abbildung 135 wird der Versuchsablauf skizziert.

In der zweiten Versuchsreihe (in Abbildung 136 skizziert) wird der Arm wieder senkrecht zur Oberfläche direkt über dem Boden positioniert. Anschließend drückt der Schunk-Arm den Fuß 10 mm senkrecht in den Boden. Nachdem der Fuß wieder zurück zur Oberfläche bewegt wurde, wird der Vorgang mit einer Tiefe von 20 mm und schließlich 30 mm wiederholt. Jede Bewegung wird innerhalb von vier Sekunden ausgeführt, so dass sich für den ersten, zweiten und dritten Bodenkontakt die Geschwindigkeiten entsprechend zu 2,5, 5,0 und 7,5 mm/s ergeben.

Wieder ausgehend von der Oberfläche wird der Fuß in der dritten Versuchsreihe 20 mm tief in den Boden gedrückt und in dieser Tiefe mit 12,5 mm/s 150 mm horizontal durch den Sand gezogen. Danach wird der Fuß wieder senkrecht aus dem Boden bewegt. Eine Skizze dieses Experiments findet sich in Abbildung 137.

In jeder Versuchsreihe wurden fünf Experimente durchgeführt. Bei der Durchführung wurden sowohl ein original SpaceClimber Fuß und der DFKI Roboter „MrSemProm“ als auch die speziell entwickelte Steuerungs- und Datenakquisitionsoftware eingesetzt.

In den beschriebenen Experimenten wurden unter anderem die sieben Gelenkwinkel, die kartesische Position und Orientierung des Endeffektors und die Kraft- und Drehmomentwerte des Sensors mit synchronisiertem Zeitstempel versehen und mit der entwickelten Datenakquisitionsoftware gespeichert.

Abbildung 138 bis 140 zeigen exemplarische Datensätze aus den drei durchgeführten Versuchsreihen.

Da die ersten Experimente zur Bodenmechanik stark schwankende Ergebnisse hervorgebracht hatten, mussten diese mehrfach wiederholt werden um eine Abschätzung des tatsächlichen Bodenverhaltens zu ermöglichen. Das Experiment für das vertikale Eindringen wurde erneut mit 15 Wiederholungen durchgeführt. Das Ergebnis ist in dem Graphen 141 dargestellt. Obwohl Wert auf ein gleichmäßiges Auflockern des Bodens nach jedem einzelnen Durchlauf gelegt wurde variieren die gemessenen Normalkräfte

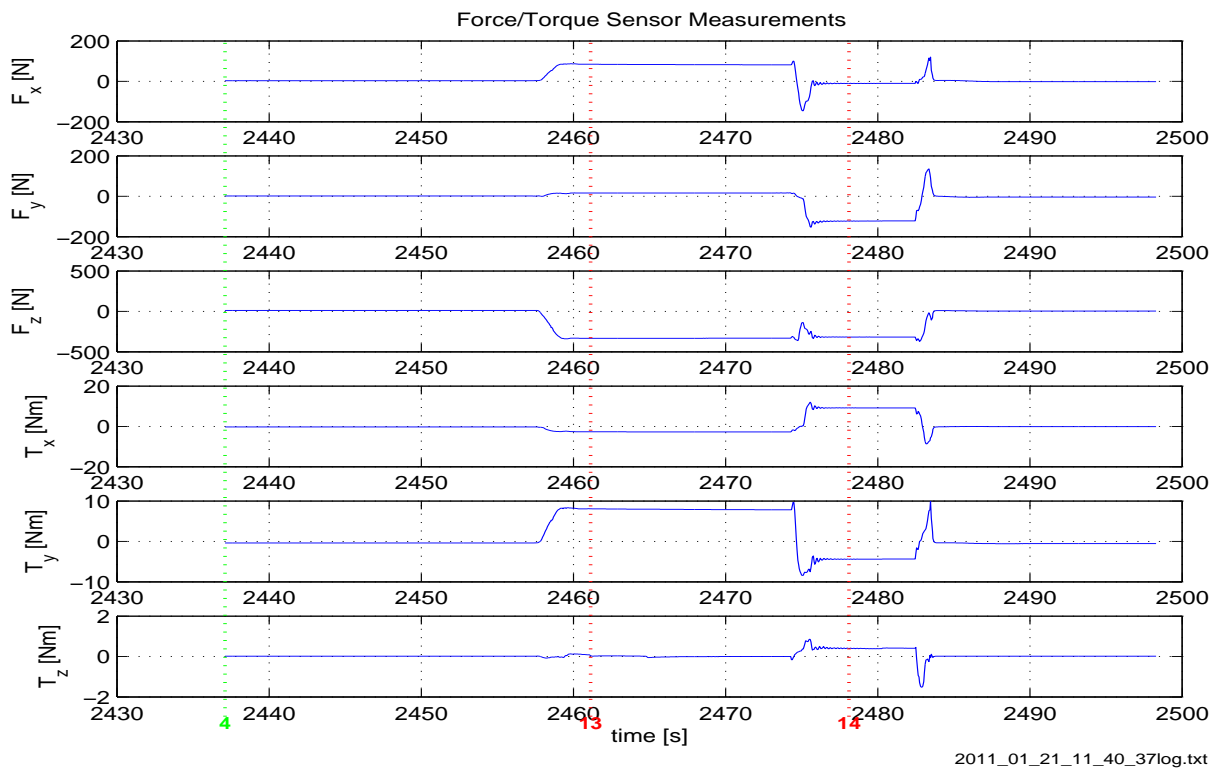
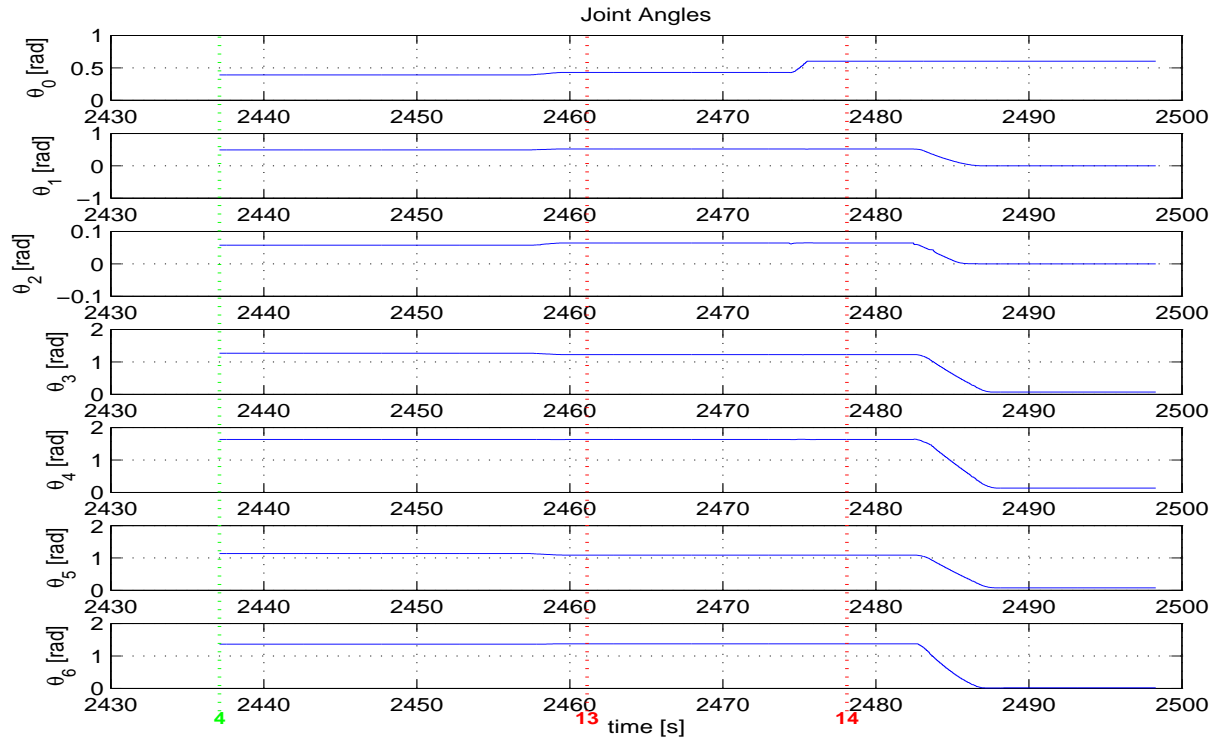


Abbildung 130: Gelenkwinkel und Endeffektor Kraftsensor Messungen von Experiment RE37.

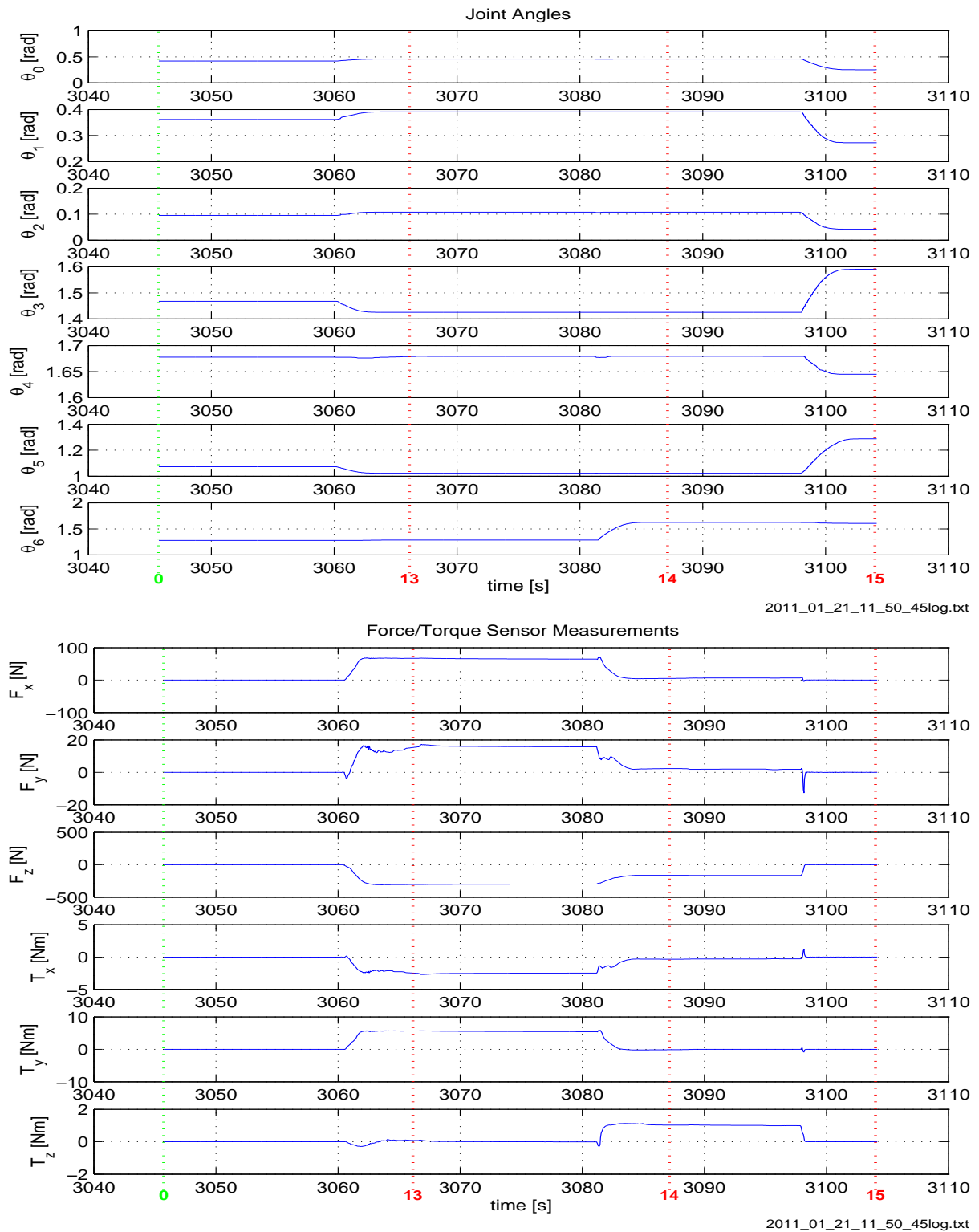
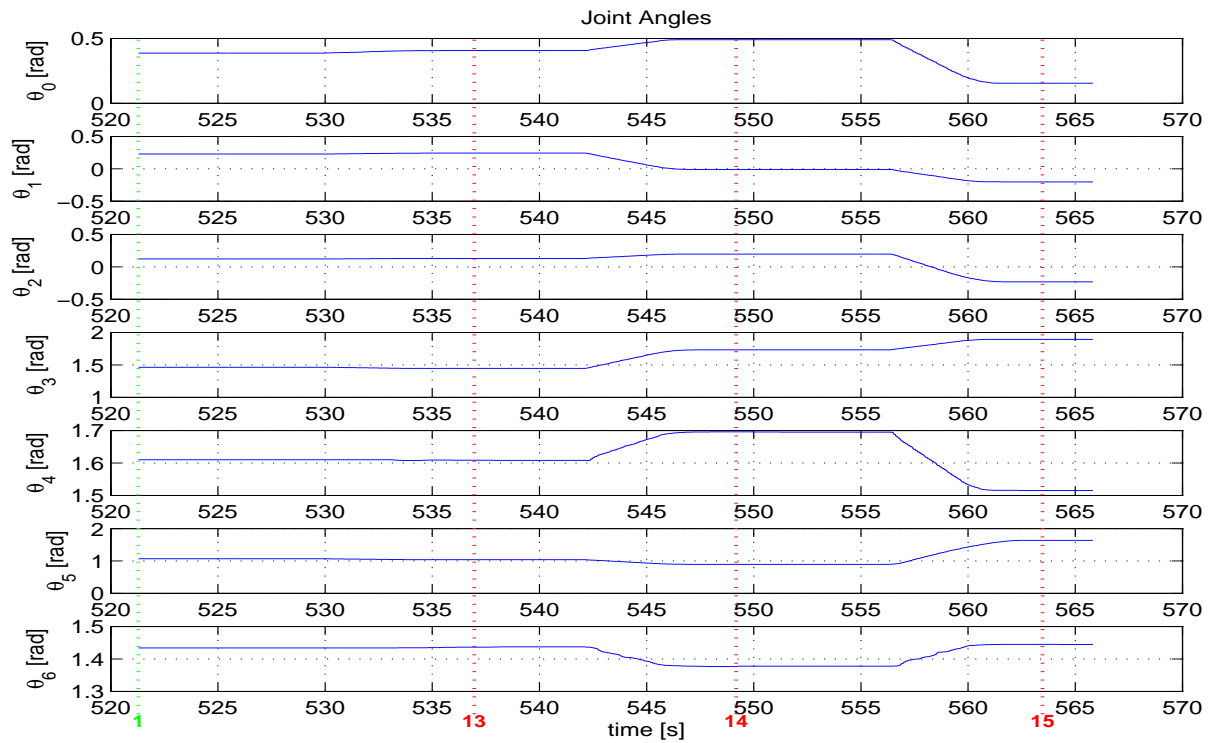
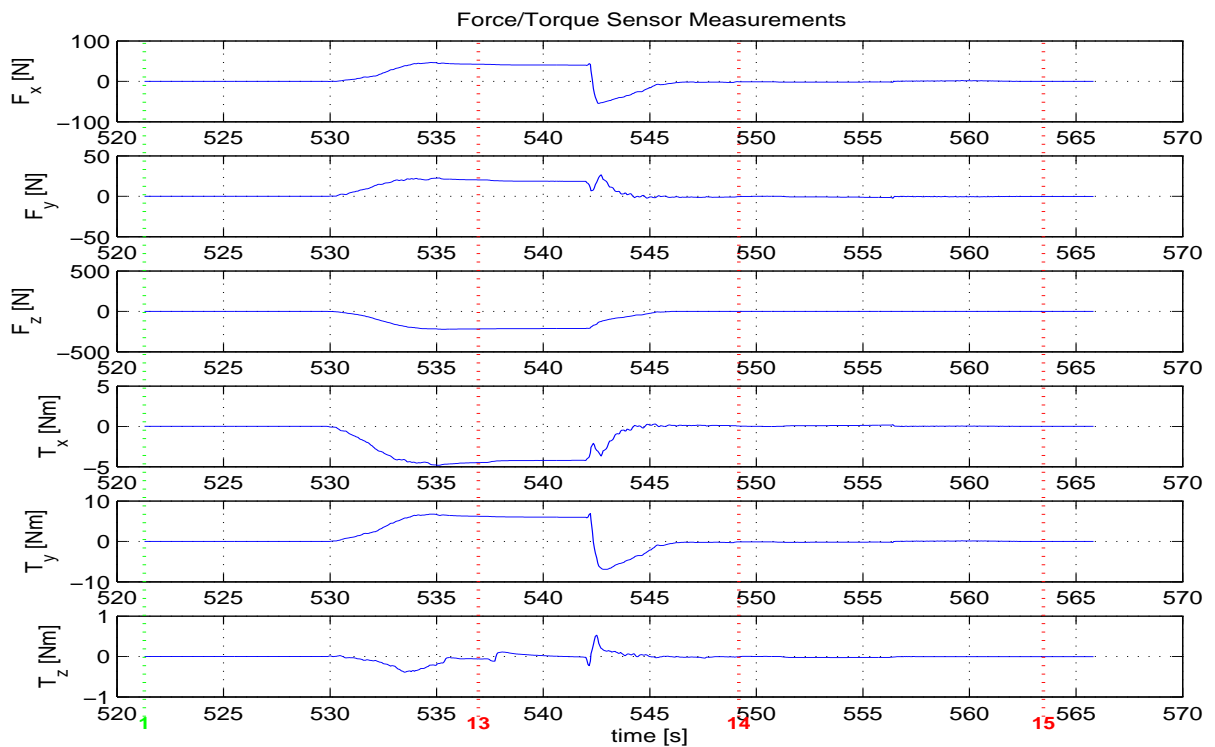


Abbildung 131: Gelenkwinkel und Endeffektor Kraftsensor Messungen von Experiment RE45.

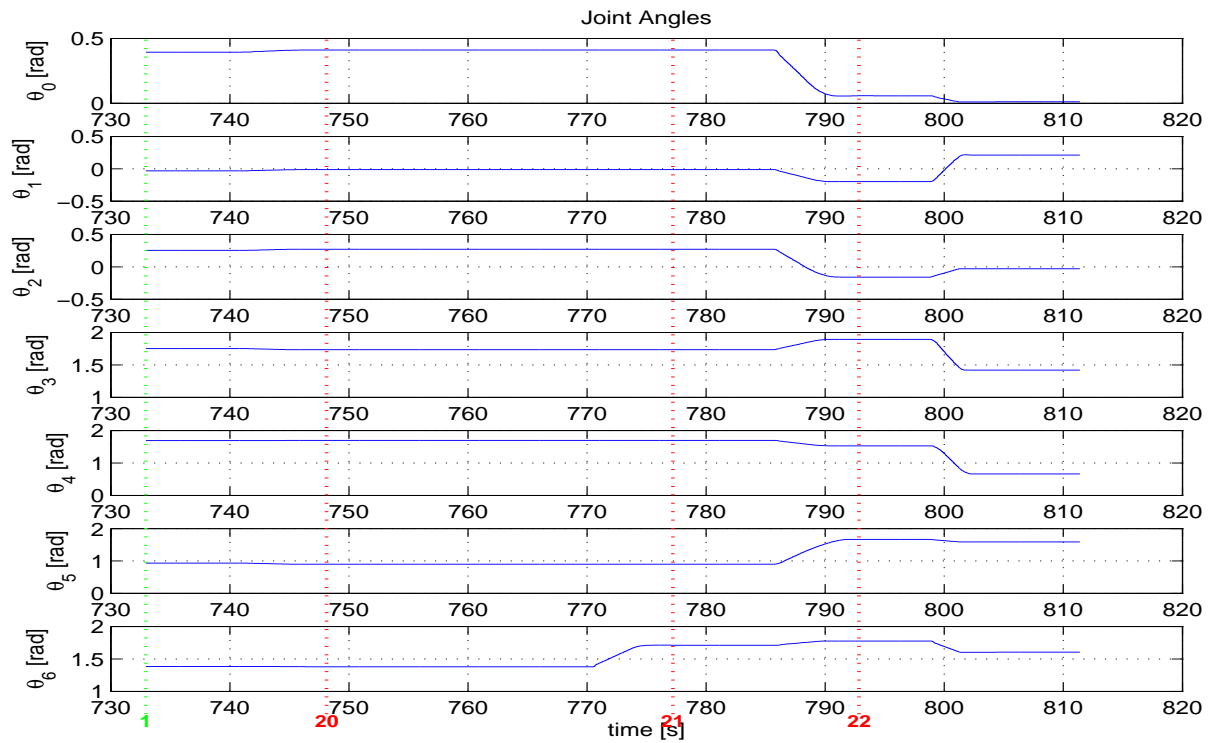


2011_01_21_13_08_41log.txt

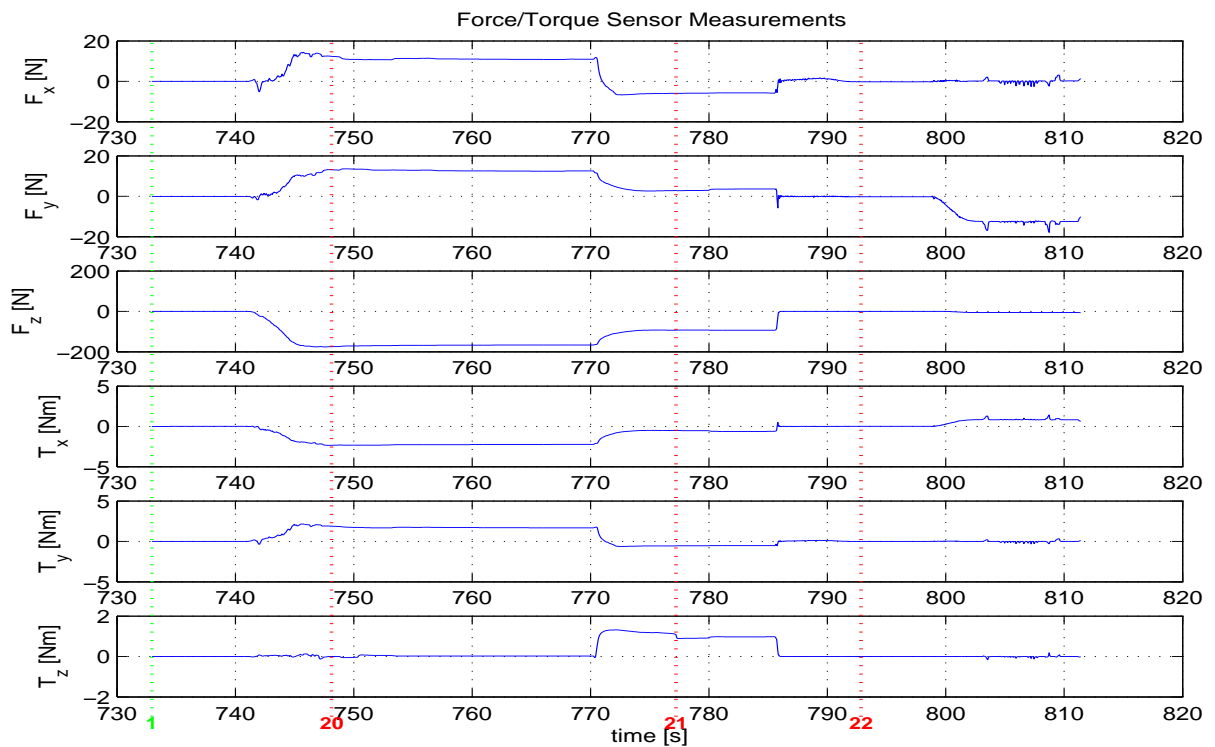


2011_01_21_13_08_41log.txt

Abbildung 132: Gelenkwinkel und Endeffektor Kraftsensor Messungen von Experiment RE41.



2011_01_21_13_12_12log.txt



2011_01_21_13_12_12log.txt

Abbildung 133: Gelenkwinkel und Endeffektor Kraftsensor Messungen von Experiment RE12.

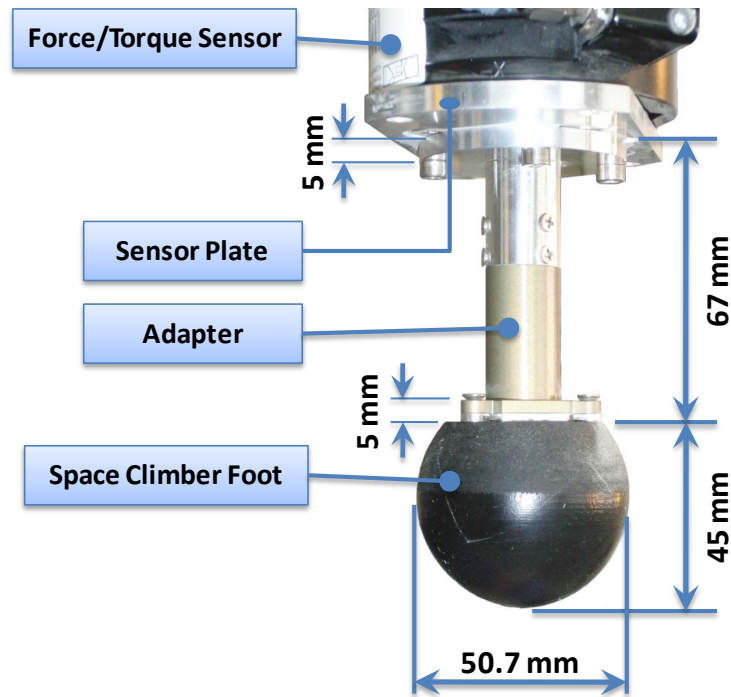


Abbildung 134: Detailaufnahme der Anbringung des SpaceClimber Fußes an den Kraft-Drehmoment-Sensor, wie sie für die Bodenmechanik Referenzexperimente verwendet wurde.

sehr stark. Die Variation hat eine Spannweite von ca. 40-100 N bei einer Eindringtiefe von 27 mm. Dies führt zu der Annahme, dass schon kleine Unterschiede bei der Verdichtung des Bodens relative starke Unterschiede bei den Gegenkräften bewirken. Mit den 15 Ergebnissen konnte grob ein Varianzbereich ermittelt werden der das Entwickeln und Optimieren der Bodenmechanik in der Simulation erlaubt.

Es wurden weitere spezialisiertere Experimente im Zuge von „AP5300 Anpassung der Simulationsmodelle an reale Ergebnisse“ durchgeführt.

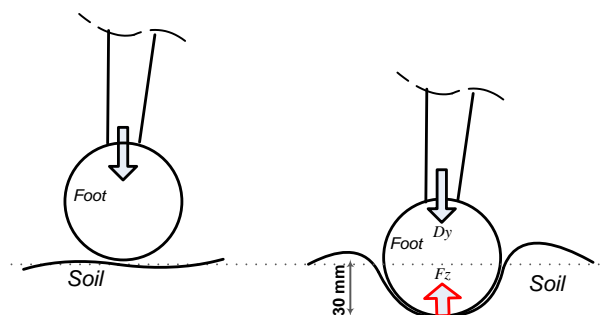


Abbildung 135: Skizze des Versuchablaufs der ersten Versuchsreihe. Der Fuß wird dreimal von der Oberfläche 30 mm nach unten bewegt.

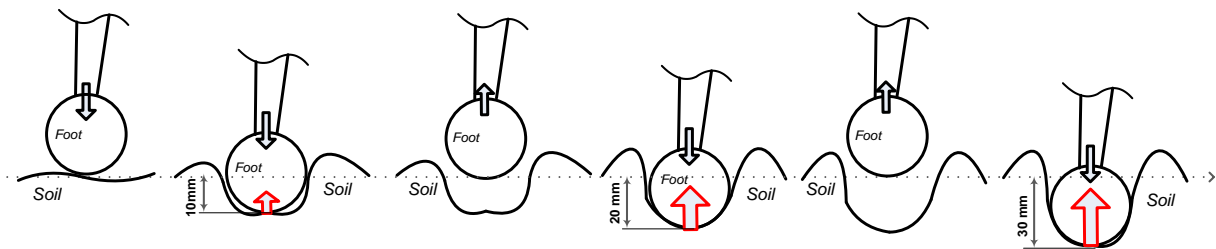


Abbildung 136: Skizze des Versuchablaufs der zweiten Versuchsreihe. Der Fuß wird zunächst 10 mm, dann 20 mm und schließlich 30 mm nach unten bewegt. Zwischen diesen Phasen wird er zurück zur Oberfläche bewegt.

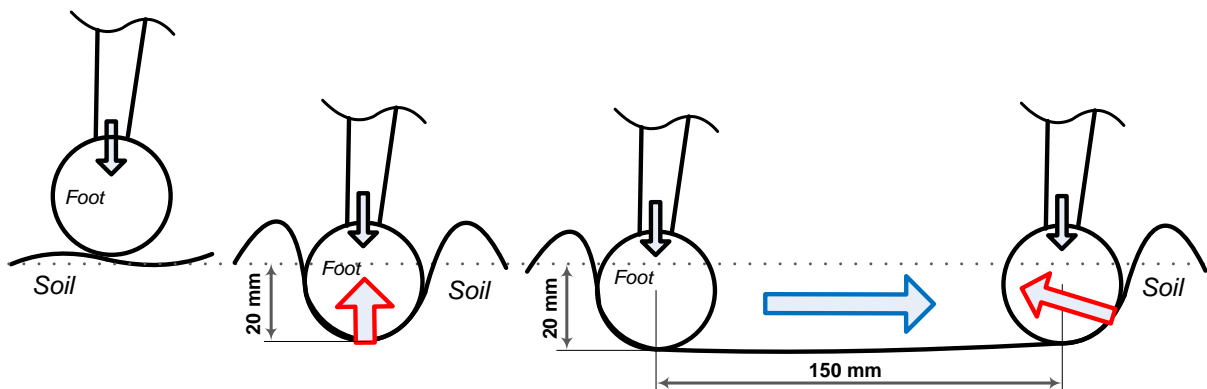


Abbildung 137: Skizze des Versuchablaufs der dritten Versuchsreihe. Dabei wird der Fuß mit einer 20 mm Eindringtiefe 150 mm horizontal durch den Sand gezogen.

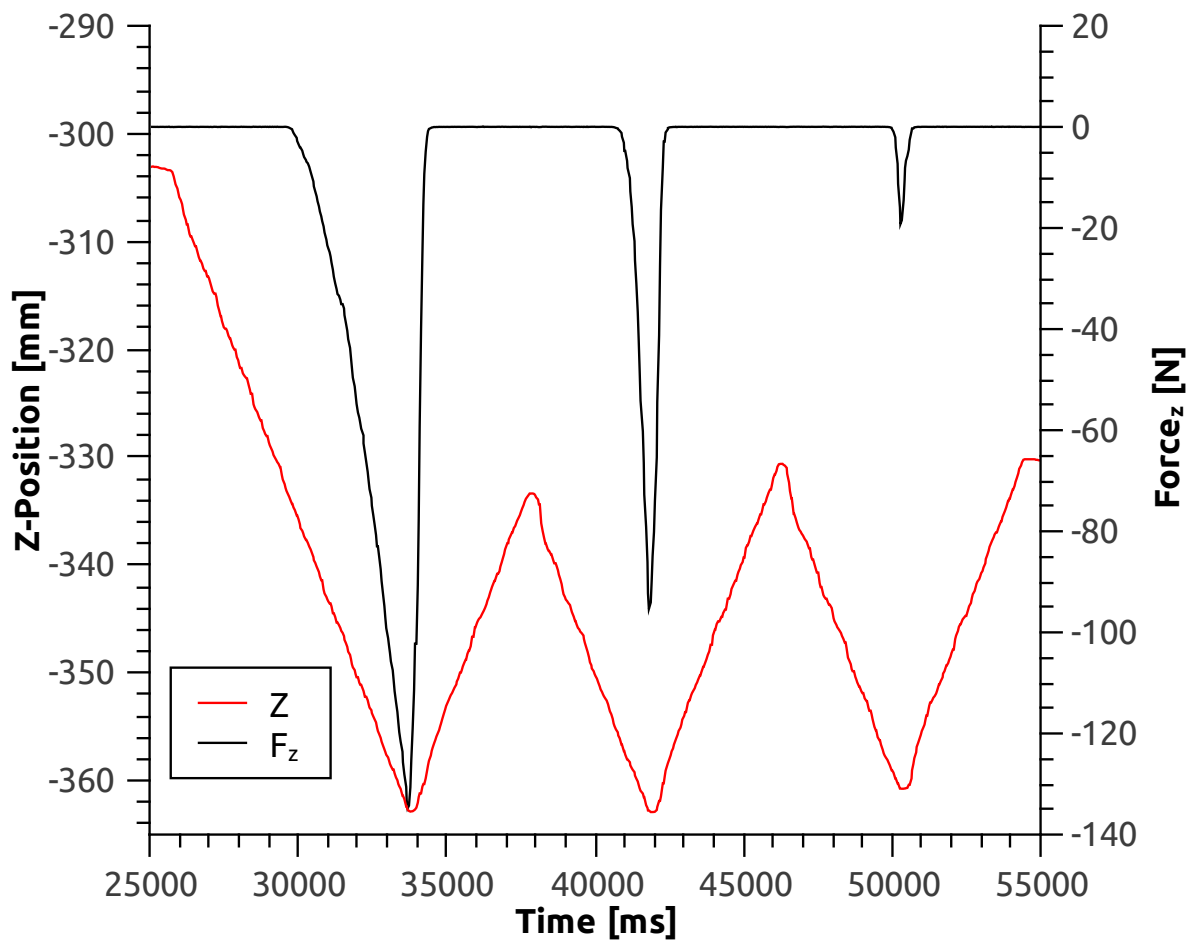


Abbildung 138: Exemplarische Daten aus der ersten Versuchsreihe.

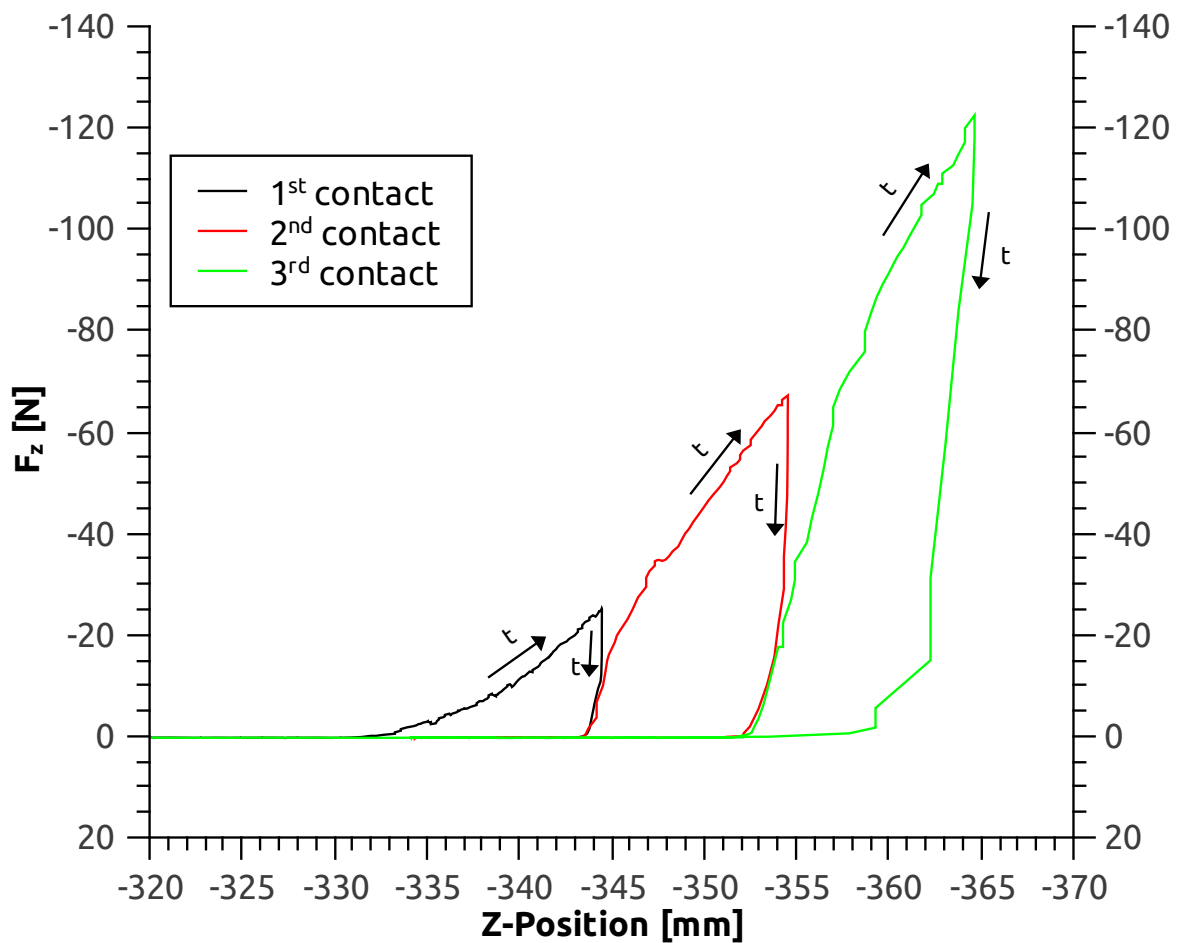


Abbildung 139: Exemplarische Daten aus der zweiten Versuchsreihe. Hier ist die vertikale Kraft gegen die Eindringtiefe des Fußes aufgetragen. Die drei Kontaktvorgänge mit unterschiedlicher Tiefe sind farblich voneinander abgesetzt. Die jeweils flachere Flanke entspricht dem Eindringen in den Boden wobei kontinuierlich eine Gegenkraft durch den Boden aufgebaut wird. Die steilen Flanken entsprechen dem Rückzug des Fußes. Der zeitliche Verlauf des Experiments ist durch die mit *t* markierten Pfeile angedeutet.

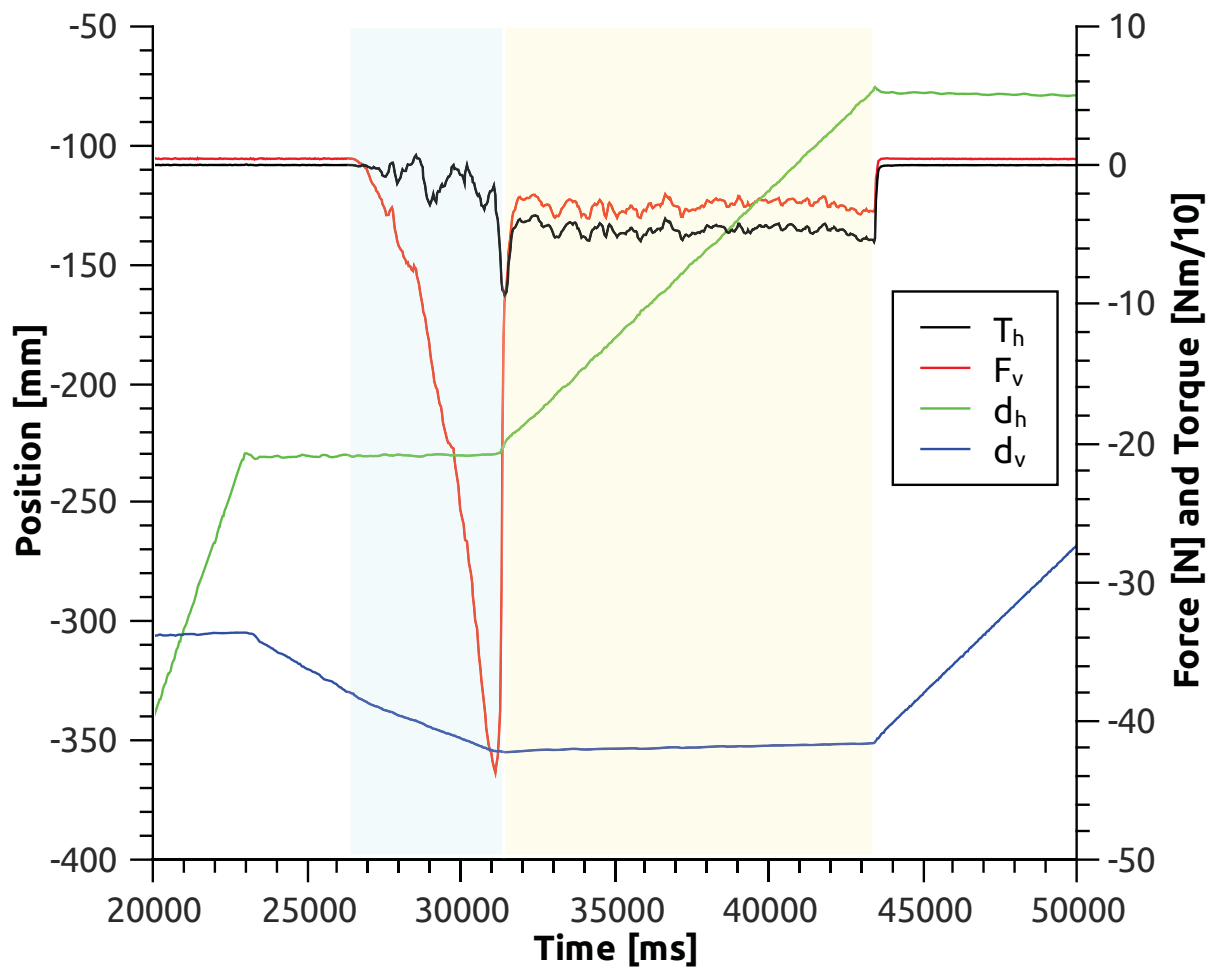


Abbildung 140: Exemplarische Daten aus der dritten Versuchsreihe. Der blau hinterlegte Bereich entspricht dem Eindringvorgang in den Boden (ab $t \approx 26,5$ s). Ab $t \approx 30,5$ s beginnt die horizontale Bewegung, die durch den gelb hinterlegten Bereich markiert ist. Dabei ist T_h das Drehmoment, dass durch die horizontale Kraft des Bodens verursacht wird und F_v die vertikale Gegenkraft des Bodens. d_h und d_v sind die in horizontaler bzw. vertikaler Richtung zurückgelegten Distanzen.

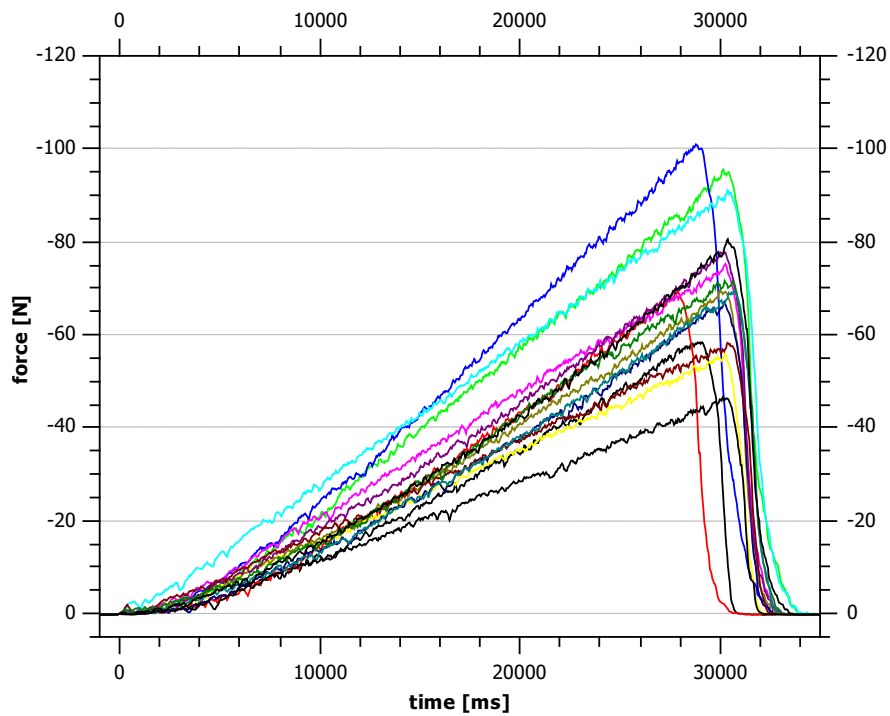


Abbildung 141: 15 Ergebnisse für das Eindringen in den Experimentboden.



Tabelle 16: Am Motorteststand (real und simuliert) durchgeführte Experimente

		PWM-Signal					
		10	50	100	150	200	250
Last	0 kg	x	x	x	x	x	x
	3.81 kg	x	x	x	x	x	x
	6.31 kg	x	x	x	x	x	x
	8.81 kg	x	x	x	x	x	x
	11.31 kg	x	x	x	x	x	x
	13.81 kg	x	x	x	x	x	x
	18.81 kg		x	x	x	x	x
	23.81 kg		x	x	x	x	x
	33.81 kg		x	x	x	x	x

3.1.44 AP5420: Durchführung im virtuellen Testbed

Motorteststand Die in AP5410 am Motorteststand durchgeführten Experimente (vgl. Tab. 16) wurden auch in der im Rahmen von AP4420 aufgebauten Simulationsumgebung durchgeführt. Dabei konnte wie erwartet ein nahezu linearer Zusammenhang zwischen Strom und Geschwindigkeit beobachtet werden. Exemplarisch ist dies in Abbildung 142 für eine Last von 23,81 kg dargestellt.

Die Einstellung der Motormodellparameter um eine genaue Übereinstimmung des Stromverbrauchs unter verschiedenen Lasten zu gewährleisten erfolgt in AP5300.

Bodenmechanik Die Referenzexperimente zu den lateralen Kräften der Bodenmechanik (siehe AP5300) wurden ebenfalls in VEROSIM[®] durchgeführt. Die bereits entwickelten Plugins wie der LogfileReader und das Optimierungs- und Automatisierungs-Plugin konnten auch hierbei erfolgreich eingesetzt werden. Der Verlauf eines exemplarischen Experiments ist in Abbildung 143 dargestellt.

Szenario zur Ableitung eines abstrakten Motormodells aus virtuellen Referenzexperimenten mit dem detaillierten Motormodell In den früheren Arbeiten (AP2200, AP3200 und AP4340) wurde eine effektive Tool-Chain zwischen präziser Simulation und Echtzeit-Simulation definiert und implementiert, um den Realismus der Simulation in VEROSIM[®] zu verbessern. Das abstrakte Motormodell wurde bisher so entwickelt, dass eine System-Identifikation durch ein Referenzexperiment mit dem echten SpaceClimber-Antrieb durchgeführt wurde und diese dann in der Echtzeitsimulation VEROSIM[®] eingesetzt und getestet wurde.

In einer Designphase, in der nur primäre Motorparameter aus den Anforderungen bekannt sind und keine echten Motor-Experimente vorliegen, soll es ebenfalls möglich sein, ein abstraktes Modell für die Echtzeitsimulation VEROSIM[®] zu entwickeln. Eine Lösung für ein solches Szenario wurde in diesem Quartal entwickelt. Die entsprechenden Motor-Parameter aus den Anforderungen des SpaceClimber-Antriebs wurden direkt für das detaillierte präzise Motormodell basierend auf Adams/Matlab verwendet.

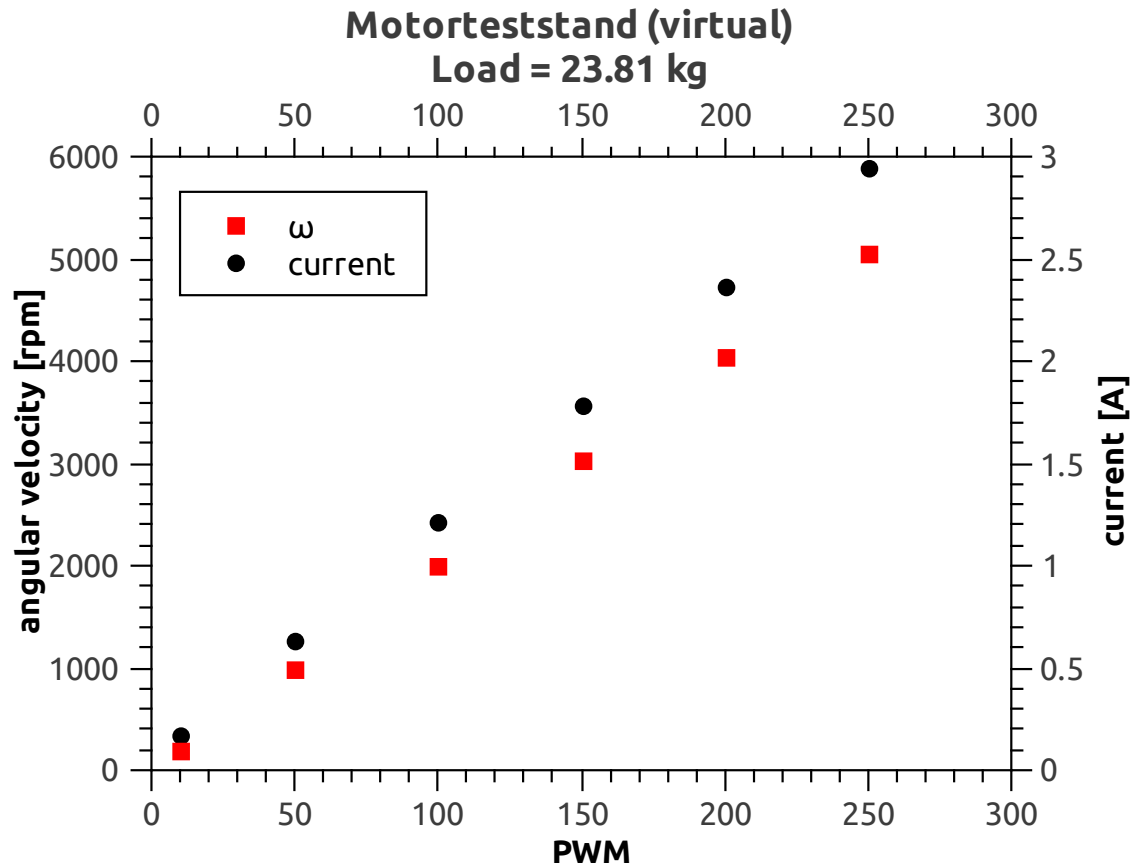


Abbildung 142: Daten aus dem virtuellen Motorteststand. Eine Last von 23,81 kg wird mit verschiedenen PWM-Signalen über eine Winde angehoben.

Mit diesem Modell wurden dann virtuelle Referenzexperimente durchgeführt. Ein abstraktes Modell wurde auf der Basis der Ergebnisse dieser virtuellen Experimente entwickelt, um es in der Echtzeitsimulation VEROSIM® einzusetzen.

Part	max. Mass
Actuator	0.45kg
Lower leg	0.3kg
Upper leg	0.15kg
Body	11.5kg
Total mass	25kg

Tabelle 17: Maximale Massen der Bestandteile des SpaceClimber Roboters als Anforderung

Tabelle 17 zeigt die maximalen Massen der Bestandteile des SpaceClimbers. Die Morphologie des SpaceClimber Roboters wurde mit diesen Massen in ein präzises Modell auf der Basis von Adams/Matlab überführt und die Kräfte, die beim Laufen mit einem Laufmuster an den Gelenken auftreten und zur Bewegung von den Aktuatoren aufgebracht werden müssen, wurden simuliert. Die Simulationsergebnisse in Abbildung 144 zeigen, dass die maximal erforderliche Gelenkkraft beim Laufen etwa 14 Nm beträgt.

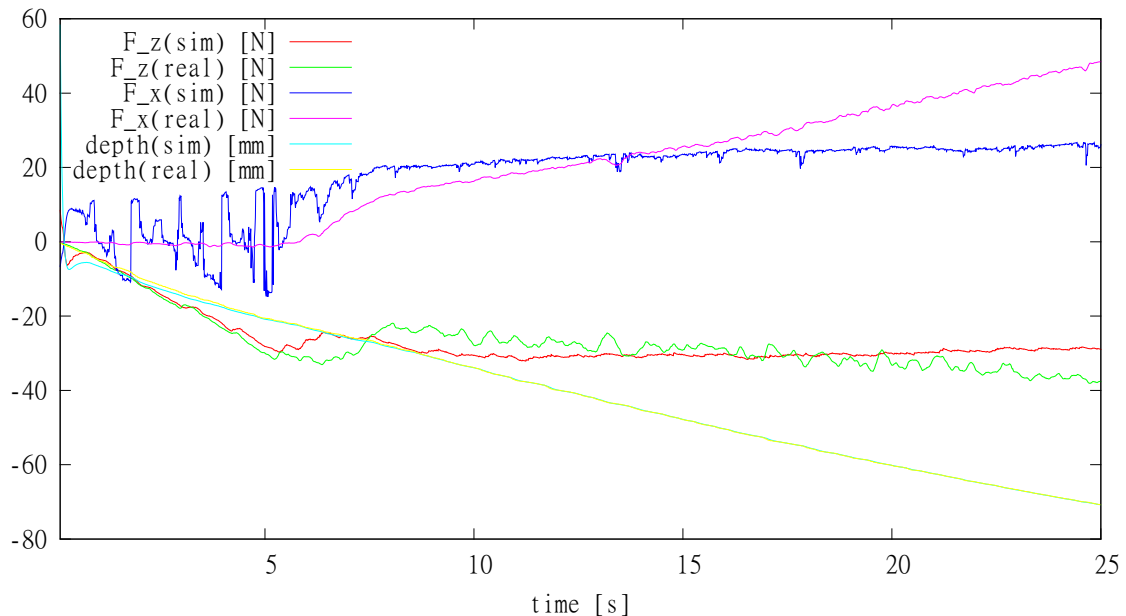


Abbildung 143: Durchführung eines virtuellen Experiments zu lateralen Kräften der Bodenmechanik. Gezeigt werden die Normalkräfte F_z , die lateralen Kräfte F_x und die Eindringtiefe $depth$. Zum Vergleich sind neben den simulierten Daten ebenfalls die Daten des realen Experiments dargestellt.

Auf Basis einer solchen Simulationsanalyse der Gelenkkkräfte kann ein realer auf dem Markt erhältlicher Motor gewählt werden. Die Tabelle 18 zeigt den Brushless Motor ILM 50×8 , der für SpaceClimber Antrieb ausgewählt wurde. Die angegebenen Drehmomente sind vor dem 100:1 Getriebe gemessen und reichen daher aus, um die erforderlichen 14 Nm zu erzeugen.

Gearbox	100 : 1
Rated Power	140 [W]
Rated Torque	0.28 [Nm]
Peak Torque	0.9 [Nm]
Rated Speed	5000 [1/min]
Torque Constant	0.075 [Nm/A]
Motor Constant	0.091 [Nm/ \sqrt{W}]
Terminal resistance	1030 [m Ω]
Terminal Inductance	1210 [μH]
Rotor Inertia	0.049 [kgcm ²]
Poles	20

Tabelle 18: Parameter des Brushless Motors ILM 50×8

Die Motorparameter in Tabelle 18 wurden im detaillierten Motormodell verwendet. Motorstrom und Geschwindigkeit wurden im präzisen am Motormodell in Adams/Matlab simuliert, mit verschiedenen den Belastungskräften und PWM-Signalen.

Abbildung 145 und 146 zeigen die Darstellungen der Simulationsergebnisse zu Geschwindigkeit und Motorstrom mit verschiedenen den Belastungskräften und PWM-

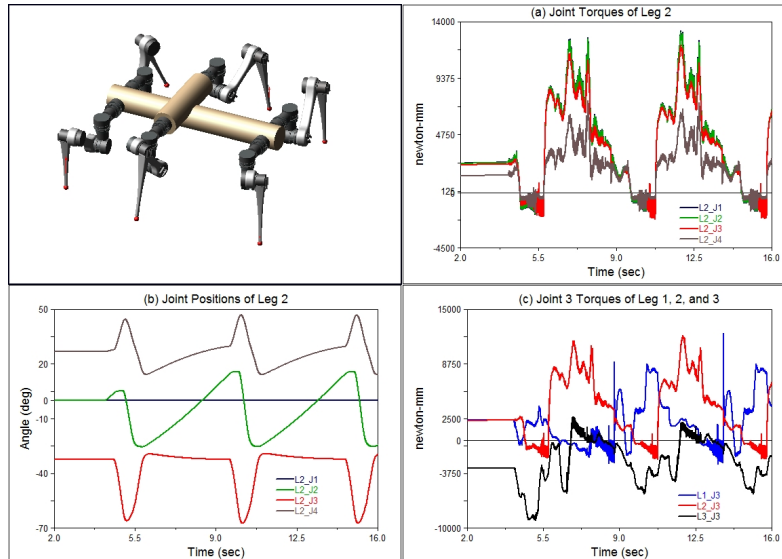


Abbildung 144: Analyse der nötigen Kräfte an den Gelenken beim Laufen

Signalen. Auf der Basis einer Lookup-Tabelle mit Werten aus den Simulationsergebnissen des präzisen Modells wurde ein abstraktes Motormodell entwickelt.

Durchführung der virtuellen Referenzexperimente zu den Sensoren Nachfolgend wird die Durchführung der einzelnen Referenzexperimente der Sensoren in der Simulation erläutert.

Force-Torque-Sensor

In der Abbildung 147 ist ein Durchgang des Experiments zum Force-Torque-Sensor zu sehen. In dem Beispiel wurde ein Gewicht von 2,544 Kg an einem 105 mm langen Hebel befestigt. Es wurde also die Position der Kugel, welche das Gewicht repräsentiert angepasst und ihr Gewicht entsprechend den Gewichten aus dem realen Experiment angepasst. Die Gewichte in dem simulierten Experiment entsprechen den Gewichten aus dem realen Experiment und sind: unbelastet, 2,544 und 5,036 Kg. Zudem wurde der Abstand des Gewichtes zur Messdose verändert. Hier beträgt der Abstand 105 und 205 mm. Im unteren Teil des Screenshots ist ein Plot zu sehen in dem sowohl die geloggtten Daten aus dem realen Experiment als auch die Werte aus der Simulation zu sehen sind. Es wurde ein Gauß-Rauschen auf die Werte des Sensors gelegt. Die Werte für dieses Rauschen wurden aus der Standardabweichung der realen Daten abgeleitet.

Joint-Position-Sensor

Bei diesem Experiment wurde das bei dem realen Experiment geloggte PWM-Signal als Input für das Motormodell genutzt. Die Abbildung 148 zeigt einen Screenshot

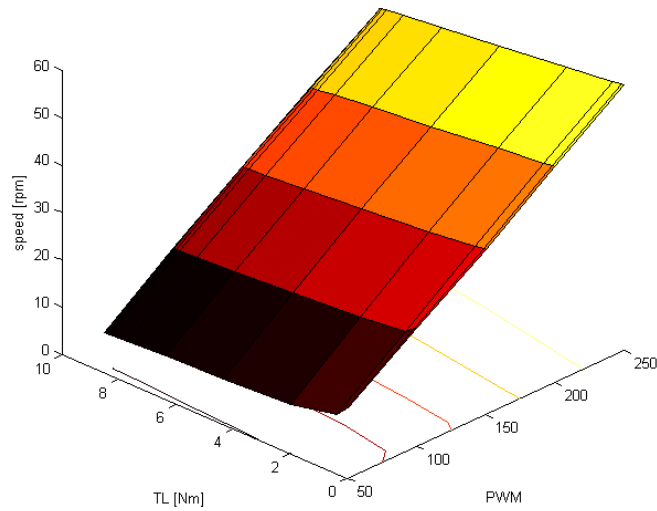


Abbildung 145: Geschwindigkeits-Werte im virtuellen Experiment

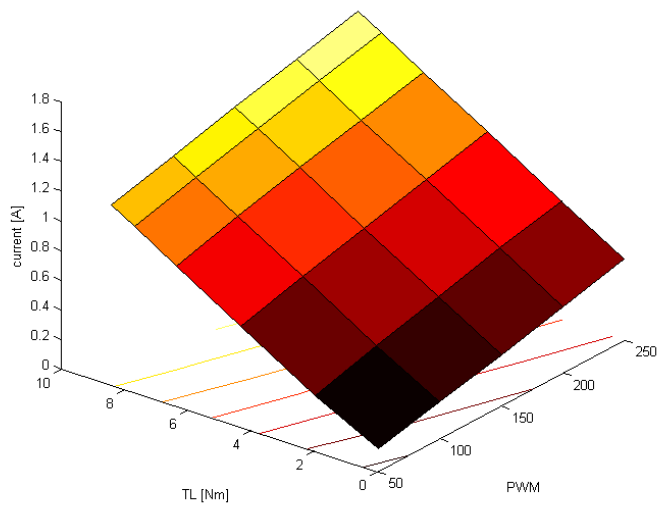


Abbildung 146: Motorstrom-Werte im virtuellen Experiment

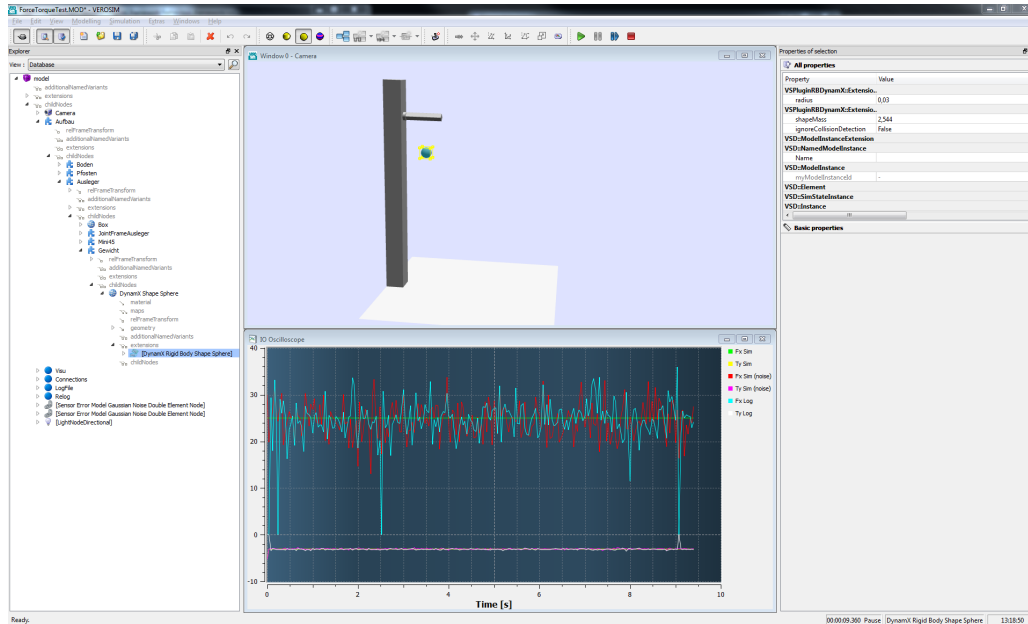


Abbildung 147: Screenshot von der Durchführung des Experiments für den Force-Torque-Sensor

des Experiments. In dem Plot ist sowohl der real gemessene inkrementelle Positions-Sensor zu sehen, wie auch der reale absolute Positions-Sensor. Zudem ist auch der virtuelle Positions-Sensor geplottet. Dies wurde für die drei vorhandenen Log-Dateien durchgeführt. Es wurde kein Rauschen auf den Sensor gelegt, da der inkrementelle Sensor des realen Roboters auch kein Rauschen aufweist. Dieser ist der Sensor auf dem die Regelung des Gelenkes arbeitet und dieser Wert wird auch an die anderen Komponenten weiter gegeben. Der Wert des absoluten Sensors soll nur zum Kalibrieren des inkrementellen Sensors beim Start des Roboters dienen.

2D-Laserscanner

Es wurden in den in der Simulation durchgeführten Experimenten ebenso wie im realen Experiment 1000 Werte pro Abstand (50mm, von 100 bis 1000mm in 100mm Schritten, von 1000 bis 5500mm in 500mm Schritten) und Farbe (Blau, Gelb, Grün, Lila, Orange, Rot, Schwarz und Weiß) durchgeführt. Diese Werte wurden zur späteren Analyse in Textdateien geschrieben.

Tiefenbild-Kamera (PMD-Kamera)

Es wurden für das virtuelle Experiment die Entfernungen von 100cm bis 400cm in 50cm Abständen vom Sensor zum Objekt eingestellt. Dies entspricht den realen Messungen aus dem realen Referenzexperiment. Der simulierte Sensor liefert ein Bild in dem die gemessene Tiefe als Farbe kodiert ist.

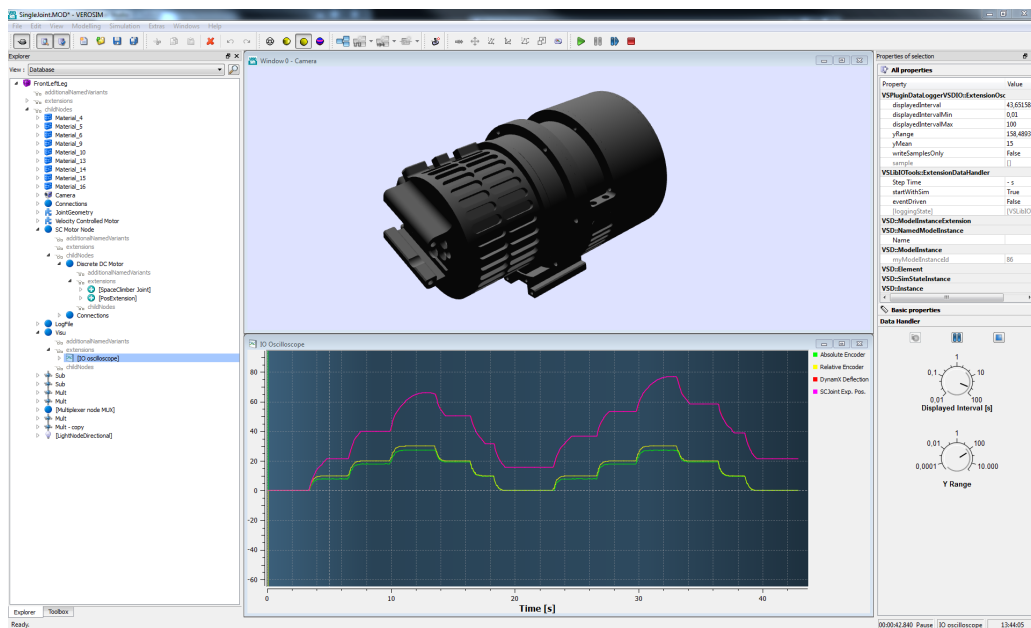


Abbildung 148: Screenshot von der Durchführung des Experiments für den SpaceClimber Joint-Position-Sensor

3.1.45 AP5430: Vergleich virtueller und realer Ergebnisse

Motormodell Ein Vergleich der von Simulation und Referenzexperiment im Motorteststand erzeugten Daten liefert interessante Ergebnisse. Zum Einen lassen sich die Motorparameter so anpassen, dass die Simulation für eine gegebene Last sehr gute Resultate erzielt, zum Anderen lässt sich kein Satz von Parametern finden, der für alle Lasten zu guten Ergebnissen führt (siehe Abb. 149).

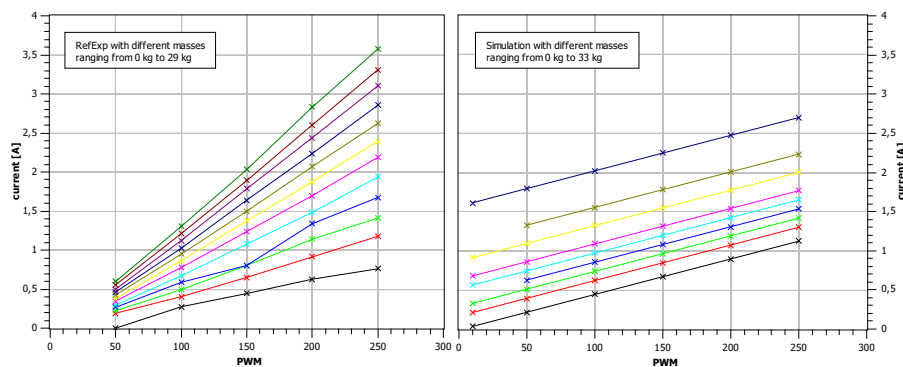


Abbildung 149: Stromverbrauch des Motors unter verschiedenen Lasten und PWM-Signalen. Die Steigung kann für eine gegebene Last angepasst werden, jedoch nie für alle zur gleichen Zeit.

Gängige theoretische Modelle für Brushless-DC-Motoren beschreiben den Stromverbrauch im stationären Fall durch

$$I_{modell}(V, T_L) = c_{T_L} \cdot T_L + c_V \cdot V(\text{PWM})$$

wobei I der Stromverbrauch des Motors an dem das Drehmoment T_L wirkt und die Spannung V anliegt bezeichnet. c_V und c_{T_L} sind Proportionalitätskonstanten und die Spannung hängt linear vom PWM-Signal ab.

Unsere empirischen Daten legen aber die Existenz eines weiteren Terms

$$I_{real} = I_{modell} + I_{empirisch}$$

mit

$$I_{empirisch} \propto T_L \cdot V(\text{PWM})$$

nahe.

Bodenmechanik Um die Flexibilität der verschiedenen Bodenmechanikimplementierungen zu untersuchen wurde der Sand im Bodenmechaniktestbed ausgetauscht. Der bisherige Boden wurde durch Basaltsand ersetzt. Dies ist das gleiche Material, dass am DFKI bereits in einer größeren Testanlage verwendet wird, in der der vollständig integrierte SpaceClimber getestet werden kann. So war es möglich Daten aus realen Experimenten, die mit dem gesamten SpaceClimber und nicht nur mit einzelnen Komponenten durchgeführt wurden, mit Daten aus der Simulation zu vergleichen.

Es wurden Experimente im Basalt durchgeführt (siehe Abbildung 150). Fünf exemplarische Datensätze für ein laterales Experiment sind in Abbildung 151 dargestellt.



Abbildung 150: Laterales Bodenmechanik Referenzexperiment mit Basaltsand.

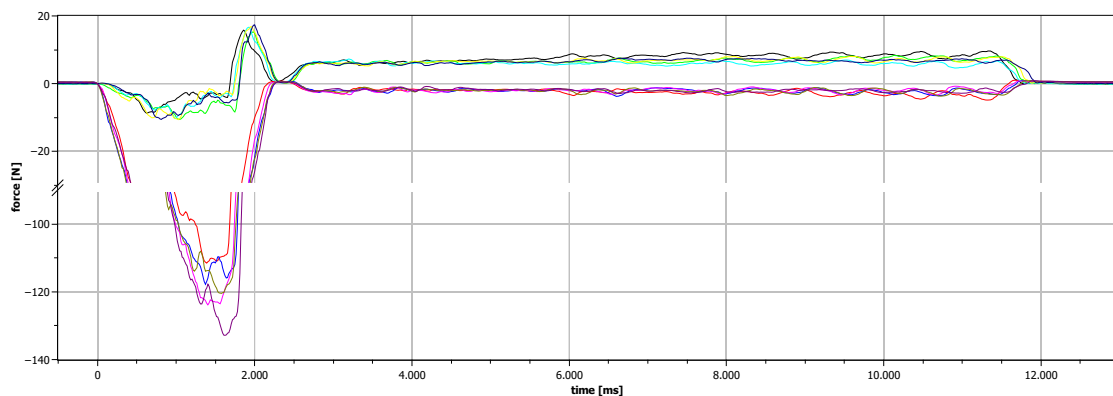


Abbildung 151: Daten fünf exemplarischer lateraler Bodenmechanik Referenzexperimente mit Basaltsand.

Bodenexperiment mit Neuronalem Netz Die Anbindung des neuronalen Netzes wurde innerhalb dieses Arbeitspaketes über den virtuellen Testaufbau (siehe 152) mit einem der real durchgeführten Experimente verglichen. In dem Experiment wurde der runde Fuss zweimal hintereinander an der gleichen Stelle in den Boden gedrückt. Das erste Eindringen wurde nach einer Tiefe von 1 cm beendet und das zweite Eindringen wurde bis zu einer Tiefe von 3 cm durchgeführt. Die Geschwindigkeit des Fusses betrug bei dem Experiment 1 mm/s. Die aus dem realen Experiment geloggten Gelenkwinkel wurden als Eingabe für den virtuellen Schunkarm verwendet. Der Variations-Eingabeparameter des neuronalen Netzes wurden manuell festgelegt. Die Abbildung 153 zeigt die gemessene Normalkraft des realen und des virtuellen Kraftsensors. Der größte Unterschied ist beim Herstellen des Kontaktes und beim Verlassen des Bodens zu erkennen.

Bodenexperiment mit PTP In Abbildung 154 ist ein Vergleich zwischen realem Referenzexperiment und Simulation mit dem Plastischen-Terramechanik-Partikel (PTP) Ansatz gezeigt. Es handelt sich dabei um das gleiche Experiment mit dem die Simulation mit

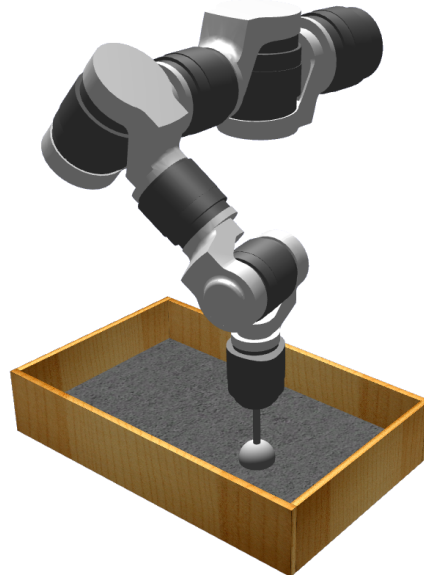


Abbildung 152: Virtueller Testaufbau zum Vergleich der Bodenmechanik in der Simulation mit den realen Experimentdaten.

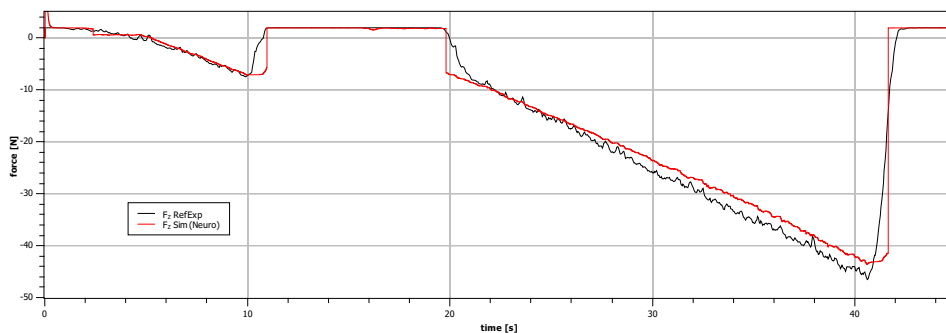


Abbildung 153: Gegenüberstellung der gemessenen Normalkräfte aus dem realen Experiment und dem virtuellen Experiment mit neuronalem Ansatz.

Neuronalem Netz verglichen wurde.

Das PTP Modell liefert durch die Wahl der richtigen Parameter in der vertikalen Komponente ein Ergebnis welches sehr nahe am Referenzexperiment liegt. Vor allem auch beim Herstellen und Abbrechen des Bodenkontakts wird der reale Kraftverlauf gut reproduziert.

Ein Vergleich zwischen den lateralen Kräften in realen Experimenten und Simulation zeigt, dass der PTP-Ansatz prinzipiell in der Lage ist realistische Vorhersagen zu treffen, auch wenn keine so gute Übereinstimmung wie im Fall der Normalkraft erreicht werden konnte (siehe Abbildung 154).

Ein im realen Experiment beobachtbarer Rückgang der Normalkraft beim Einsetzen der lateralen Bewegung (bei $t \approx 5$ s) wird qualitativ reproduziert. Allerdings scheint es

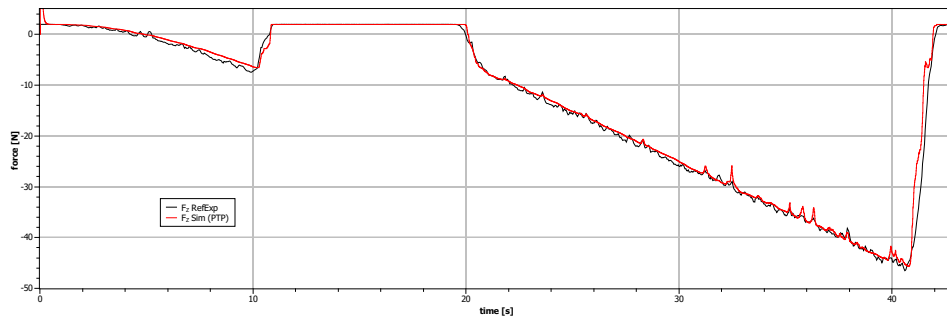


Abbildung 154: Gegenüberstellung der gemessenen Normalkräfte aus dem realen Experiment und dem virtuellen Experiment mit PTP Ansatz.

in der Simulation eine Sättigung sowohl der Normalkraft als auch der lateralen Kraft zu geben, die im realen Experiment nicht eintritt. Auch das Rauschen der lateralen Kraft ohne laterale Belastung ($t \lesssim 0$ s) sollte noch reduziert werden.

Vergleich des PTP Modells mit den lateralen Referenzexperimenten

Um das neue Lookup-Table basierte PTP Modell zu testen wurden Laterale Experimente mit einem $2 \times 3 \times 5$ cm³ Quader durchgeführt. Dieser wurde wie schon zuvor in den Boden gedrückt und dann horizontal durch den Untergrund gezogen. Dieses Experiment hat den Vorteil, dass die Fläche mit der der Quader gegen den Boden drückt genau bekannt ist und sich so die Gegenkraft gut in einen Druck umrechnen läßt. Dies ist Voraussetzung für eine Nutzung im PTP Modell. In [Abbildung 155](#) ist ein exemplarischer Datensatz zu sehen.

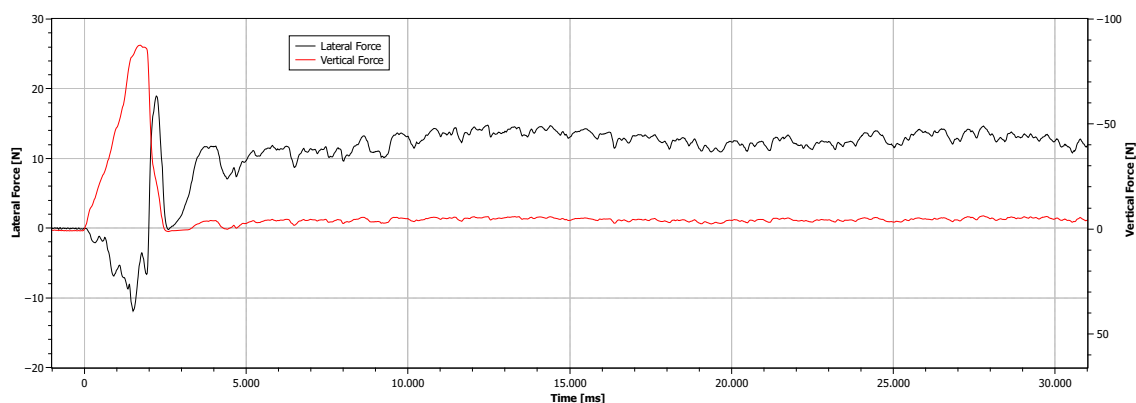


Abbildung 155: Daten eines exemplarischen lateralen Bodenmechanik Referenzexperimente mit einem $2 \times 3 \times 5$ cm³ großen Quader. Bei der lateralen Bewegung drückt die 3×5 cm² Fläche gegen den Sand.

Die Laterale Kraft scheint über weite Strecken konstant zu sein, wenn man von experimentellem Rauschen absieht. In diesem Fall ist die Lookup Tabelle trivial und gibt immer den gleichen konstanten Wert zurück.

Zur Verifikation des PTP Modells wurde in der Simulation ein Kugelfuß durch den Bo-

den gezogen und mit alten Referenzdaten verglichen. Diese Gegenüberstellung ist in Abbildung 156 zu sehen.

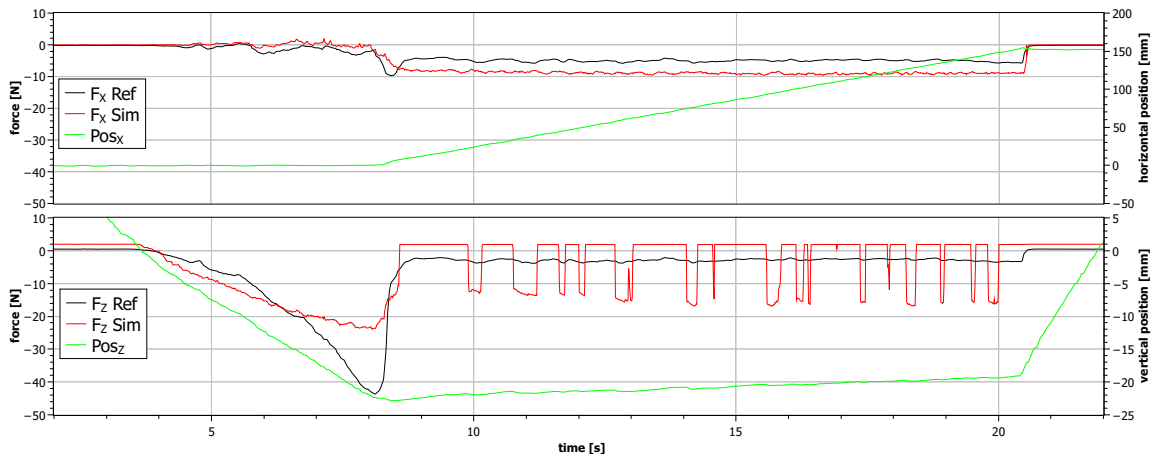


Abbildung 156: Vergleich von realem Experiment mit Simulation des PTP Modells bei lateraler Bewegung.

Die Übereinstimmung des Modells bei der lateralen Bewegung ist nicht so überzeugend wie im Falle einer rein vertikalen Bewegung. Die Kraft ist in der lateralen Phase konstant zu hoch.

Force-Torque-Sensor

Der Aufbau des virtuellen Experimentes nach Vorbild des realen Experimentes stellte kein Problem dar. Der Aufbau ist relativ einfach und wurde nicht durch äußere Einflüsse gestört. Ein Problem bei dem Nachbau war die nicht klare Angabe an welcher Stelle der genaue Punkt der Kraftmessung ist. Dieser hat einen Einfluss auf den gemessenen Torque.

Die Abbildung 157 zeigt den ersten Versuch im virtuellen Experiment. Es ist gut zu erkennen, dass die Graphen für die gemessene und simulierte Kraft F_x gut zusammenpassen. Die Messwerte von "0" in der Log-Datei sind auf einen Fehler des Programms, welches die Datei erstellt hat zurück zu führen und keine Werte, die der Sensor liefert. Diese falschen Messwerte treten bei dem Einsatz im Roboter nicht auf und dürfen nicht von der Simulation mit simuliert werden.

Für den Torque-Wert ist ein Fehler in Form einer Verschiebung des simulierten Wertes festzustellen. Dieser Fehler ist auf eine falsche Position des Sensors in der Simulation zurück zu führen. Es wurde angenommen, dass der Punkt der Messung sich zwischen dem Sensor und dem Hebel befindet. Dieser Punkt befindet sich aber auf der anderen Seite der Messdose zwischen Dose und (im realen Experiment) dem Tischbein. In Abbildung 158 ist das gleiche Experiment mit der korrigierten Position zu sehen. Nun sind auch die Werte für T_y im Rahmen des Rauschens identisch.

In der Abbildung 159 sind die Messwerte für das gleiche Gewicht aber einem verkürzten Hebel von 105 mm zu sehen. Auch in diesem Fall decken sich die Ergebnisse aus dem virtuellen und dem realen Experiment.

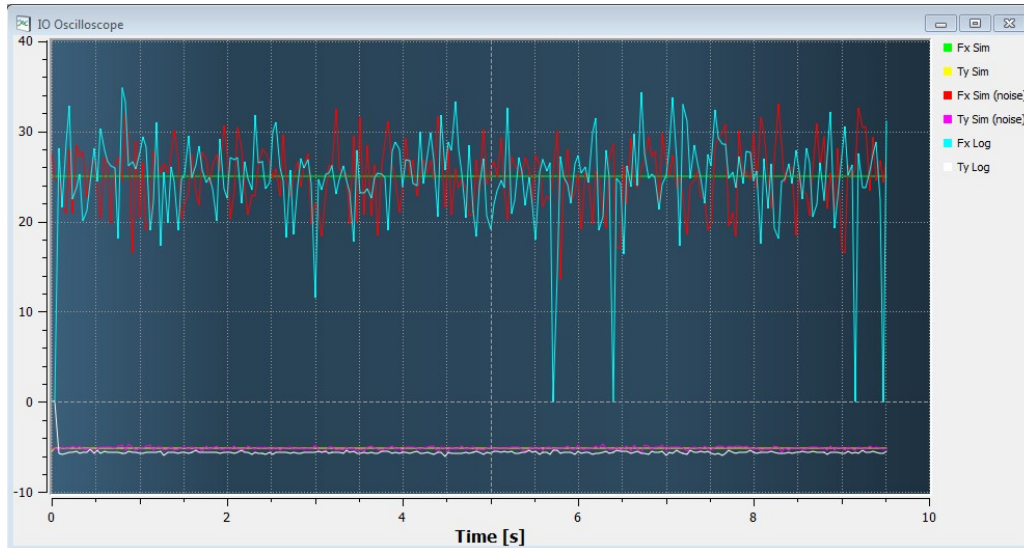


Abbildung 157: Plot mit der falschen Position des Sensors (2,544 Kg und 205 mm Hebel)

Joint-Position-Sensor

Der Nachbau des Experimentes erwies sich als problematisch. Dies hat mehrere Gründe. Als erstes zeigen die Werte des realen Experimentes ein Problem mit dem absoluten Sensor im realen Gelenk. In [Abbildung 160](#) sind die Werte für den ersten Durchgang zu sehen. Es wurden dem Gelenk eine Abfolge von Positionen zum Anfahren vorgegeben. In dem Fall 0° , 10° , 20° , 30° dann wieder über 20° und 10° auf 0° und im Anschluss nochmal die gleiche Abfolge. Zu beobachten ist, dass der Absolutsensor die Werte des inkrementellen Sensors nicht erreicht. Das Gelenk selbst hat die Positionen erreicht. Der inkrementelle Sensor funktioniert folglich. Das Problem des absoluten Sensors liegt in seiner Konstruktion begründet. Es wird eine Deckoderscheibe mit einer 2/1 Untersetzung angetrieben. Diese Untersetzung hat etwas Spiel, was zu den falschen Werten führt. In dem ersten Durchgang könnte man noch davon ausgehen, dass dieser Fehler nur in eine Drehrichtung auftritt. Ein weiterer Versuch zeigt aber deutlich, dass der Fehler in beide Drehrichtungen auftritt und abhängig ist, von der Stellung des Gelenks beim Einschalten. Dies ist in der [Abbildung 161](#) zu sehen.

Aufgrund dieser Probleme wurde der Sensor auch nicht zum Kalibrieren der Gelenke beim Einschalten genutzt. In einer überarbeiteten Version des Gelenkes soll dieser Fehler behoben werden. Es stand jedoch kein neues Gelenk für erneute Experimente zur Verfügung.

Trotz dieser Probleme mit dem realen Experiment wurden die Experimente in der Simulation durchgeführt. Ein Plot dieser virtuellen Experimente ist in [Abbildung 162](#) zu sehen. Hier fällt auf, dass die Position in der Simulation deutlich von der Position des realen Gelenks abweicht. Diese Abweichung liegt daran, dass im realen Experiment das Gelenk auf eine Position geregelt wurde. In dem simulierten Fall wurde das PWM-Signal aus der Log-Datei zum Ansteuern des Motors genutzt. Das Problem liegt in der Frequenz mit dem die Werte in die Log-Datei geschrieben wurden und der Frequenz

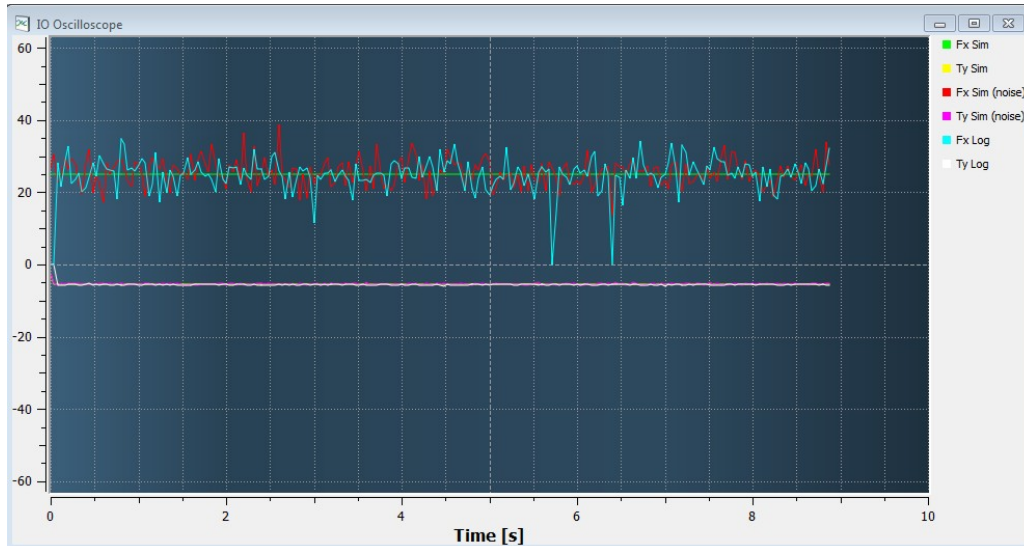


Abbildung 158: Plot mit der korrigierten Position des Sensors (2,544 Kg und 205 mm Hebel)

mit der das PWM-Signal im echten Gelenk geregelt wurde. Die interne Frequenz ist um ein Vielfaches höher, so dass die Werte in dem Log-File nicht direkt für die Ansteuerung genutzt werden können, da viele Zwischenschritte einfach nicht abgebildet sind.

Eine Möglichkeit wäre es gewesen, auch in der Simulation eine Regelung auf eine Soll-Position durch zu führen. Da aber sowohl in der Realität als auch in der Simulation der inkrementelle Sensor ohne Rauschen mit einer sehr hohen Genauigkeit funktionieren wurde in dieses Experiment kein weiterer Aufwand investiert, da weitere Ergebnisse keinen relevanten Beitrag zur Verbesserung der Gesamtsimulation des Roboter-Systems gebracht hätten.

2D-Laserscanner

Zunächst wurden die Ergebnisse der realen Experimente untersucht. In den Abbildungen 163 und 164 wurde jeweils einer der Messdurchgänge mit jeweils 1000 Messungen geplottet. Einmal würde die Messung mit einem Abstand von 100 mm und einmal mit 2500 mm zum Objekt durchgeführt. Auffällig ist das Schwingen des Messwertes am Anfang einer Messung. Dieses Schwingen tritt nicht jedes mal auf. Es ist nicht klar, wo dieser Fehler her kommt. Es besteht die Möglichkeit, dass dieser Fehler auf die Messungen zurück zu führen sind, bzw. auf eine Einwirkung von der Person, die die Messung durchgeführt hat. Es ist denkbar und wahrscheinlich, dass die durchführende Person beim Start der Messung den Sensor ungewollt bewegt oder in eine leichte Schwingung versetzt hat, was zu diesem Messfehler geführt hat. Für die Betrachtung des Fehlermodells wird dieser nur am Anfang einer Messung auftretende Fehler außer Acht gelassen.

Der Sensor ist auf eine Reichweite von 4 m ausgelegt. Sofern der Sensor einen Messwert über 4 m liefert sollte dieser Wert nicht oder nur sehr vorsichtig von einer auswer-

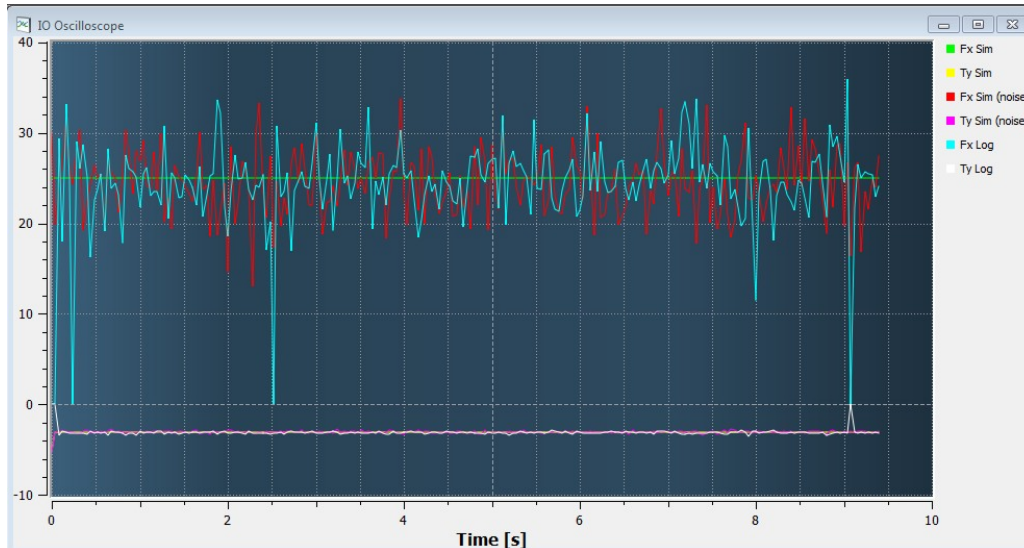


Abbildung 159: Plot mit der korrigierten Position des Sensors (2,544 Kg und 105 mm Hebel)

tenden Software eingesetzt werden. Wenn ein Objekt deutlich über 4 m entfernt ist, so gibt der Sensor einen Fehler-Code, in diesem Fall "1äus. Somit können und sollten diese Werte leicht von der auswertenden Software ausgefiltert werden. Es wurde sich also bei den Experimenten und dem Vergleich der Sensoren auf das Rauschen des Sensors konzentriert, da dieses die höchste Relevanz für eine auf dem Robotersystem laufende Software hat.

Für die weiteren Untersuchungen wurden bei den realen Experimenten die ersten 200 Messwerte nicht berücksichtigt. Es wurde bei den Messwerten eine Verschiebung der Messwerte nach oben festgestellt. Es wird also im Mittel eine etwas größere Entfernung angegeben. Diese Verschiebung wächst leicht mit dem Messwerten an. Zu dieser Verschiebung kommt noch ein Rauschen. Dieses Rauschen ist über die unterschiedlichen Distanzen nahezu konstant.

Wie die Plots in den Abbildungen 165 und 166 zeigen, sind die simulierten Werte für eine Entfernung von 100mm etwas besser. Dies liegt an der nicht mit modellierten Verschiebung der Messwerte. Um die Werte besser Abbilden zu können, müsste dies noch mit in das Fehlermodell des Sensors eingebunden werden.

Tiefenbild-Kamera (PMD-Kamera)

In der Form wie die Ergebnisse der virtuellen Referenzexperimente zur PMD-Kamera und die Ergebnisse der in der Realität durchgeführten Ergebnisse vorliegen, ist ein direkter Vergleich der Daten nicht möglich. Die Daten aus dem realen Versuch liegen als Zahlenwerte vor, während die Ergebnisse aus der Simulation nur als Bilder vorliegen. Hier hätte das virtuelle Experiment nachbearbeitet werden müssen, um vergleichbare Daten zu liefern. Da aber auf dem zu simulierenden Roboter SpaceClimber keine PMD-Kamera verbaut ist, wurde dies nicht weiter verfolgt. Zur Simulation des Robotersystems ist dieser Sensor nicht notwendig und es wurde sich bei den Experimenten

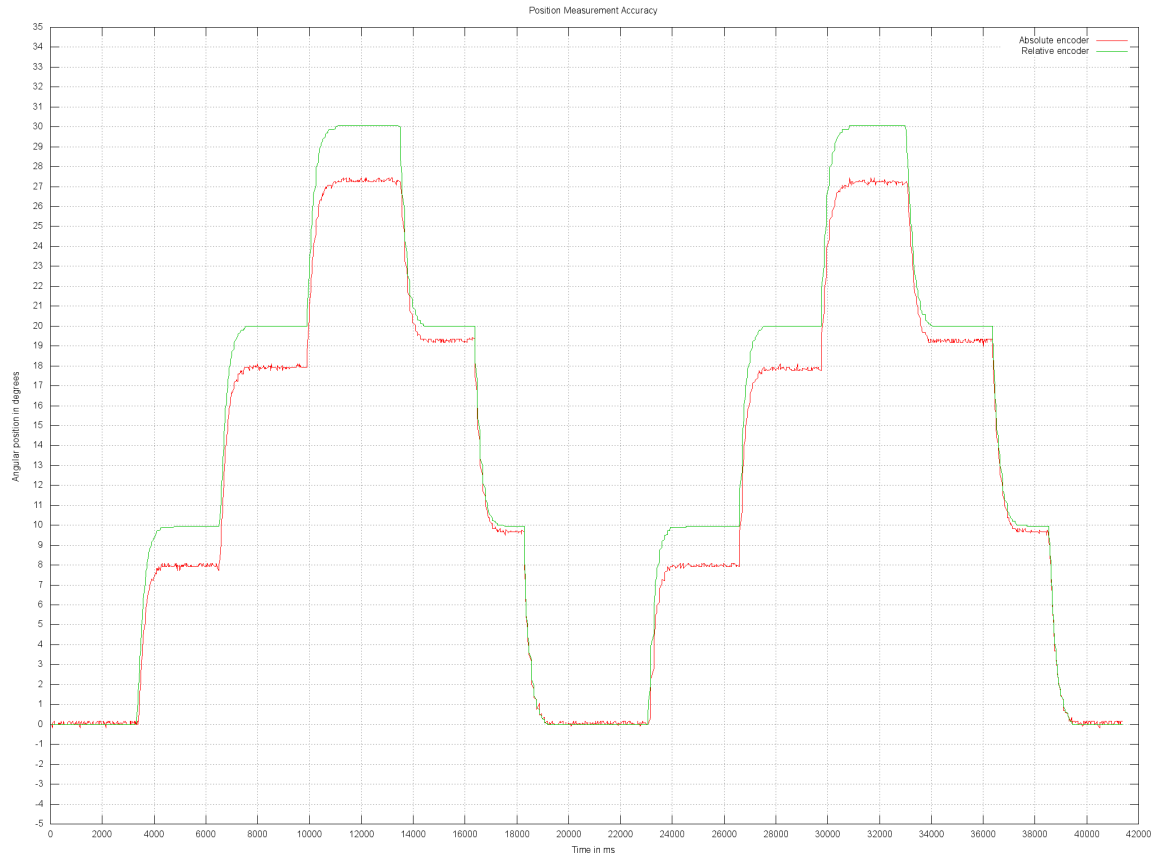


Abbildung 160: Plot der Werte real gemessenen Positionswerte vom SC-Gelenk (1)

auf andere Sensoren bzw. die Bodendynamik konzentriert.

Aber auch wenn kein direkter Vergleich zwischen dem realen und dem simulierten Sensor durchgeführt wurde, so ist es mit dem Simulationssystem VEROSIM[®] möglich, eine PMD-Kamera zu simulieren und auch das eingebaute Fehlermodell auf den Sensor anzuwenden.

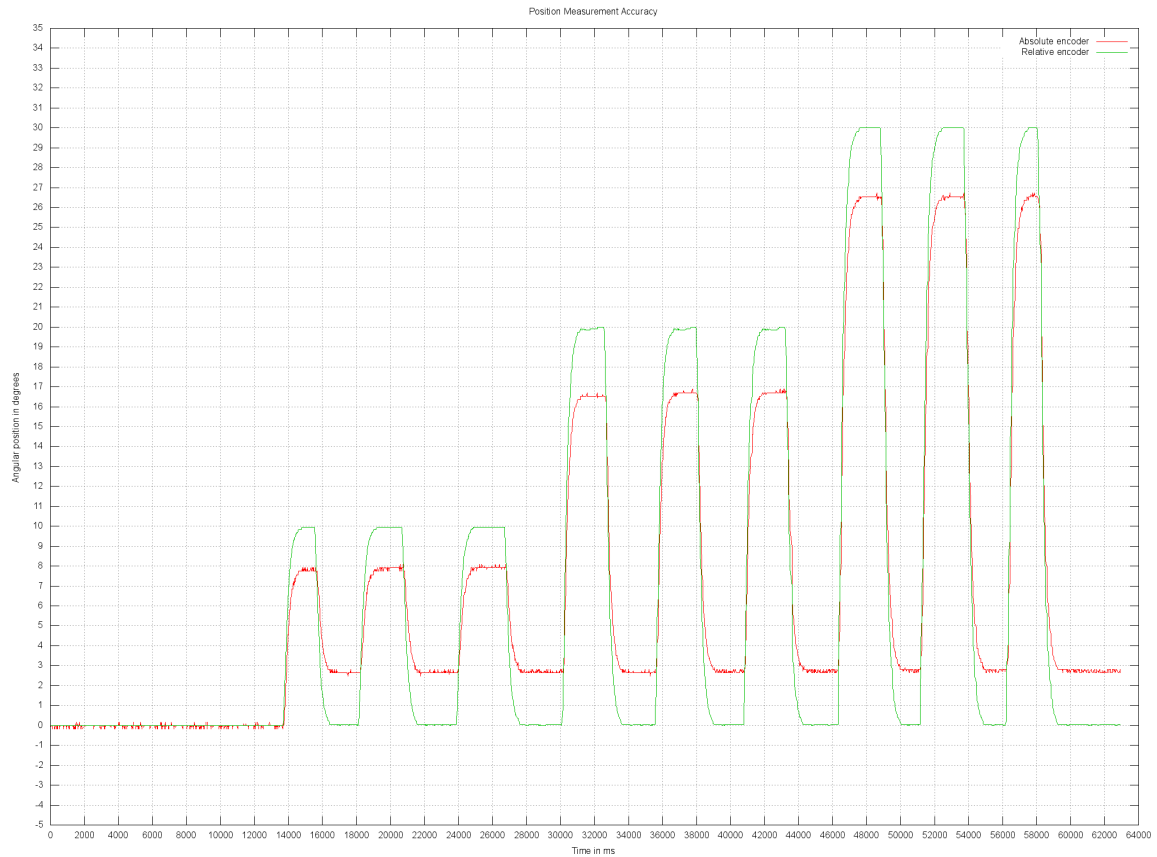


Abbildung 161: Plot der Werte real gemessenen Positionswerte vom SC-Gelenk (2)

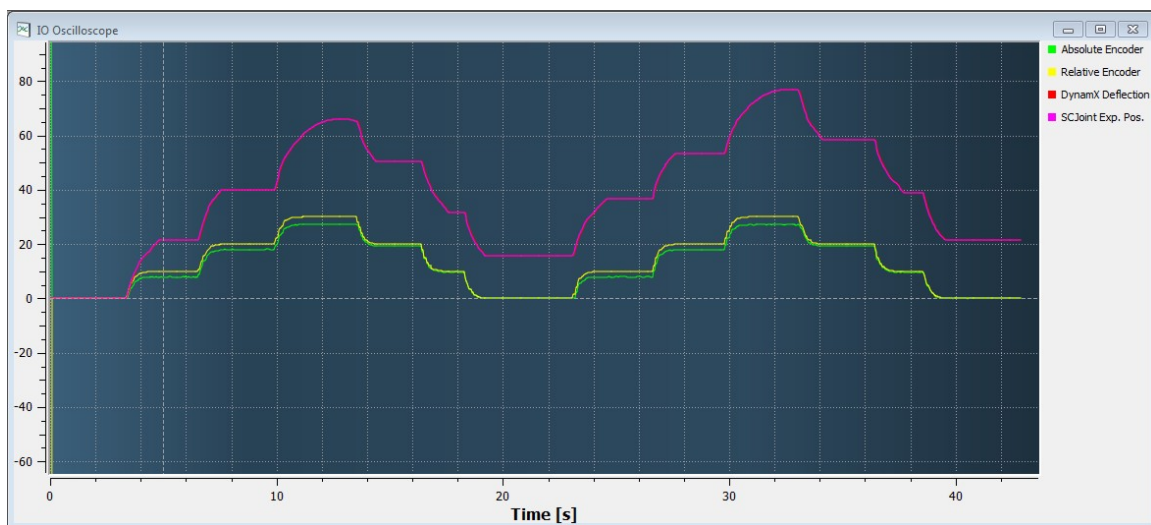


Abbildung 162: Plot der Positionswerte des SC-Gelenks aus der Simulation und reale Werte

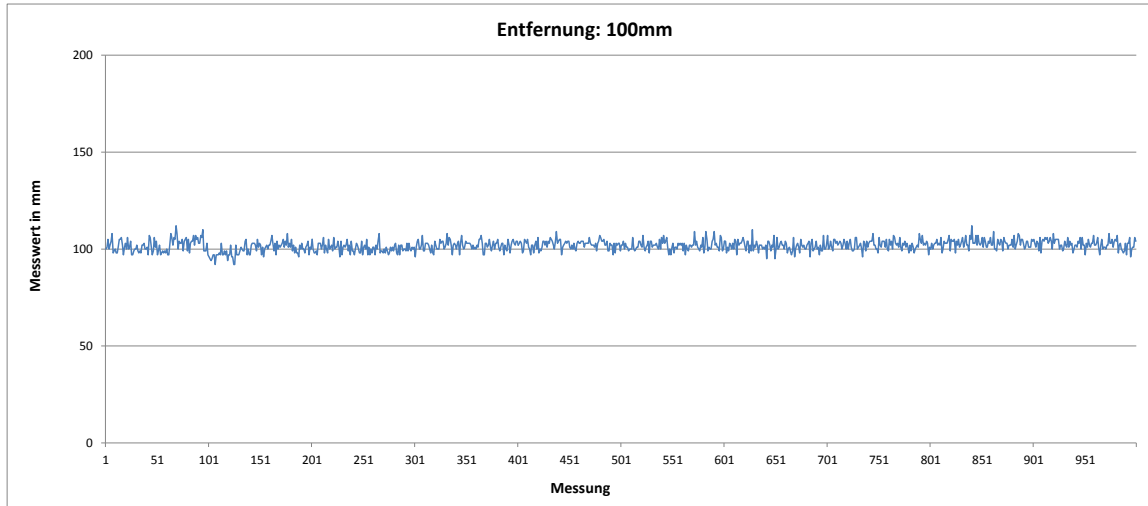


Abbildung 163: Plot der realen Entfernungsmessung. 100 mm Entfernung

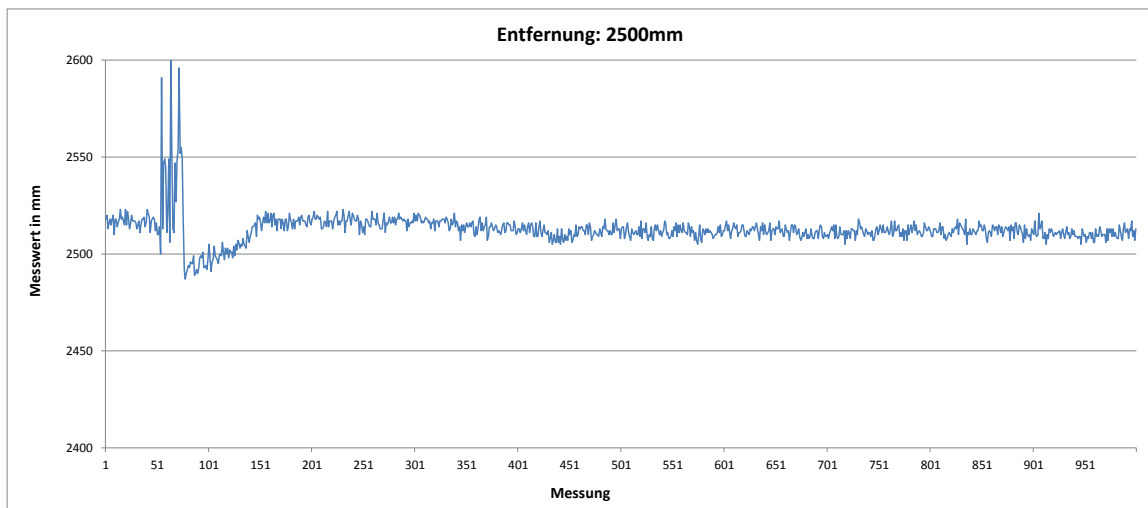


Abbildung 164: Plot der realen Entfernungsmessung. 2500 mm Entfernung

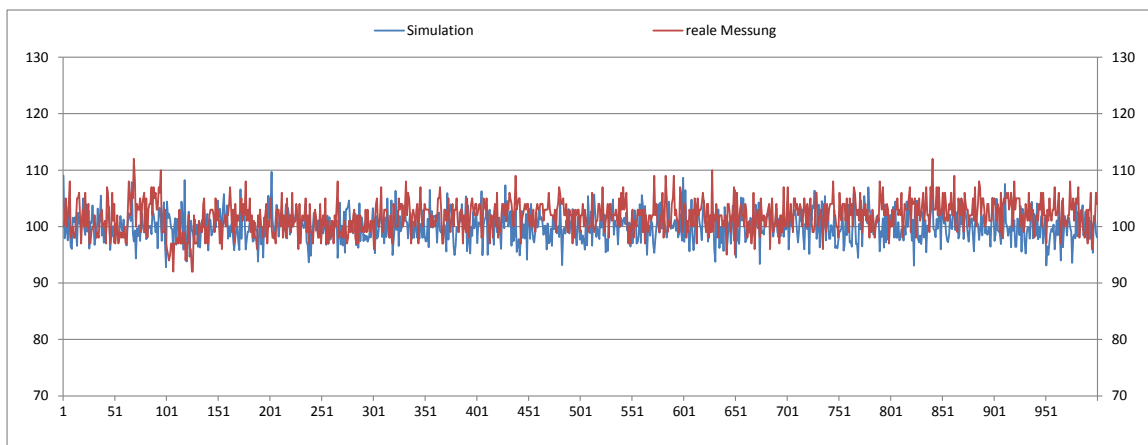


Abbildung 165: Plot der realen und virtuellen Entfernungsmessung im Vergleich: 100 mm Entfernung

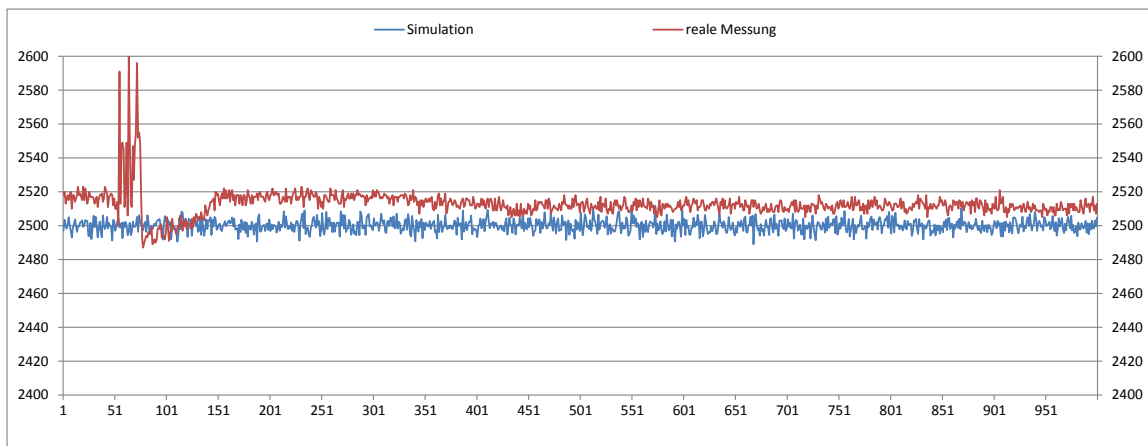


Abbildung 166: Plot der realen und virtuellen Entfernungsmessung im Vergleich: 2500 mm Entfernung



3.1.46 AP6000: Einsatz und Extrapolation

3.1.47 AP6100: Definition einer Referenz-Mondmission

Mondmissionen im LUNARES-Projekt Im LUNARES-Projekt wurden verschiedene Weltraumszenarien wie zum Beispiel „Ausbringung eines Geophysikalischen Environment Packages (GEP) mit Seismometer und Wärmestromsensor“ oder „Astro-Habitat mit biologischen Experimenten zur Strahlungsumgebung“ auf ihre Eignung als Referenz Szenario evaluiert. Es wurde das Szenario „In-Situ Analyse von Bodenproben aus einem polaren Mondkrater auf der zentralen Messstation an Bord einer Landefähre“ ausgewählt. Dieses Missionsszenario soll im folgenden näher beschrieben werden, da sich die Referenz-Mondmission von Virtual-Crater an diesem Szenario orientieren wird.

Nach der Landung einer Raumfähre auf dem Mond wird ein heterogener Roboter-verbund aus einem Rover und einem Laufroboter („Scout“) abgesetzt. Der Rover nähert sich von der Fähre geleitet dieser an, um sich mit einer Nutzlast bestücken zu lassen. Als nächstes fährt der Rover selbstständig zum Rand des Kraters um dort den Laufroboter abzusetzen.

Der Scout beginnt darauf hin den Abstieg in den Krater wobei er Steigungen von bis zu 35° und verschiedene Unebenheiten bewältigen muss. Er wird dabei nach Möglichkeit vom Rover durch Beleuchtung des Kraters und visuellen Input unterstützt. Am Boden des Kraters sammelt der Scout eine Gesteinsprobe ein. Dieser Vorgang kann auch als stellvertretend für die In-Situ Analyse betrachtet werden. Nach erfolgreicher Probenentnahme kehrt der Laufroboter durch selbstständiges Erklimmen der Kraterwand zum Rover zurück.

Die beiden mobilen Roboter führen dann einen kooperativen Dockingvorgang aus, um als Verbund zur Landefähre zurück zu kehren. Durch eine Übergabe der Bodenprobe an die Landefähre endet die Mission.

Anpassung der Mission an Virtual-Crater Der Fokus des Virtual-Crater Projekts liegt auf Simulation, Interaktion und Visualisierung. Dadurch ergeben sich einige Unterschiede zu LUNARES hinsichtlich der Anforderungen an die Referenz-Mission. Zum Beispiel wurde sich auf die Simulation des Laufroboters SpaceClimber konzentriert, so dass eine Kollaboration mit einer Landefähre oder einem Rover im Rahmen von Virtual-Crater nicht vorgesehen ist.

Es wird stattdessen Wert auf die Simulation der Bodenmechanik und Motoren und der intuitiven Visualisierung von Größen wie auftretenden Kräften bei Bodenkontakt oder in den Motoren gelegt. Weiterhin ist geplant die Interaktionsmöglichkeiten die eine Visualisierung in der Mehrfach-Projektions-Umgebung bietet zum Bestandteil der Referenzmission zu machen.

Definition der Referenz-Mission Die Mission stellt eine simulierte Erkundung eines Mondkraters mittels des sechsbeinigen DFKI Roboters SpaceClimber dar. Ziel der



Mission ist anhand eines realistischen Szenarios die im Rahmen des Virtual-Crater Projekts entwickelten Simulationsmodule zu demonstrieren. Das Szenario entspricht wie im LUNARES-Projekt der Erkundung eines Mondkraters, wie sie zur Beantwortung vielfacher wissenschaftlicher Fragestellungen nötig ist. Es wird angenommen, der Roboter wurde von einem geeigneten Rover von einer Landefähre zu einem zu erkundenden Krater gebracht und dort abgeladen. Der SpaceClimber befindet sich zu Beginn der Mission also auf ebenem Terrain nahe dem Einstiegspunkt in den Krater (siehe Abbildung 167). Dort startend soll der SpaceClimber einem vorgegebenen Pfad in den Krater hinunter folgen. Dabei muss das Laufverhalten an die Steigung des Kraters angepasst werden. Am Kraterrand ist ein weiteres robotisches System welches mit einem Strahler den Weg für den SpaceClimber ausleuchtet. Unten angekommen verrichtet der SpaceClimber seine Arbeit und steigt danach den Kraterrand wieder hinauf. Innerhalb der Mondmission sollen verschiedene Aspekte der Simulation, wie die Bodendynamik und die Visualisierungsmöglichkeiten von Systemzuständen, demonstriert werden. Dafür wurde folgender Ablauf definiert:

1. Mit dem Verändern der Roboterhaltung soll der simulierte Neigungssensor mit realen Sensordaten verglichen werden.
2. Zudem soll durch das Verändern der Haltung die visuelle Repräsentation der Motorbelastungen demonstriert werden.
3. Während sich der SpaceClimber auf den Krater zu bewegt soll die Bodenmechanik genauer vorgestellt werden.
4. In der Schräge soll nochmals auf die Bodendynamik eingegangen werden indem ein steigungsabhängiges Rutschen demonstriert wird. Dieses Rutschen verhindert nicht das vorankommen des Roboters weist aber auf eine Instabilität hin die durchaus kritisch sein kann. Durch die genaue Simulation dieses Verhaltens wird eine Abschätzung ermöglicht unter welchen Bedingungen welche Steigungen bewältigt werden können ohne das System unnötig einer Gefahr auszusetzen.
5. Für die vorherigen Punkte soll die realistische Ausleuchtung des Kraters deaktiviert sein, um das Simulationsverhalten bestmöglich wahrnehmen zu können. Wenn das Kraterinnere vom SpaceClimber erreicht wurde soll die realistische Beleuchtung der Simulation demonstriert werden.
6. Mit den realistischen Lichtverhältnissen soll der Kamerasensor vorgestellt werden. Währenddessen wendet sich der SpaceClimber wieder dem Kraterrand zu.
7. Durch das Einschalten des Strahlers zum Ausleuchten des Krater sollen weitere Aspekte der Simulationsbeleuchtung vorgestellt werden.
8. Während der SpaceClimber den Kraterrand wieder hinaufklettert wird der harte Übergang des schattierten Kraterinneren zum belichteten Kraterrand vorgestellt.
9. Beim Hinaufklettern soll nochmals der Einfluss der Laufparameter (Haltung des Roboters) auf die Effizienz des Laufverhaltens demonstriert werden.

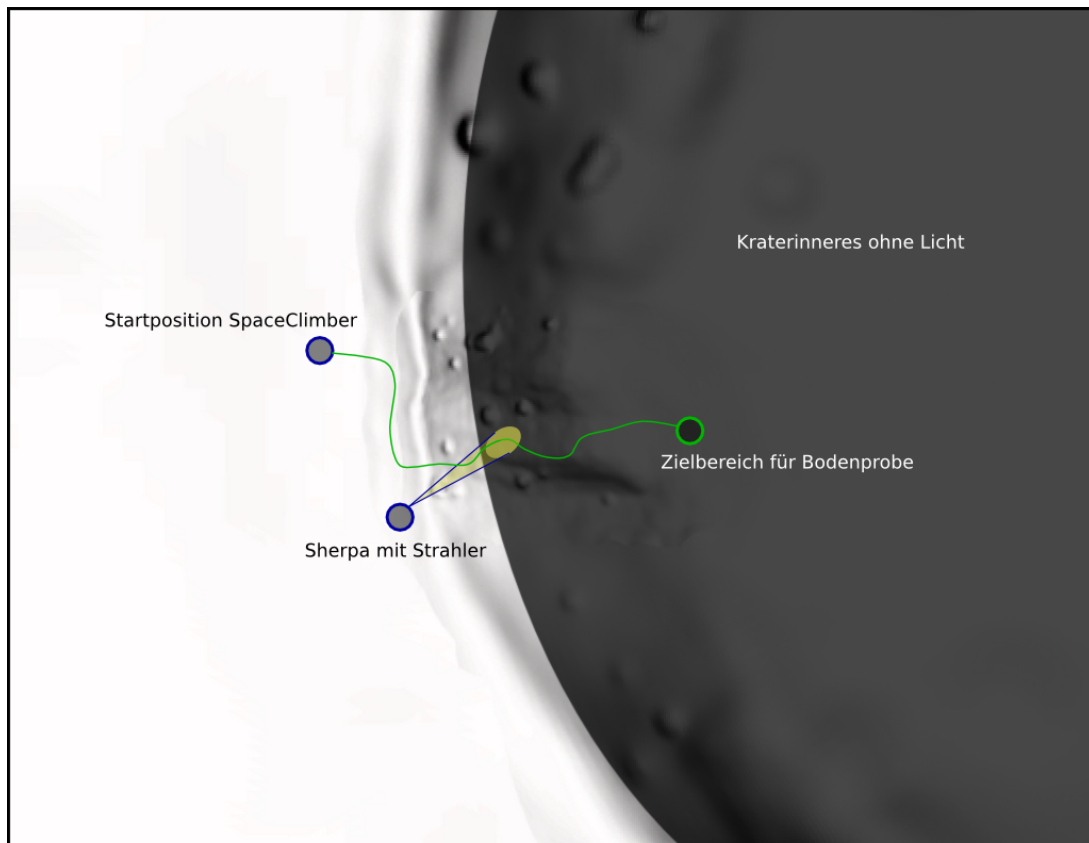


Abbildung 167: Mondmissions-Übersicht.

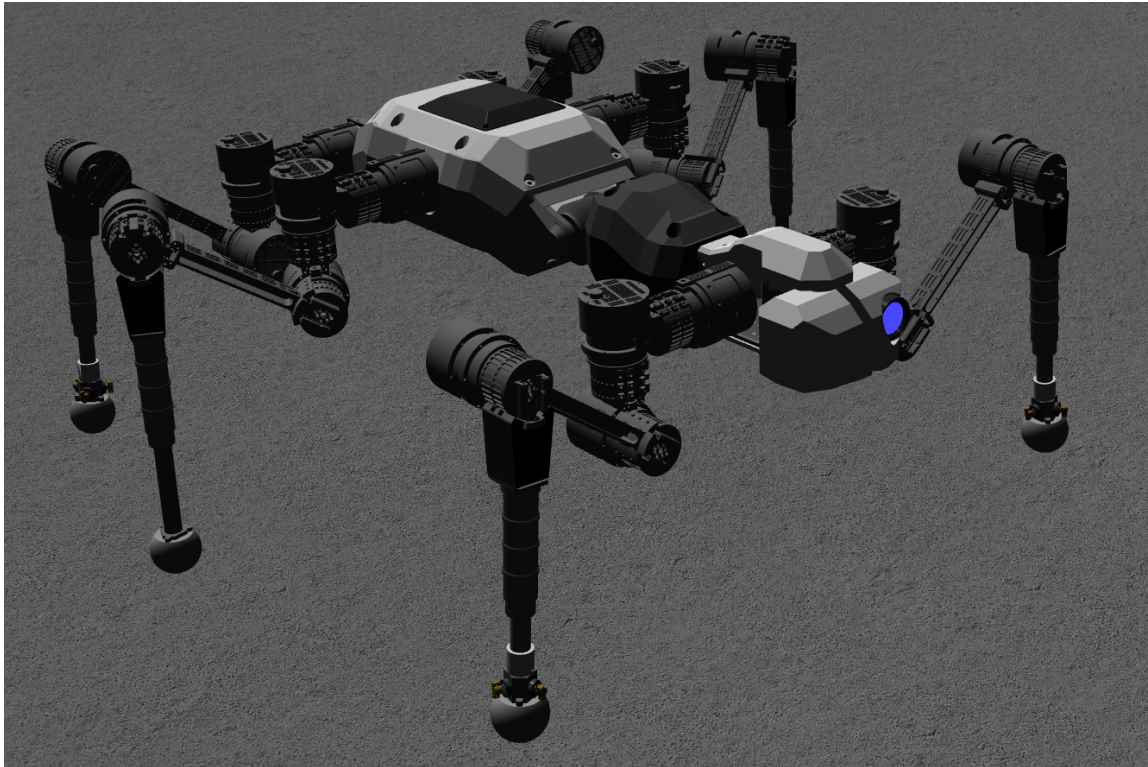


Abbildung 168: Ein detailliertes Modell des SpaceClimbers.

3.1.48 AP6200: Modellierung and Implementierung einer virtuellen Referenz- Mondmission

Eine fiktive Mondkraterumgebung, die als Ausschnitt ein Modell der Kraterwand aus der DFKI Weltraumexplorationshalle enthält, wurde in VEROSIM[®] erstellt. Diese besteht aus mehreren Höhenkarten mit variierender Auflösung um den Rechen- und Speicheraufwand überschaubar zu halten und dennoch an den entscheidenden Stellen die benötigte Auflösung zu bieten. Die DFKI Kraterwand hat dabei die höchste Auflösung von etwa 10 cm. Die Höheninformation ist als 16-Bit Grauwerte abgespeichert, so dass bei einer Höhendifferenz von 10 m eine Auflösung von knapp 0,15 mm erreicht wird.

Um die Portierung von VEROSIM[®] Modellen, die auf dem Desktop entwickelt wurden, auf die CAVE zu erleichtern, wurde daran gearbeitet einen Workflow zu etablieren. Dazu wurden die durchzuführenden Schritte identifiziert und in einem Best Practice Guide festgehalten. Auch gängige Fehler und Problem wurden zusammengetragen und mit möglichen Lösungen dokumentiert. Dies sollte es in Zukunft erleichtern Entwicklungen vom Desktop schnell auf die CAVE zu portieren. In einem weiteren Schritt könnte auf Grundlage dieser Arbeit an einer Automatisierung des Prozesses gearbeitet werden.

Neben der bereits bestehenden Modellierung des Kraters wurde ein neues detaillierteres Model des SpaceClimbers aus einem CAD-Modell importiert (siehe Abbildung 168).

Desweiteren wurde ein statisches Modell des DFKI Sherpa Roboters in die Szene

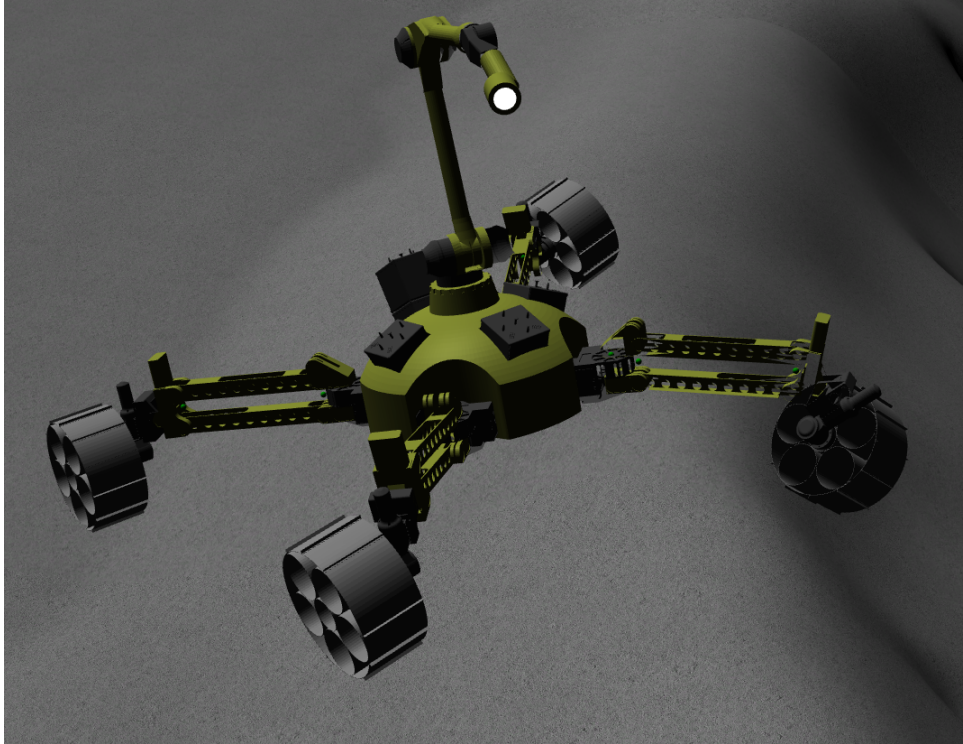


Abbildung 169: Ein Modell des DFKI Sherpa Rovers, der dem SpaceClimber beim Abstieg in den Krater den Weg leuchten soll.

integriert. Dieser soll wie in der Referenzmissionsdefinition beschrieben, dem SpaceClimber beim Abstieg in den Krater den Weg leuchten. Dazu wurde ein neues Plugin implementiert, welches dafür sorgt, dass ein Objekt (in diesem Fall der Scheinwerfer) immer in Richtung eines Referenzobjekts (hier ein Punkt vor dem SpaceClimber) ausgerichtet ist. Das Modell des Rovers ist in [Abbildung 169](#) zu sehen.

Die VEROSIM[®] Kontrollsoftware wurde dahin gehend erweitert, dass es nun möglich ist in der laufenden Mission verschiedene informative Elemente, wie ein BarChart der Gelenkwinkel und ein Oszilloskop mit Gelenkdaten, ein und aus zu schalten (siehe [Abbildung 170](#)). Dies ist ebenfalls mit den Visualisierungen des Laserscanners und des Kamerasensors möglich. Ebenfalls läßt sich in der laufenden Simulation zwischen realistischer Beleuchtung mit Schatten und einer unrealistischen globalen Beleuchtung umschalten.

Die im Quartalsbericht 2012Q2 in AP4350 beschriebene Visualisierung der Fußabdrücke wurde ebenfalls in das Referenzmissionsmodell integriert und vom Projektpartner MMI mit einem OpenGL Shader ausgestattet, der die visuelle Darstellung inklusive Schatten, Texturen und Normalmapping an die bestehende Szenerie anpasst (siehe [Abbildung 171](#)).

Ein einfaches Pfadverfolgungsplugin wurde derart angepasst, dass die Parameter wie Start- und Stopwinkel der Bewegung nun direkt in VEROSIM[®] über Properties eingestellt werden können.

Eine Übersicht über die laufende Referenzmission mit realistischer Beleuchtung und

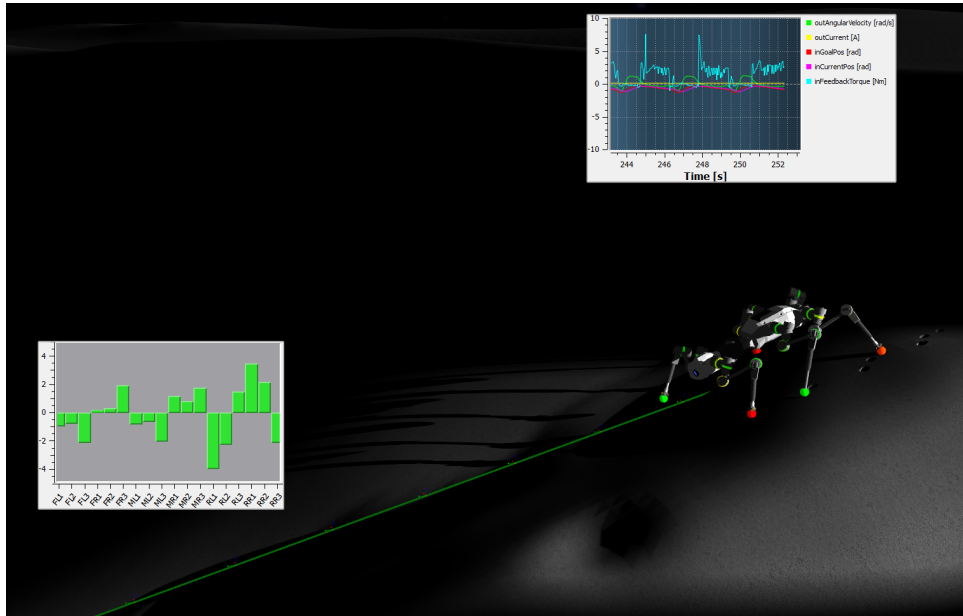


Abbildung 170: Verschiedene Visualisierungsmöglichkeiten. Unten links: Balkendiagramm mit Gelenkwinkeln. Oben rechts: Oszilloskop mit Motorwerten. Am SpaceClimber: Farbkodierung der Gelenkbelastungen. All diese Elemente lassen sich während der Simulation anzeigen und verbergen.

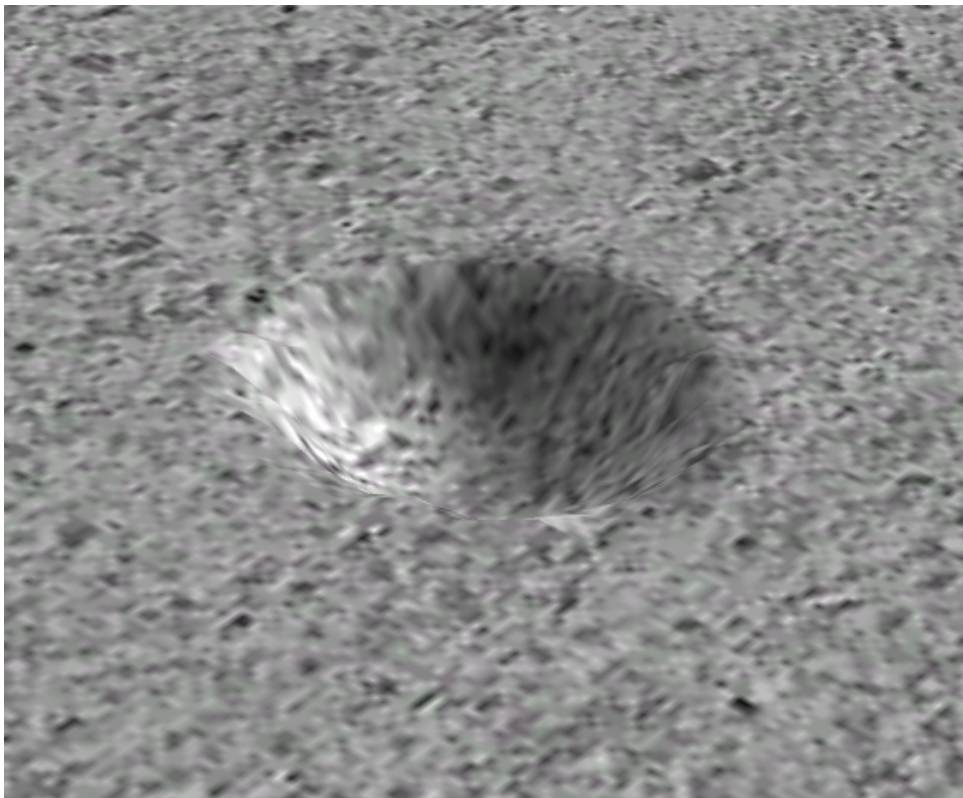


Abbildung 171: Ein detaillierter Fußabdruck mit Textur und Schatten in VEROSIM®.

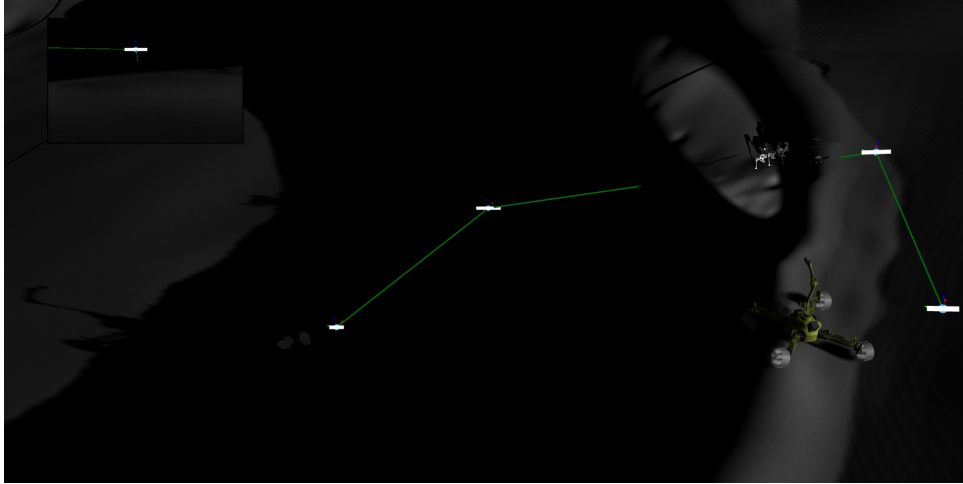


Abbildung 172: Übersicht über die laufende Referenzmission mit realistischer Beleuchtung und aktiviertem Kamerasensor. Im Bild des Kamerasensors (oben links) ist deutlich der Kegel des Scheinwerfers zu sehen. Der geplante Pfad ist als grüne Linie dargestellt.

aktiviertem Kamerasensor ist in [Abbildung 172](#) zu sehen.



3.1.49 AP6300: Demonstration Referenz-Mondmission

Die Simulationssoftware und das Modell der Referenzmission wurden auf einen finalen Stand gebracht und auf die Mehrfachprojektionsumgebung übertragen. Ein interner Testlauf für die Enddemonstration gegenüber dem Projekträger am 30.10.2012 wurde erfolgreich durchgeführt. Die getestete Funktionalität umfasst dabei unter anderem die Darstellung detaillierter CAD Modelle in einer großen Kraterumgebung (AP4360), eine realistische Ausleuchtung dieser Szenerie inklusive kontrastreicher Schatten (AP4310), echtzeitfähige Bodenmechanik in Form des neuronalen Netz Ansatzes inklusive detailreicher Visualisierung (AP4350), 24 abstrakte Motormodelle zur Simulation der SpaceClimber Aktuatoren inklusive Simulation der erwarteten Stromaufnahme (AP4340) sowie verschiedene Visualisierungsmöglichkeiten wie Oszilloskop, Balkendiagramm, Kamerasensor, Kraftdrehmomentssensor und Laserscanner (AP4310 und AP4330). Bilder und nähere Beschreibungen dieser Komponenten können den entsprechenden Arbeitspaketen und AP6200 entnommen werden.

Neben der komponentenweisen Überprüfung wurde auch sichergestellt, dass die oben genannten Module gemeinsam in einer integrierten Simulation funktionieren.

3.1.50 AP6400: Auswertung der Ergebnisse der virtuellen Referenzmission

Zusätzlich zu der Referenzmission wurde der Gesamtroboter mit Aufnahmen aus dem SpaceClimber Projekt verglichen. Hierfür wurden Daten verwendet die auf einem harten Untergrund mit vier verschiedenen Laufverhalten aufgenommen wurden. In Graphen dargestellt wird die zurückgelegte Distanz, die Geschwindigkeit des Roboters, die für die Fortbewegung verwendeten Gelenkpositionen eines mittleren Beines und die Gesamtstromaufnahme des Roboters.

Trotz einiger Faktoren die nicht näher in dem Projekt betrachtet werden konnten, kann eine relativ gute Vorhersage über das Roboterverhalten mit der Simulation gemacht werden. Die Gelenkpositionen des simulierten Roboters haben eine durchschnittliche Abweichung von ca. einem Grad zu dem realen System und die durchschnittliche Geschwindigkeit des simulierten Systems weicht um ca. 7 mm/s ab, was vier Prozent der realen Geschwindigkeit entspricht. An dem Geschwindigkeitsgraphen sieht man allerdings, dass sich die Form der Kurven im Detail unterscheiden. Diese Unterschiede resultieren aus mechanischen Verbindungen die auf Grund der Bauweise Elastizitäten und somit Schwingungen in das Gesamtsystem bringen. Um diese Schwingungen simulieren zu können müssten Modelle für verformbare Strukturen erstellt werden. Ein weiterer Unterschied folgt aus der ungewollten Eigenschaft der Federn in den Unterbeinen, die bei einer seitlichen Belastung leicht verkanten und damit ihre Federeigenschaften zeitweilig verlieren. Trotz dieser Umstände erlaubt die Simulation eine Vorhersage bezüglich der Passierbarkeit einer Umgebung und dem zu erwartenden Gesamtstromverbrauch. Im Fall von dem Stromverbrauch liegen leider keine vergleichbaren Strommessungen zu den einzelnen Gelenken des realen Systems vor, weshalb nur der Stromverbrauch des gesamten Systems mit der Simulation verglichen werden kann. Auch die Form der simulierten Stromverbrauchskurve unterscheidet sich von der realen Messung. In drei von den vier Fällen passt aber der Mittelwert und die Ampli-



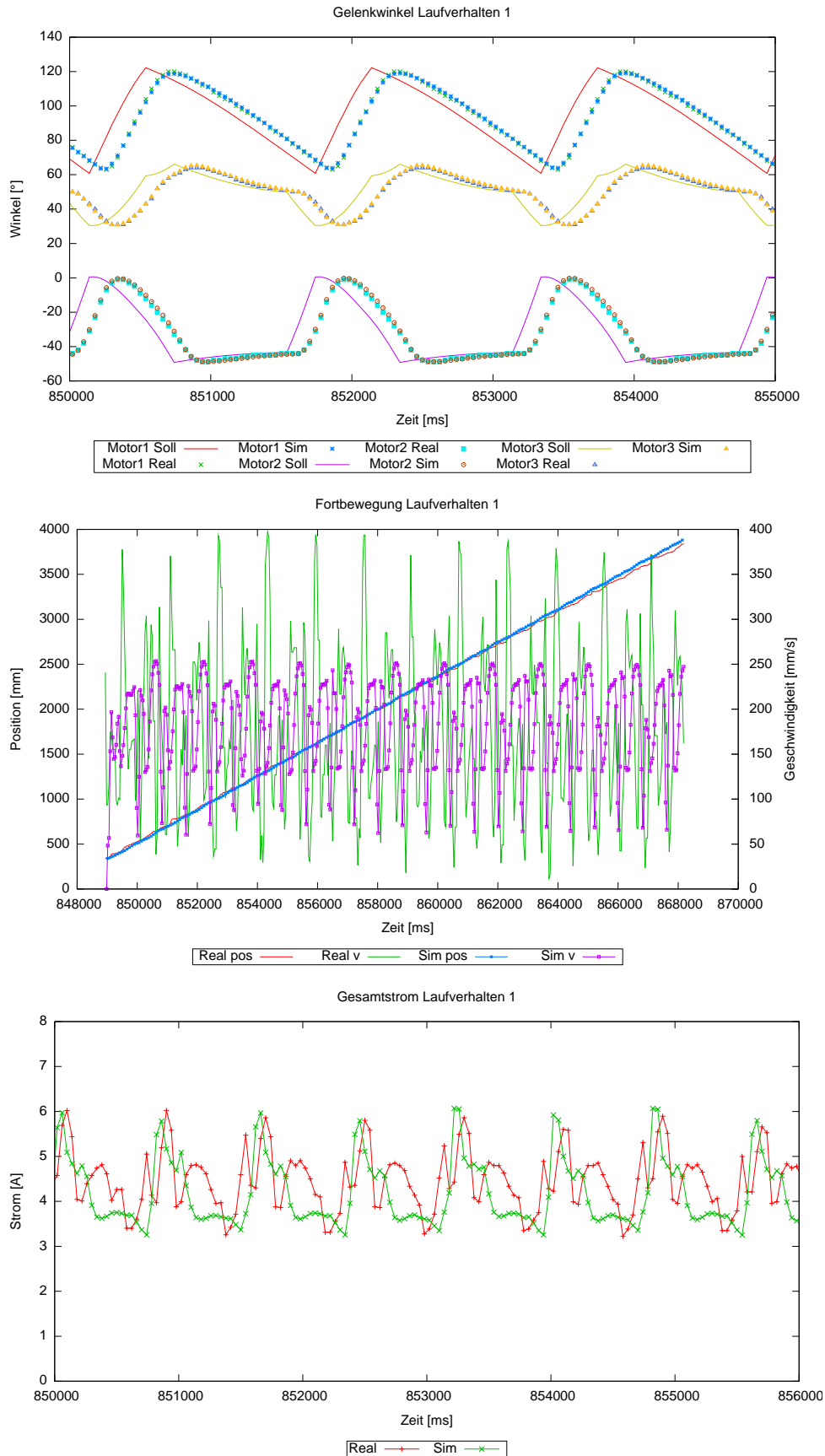
tude der Kurve gut zu den realen Daten. Im Fall des zweiten Laufverhaltens weicht die Amplitude ab. Woher diese Abweichung stammt konnte aus zeitlichen Gründen im Rahmen von VirtualCrater nicht weiter untersucht werden.

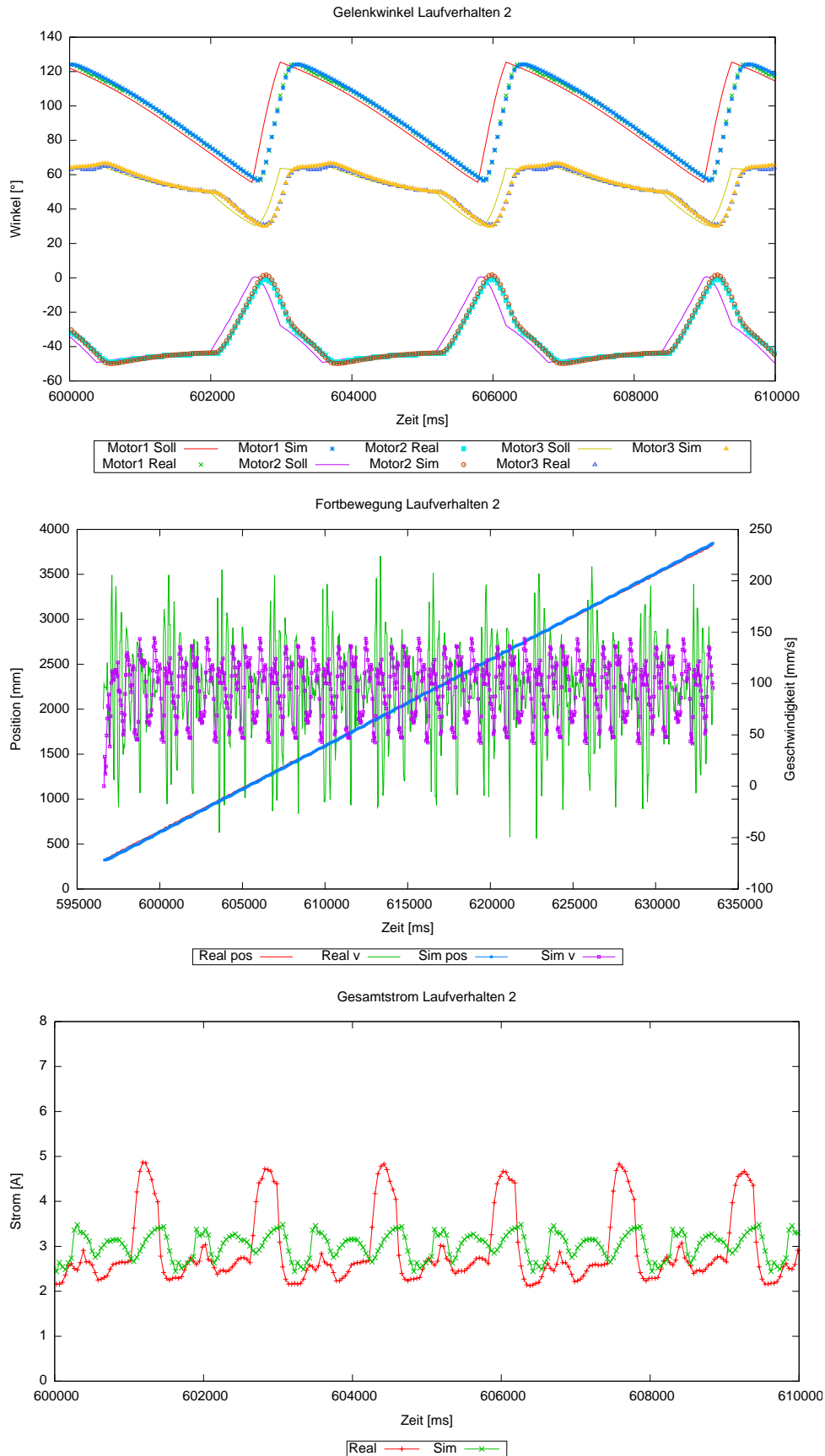
Bezüglich der Bodenmechanik konnte kein Vergleich des realen Gesamtsystems mit der Simulation gemacht werden. Wie in AP5300 beschrieben bildet das Simulationsmodell die aufgenommenen Referenzdaten einzelner Fußkontakte aus dem Bodenmechanik-Testbed gut ab, allerdings repräsentieren diese Referenzdaten wahrscheinlich aufgrund von Unzulänglichkeiten des Bodenmechanik-Testbeds nicht ausreichend das tatsächliche Verhalten der Bein-Bodeninteraktion des gesamten Roboters. Es ist anzunehmen, dass die unterschiedlichen Methoden zur Bodensimulation unter Herannahme besserer Referenzdaten auch gute Resultate für die Gesamtrobotersimulation liefern können. Im Rahmen von VirtualCrater wurden mehrere Paper über die Ansätze publiziert und die Arbeiten wurden mit großem Interesse von der Terramechanics-Community verfolgt. Ein Feedback auf die letzten Publikation in diesem Bereich war zudem eine Einladung, die Ergebnisse aus VirtualCrater in einer Journal-Publikation zu veröffentlichen.

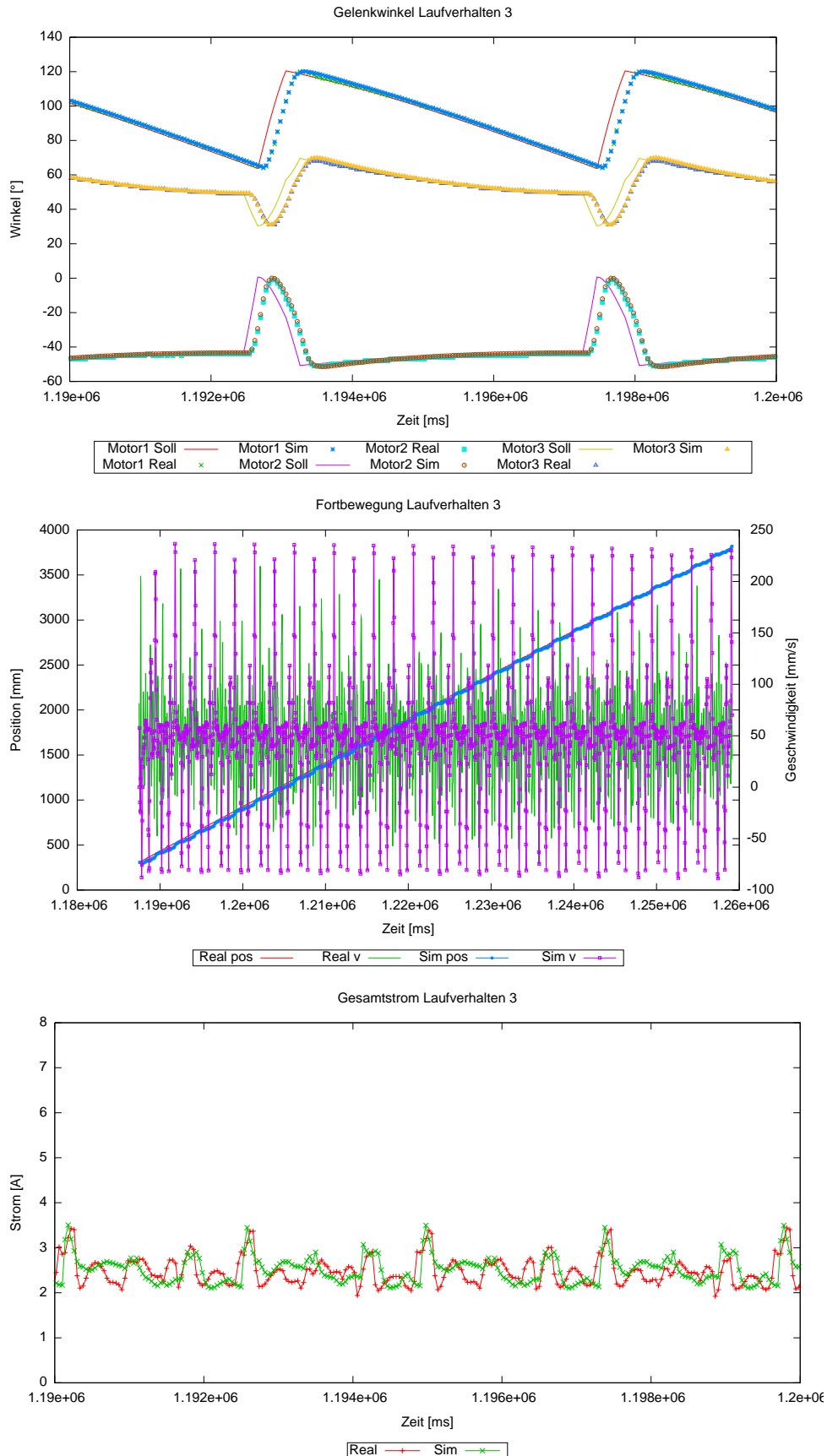
Auch das Untersuchen von verschiedenen Lösungen hat sich als sinnvoll erwiesen. So konnte gezeigt werden, dass der Partikel basierte Ansatz eine genauere Simulation der Interaktion ermöglicht aber nicht performant genug ist um in einer Echtzeit-Simulation eingesetzt zu werden. Der Ansatz über die neuronalen Netze liefert auch schon eine gute Approximation der Interaktion und hat zudem einen sehr geringen Rechenaufwand. Eine Kombination beider Ansätze wäre denkbar. Dadurch, dass bei dem Partikel basierten Ansatz analytische Methoden auf Basis der Bekker Theorie verwendet werden, könnte über eine Weiterentwicklung und Verifikation der Methoden ein Modell entstehen, welches in der Lage ist eine Bodeninteraktion ohne reale Referenzdaten hervorzusagen. Dieses Modell könnte dann verwendet werden um eine Abbildung über die neuronalen Netze zu erstellen und die Interaktion auch in einer Echtzeit-Simulation zu Verfügung zu stellen.

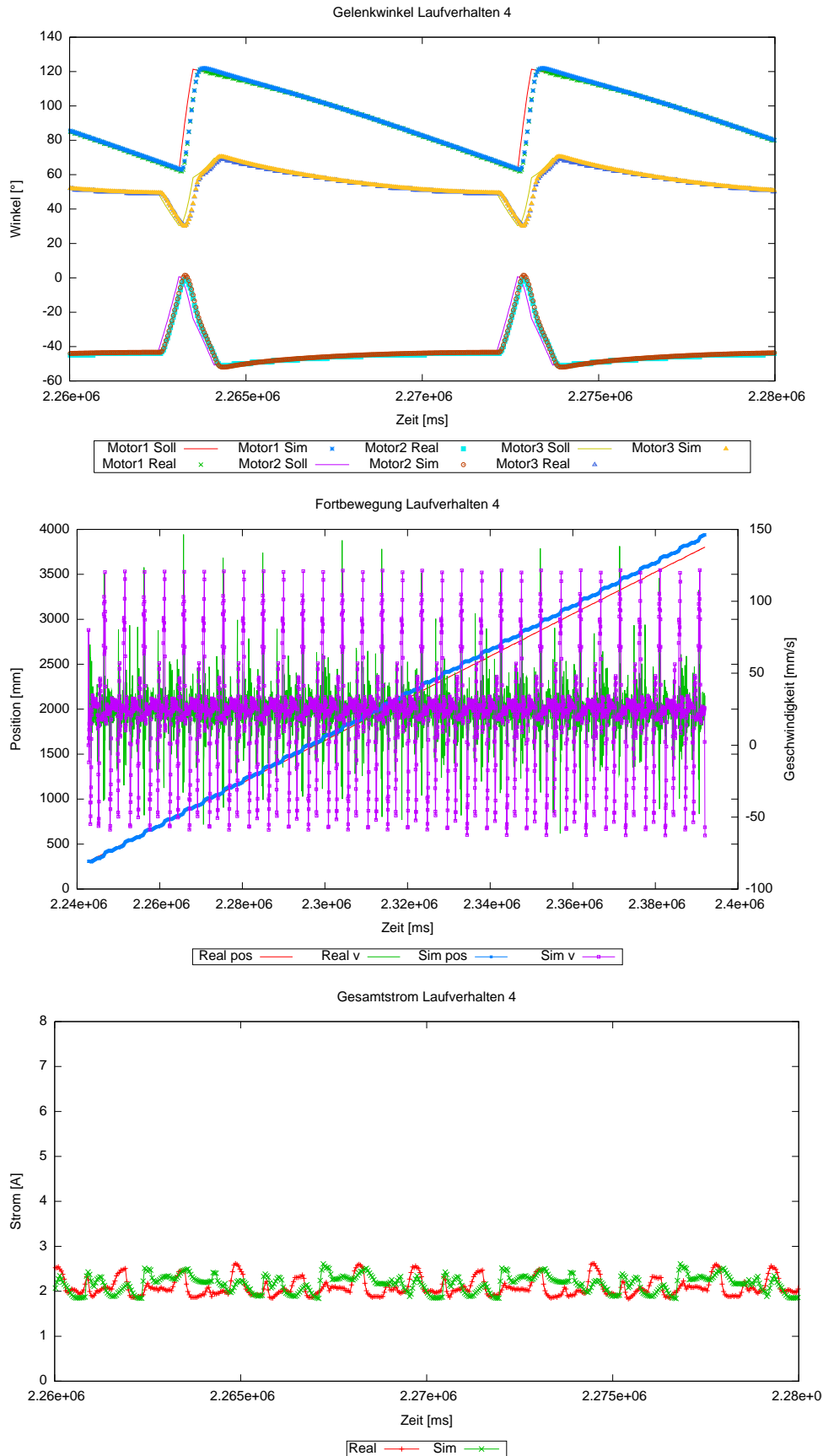
In der virtuellen Referenzmission wertet der SpaceClimber kaum seine integrierte Sensorik. Dies liegt vor allem daran, dass der echte SpaceClimber selbst diese Sensoren nicht unbedingt auswertet. Ein anderer Grund ist, dass die vorgesehene Sensorik entweder nicht, oder nicht in der ursprünglich geplanten Version eingesetzt wird. So war zu Beginn des Projektes ein anderer Laserscanner geplant. Da aber im Laufe des SpaceClimber Projektes ein neueres Modell verfügbar war, wurde dieses genutzt. Zu dem Zeitpunkt waren aber schon die Sensor-Experimente mit einem älteren Modell abgeschlossen und die Zeitplanung gab keinen Raum, um auch noch Experimente mit dem neuen Sensor durchzuführen.

Nichtsdestotrotz konnten die Sensoren in der Simulation nachgebildet werden und mit Hilfe des Fehlermodells die simulierten Werte in gewissen Bereichen an die realen Messungen angeglichen werden. Dies funktioniert zumindest für ein relatives bzw. absolutes Rauschen auf den Messwerten gut. Systematische Fehler oder Merkwürdigkeiten in den Sensoren können jedoch von diesen Modellen zur Zeit nicht abgedeckt werden. Hierzu wären in erster Linie weitere und deutlich umfangreichere Experimente mit den echten Sensoren notwendig, die aber sowohl zeitlich als auch vom Material-











aufwand her im Rahmen des Projektes nicht möglich waren. Als Beispiel sei hier der Laserscanner genannt. Hier wäre ein aufwändiger Aufbau zur Vermessung des Einflusses des Winkels interessant gewesen, mit dem Laser auf eine Oberfläche trifft. Eine solche Messung war mit vorhanden Mitteln nicht möglich. Es konnte auch kein Experiment durchgeführt werden, welches untersucht, ob es einen Unterschied zwischen dem Austrittswinkels des Lasers und den Messungen gibt. Es wurde in den Experimenten nur der "mittlere" Messpunkt genutzt und nicht die möglichen 240° .

Die Sensorexperimente haben gezeigt, dass das Sensor-Fehler-Modell in VEROSIM[®] in der Lage ist, den größten Teil an auftretenden Fehler, das Rauschen, abzudecken, und das damit eine Annäherung an reale Messergebnisse möglich ist. Es wurde aber auch deutlich, dass für eine vollständige Modellierung eines Sensors und seiner typischen Fehler (Laserscanner liefert ab einer bestimmten Entfernung sporadisch Fehlercodes) deutlich mehr nötig ist. Es zeigte sich aber auch, dass nicht unbedingt jeder Fehler bzw. jedes Verhalten simuliert werden muss. Sofern ein Fehler sehr genau in der Software erkannt wird und dieser keinen Einfluss auf das Verhalten hat, so muss dieser auch nicht zwangsläufig simuliert werden. Dies hängt jedoch stark von dem Einsatzzweck ab.

Ein weitere interessanter Effekt ist beim Force-Torque-Sensor aufgetreten. Dieser Sensor konnte sehr gut in der Simulation auch in seinem Fehlermodell nachgebildet werden. Jedoch wurden nur die Fehler nachgebildet, die bei den Referenzexperimenten auch aufgetreten sind. Wird nun aber der simulierte Sensor im Kontext des gesamten Roboters eingesetzt, liefert der simulierte Sensor immer noch die gleiche Güte an Daten, während ein viel stärkeres nicht gleich verteiltes Rauschen in dem echten Roboter auftritt. Dies ist auf die Nähe des Sensors zu dem Elektromotoren zurück zu führen. Diese streuen Störungen in die Messleitungen des Sensors ein. Hieraus können zwei Erkenntnisse gezogen werden. Zum einen können solche Vergleiche auf Probleme bei den echten Sensoren bzw. bei ihrem Einsatz hinweisen. Zum anderen reicht es nicht aus, den Sensor an sich gut simulieren zu können. Es ist auch erforderlich mögliche Störquellen bzw. Störgrößen herauszufinden und diese mit in ein Fehlermodell einzubeziehen, um ein Sensormodell auch in verschiedenen Kontexten einsetzen zu können.



3.2 Die wichtigsten Positionen des zahlenmäßigen Nachweises

Position		Gesamtvorkalkulation (€)	Gesamtnachkalkulation (€)
Material	0813	- €	- €
FE-Fremdleistungen	0823	782.652,00 €	782.651,64 €
Summe Personalkosten	0837	591.307,00 €	656.846,21 €
Reisekosten	0838	- €	11.393,78 €
Abschreibungen auf vorhabenspezifische Anlagen	0847	- €	4.076,84 €
Abschreibungen auf sonstige Anlagen	0848	- €	- €
Sonstige unmittelbare Vorhabenkosten	0850	- €	- €
Innerbetriebliche Leistungen	0856	- €	- €
Verwaltungskosten	0860	357.741,00 €	331.750,51 €
gesamte Selbstkosten des Vorhabens		1.731.700,00 €	1.786.718,99 €

Tabelle 19: Die wichtigsten Positionen des zahlenmäßigen Nachweises

3.3 Nutzen und Verwertungsplan

Das in Virtual Crater entwickelte Simulationssystem ist grundsätzlich so modular ausgelegt, das verschiedenste autonome Systeme und auch Umgebungen simuliert und immersiv dargestellt werden können.

Die Simulation und immersive Darstellung von Umgebungen ist für potentielle Industriekunden bereits eine wichtige Anwendung. Dazu können z.B. Passagierkabinen im Flugzeugbau, Gebäude in der Architekturplanung in Außen- und Innenansicht sowie allgemein jede Form von Aufbauten gehören. So könnten Kunden Bauplanungen und Konstruktionen verschiedenster Art firmenintern oder eigenen Kunden präsentieren, so dass diese realitätsnah evaluiert werden können, ohne diese kostenintensiv real aufbauen zu müssen. Das Simulationssystem aus Virtual Crater kann durch Erhöhung der Bandbreite an Modellimport-Fähigkeiten und durch eine Erhöhung der modularen Flexibilität in der Definition funktionaler Komponenten grundsätzlich in die Lage versetzt werden, entsprechende virtuelle Umgebungen für Industriekunden umzusetzen, immersiv darzustellen und diese auch funktional zu evaluieren. Durch die hohe Anzahl verschiedener CAD- und Planungswerkzeuge und entsprechender Dateiformate ist es bisher jedoch noch nicht möglich, beliebige Modelle automatisch in das Simulationssystem von Virtual Crater zu übernehmen, zumal je nach Komplexität des Ausgangsmaterials auch eine Vereinfachung der Modelle erforderlich sein kann, um die Echtzeitfähigkeit zu behalten.

Eine weitergehende Herausforderung ist die Simulation von mechanischen Konstruktionen, die ggf. auch autonom oder durch einen Bediener gesteuert sein können. Neben weiteren Weltraumanwendungen wie z.B. robotischen orbitalen Manövern wären z.B. auch Anwendungen aus der Automobilindustrie denkbar wie die interaktive Evaluation virtueller Fahrzeugprototypen. Auch Schulungen von Personal in der Bedienung solcher Systeme wären grundsätzlich denkbar. Hierzu wäre jedoch eine Erhöhung der



Flexibilität und Vereinfachung der Definition von Objektfunktionen erforderlich, um Kundenaufträge effizient und zeitnah umsetzen zu können bzw. deren virtuelle Modelle möglichst direkt übernehmen zu können. Um die Bodenmechanik radgetriebener Systeme zu simulieren wäre ferner eine entsprechende Erweiterung der Simulationskomponenten für die Bein-Boden-Interaktion erforderlich.

Das in Virtual Crater entwickelte Simulationssystem stellt eine sehr gute Grundlage zur Unterstützung der Entwicklung von Laufrobotern dar und kann somit auch in Zukunft für die Entwicklung neuer Systeme eingesetzt werden.

Insbesondere im Bereich der Weltraumrobotik sind simulative Tests von Roboter-Prototypen unerlässlich, weil so teure Entwicklungszyklen mit real aufgebauten Test-Systemen eingespart werden können und weil die Umgebungen, in denen die Roboter eingesetzt werden sollen, auf der Erde nicht vollständig nachgebildet werden können. So kann einerseits z.B. die von der Erdgravitation abweichende Gravitation auf anderen Planeten und Monden mit dem System aus Virtual Crater simuliert werden, gleichzeitig mit der entsprechenden Bodenmechanik der im Zielgebiet zu erwartenden Bodenarten. Ferner ist in einer solchen Simulation der Einsatz von adaptiven maschinellen Lernverfahren zur Systemdesign-Optimierung möglich, wobei eine möglichst schnelle Berechnung der Simulation in Echtzeit oder sogar schneller wünschenswert ist, um die Verfahren schneller als in der Realität möglich ablaufen zu lassen. Selbst einige der Bodenmechanik-Simulationsmethoden aus Virtual Crater sind dazu grundsätzlich in der Lage.

Die immersive Projektionsumgebung kann zusammen mit dem Simulationssystem in weiteren wissenschaftlichen Projekten eingesetzt werden. So ist in den geplanten Projekten RTES und TransTerra der Einsatz des Simulationssystems geplant. In RTES sollen mögliche Capture Systeme im Bereich Space Debris Removal mit dem System simuliert werden, ggf. direkt verbunden mit der Hardware-in-the-Loop Rendez-Vous-and-Docking-Simulationsanlage aus dem abgeschlossenen Projekt INVERITAS, das am DFKI unter der Konsortialführung von Astrium entwickelt wurde. In TransTerra ist geplant, die immersive Darstellung sowie die Interaktionsmöglichkeiten der stereoskopischen Mehrschirmpjektionsumgebung als Leitstand für die intuitive Kontrolle von Robotern einzusetzen.

In Folgeprojekten würden sich zahlreiche mögliche Weiterentwicklungen ergeben. Wichtig wäre eine genauere Untersuchung der lateralen (seitlichen) Kräfte bei der Bein-Boden-Interaktion sowie der Effekte von dynamischen Bodenkontakten mit hohen oder sehr veränderlichen Geschwindigkeiten, um das Verhalten von Laufrobotern noch präziser vorhersagen zu können.

Es wäre ein weiterer wichtiger Punkt, die Effekte der Temperaturentwicklung in den Robotern zu simulieren. So zeigte sich in Virtual Crater bereits, dass die Eigenerwärmung von Aktuatoren signifikante Auswirkungen auf deren Verhalten zu haben scheinen. Die Wärmeabfuhr im Weltraum oder auf atmosphärelosen Himmelskörpern ist ein generelles Problem für Roboter, die in solchen Umgebungen eingesetzt werden sollen. Die Formen von Eigenerwärmung, Wärmeabfuhr (z.B. durch Wärmestrahlung) sowie Wärmezufuhr (z.B. durch die Sonne) und ihre Auswirkungen auf die Systemkomponenten eines Roboters zu untersuchen und Simulationsmodelle dafür zu entwickeln würde die



Simulation von robotischen Systemen durch diesen wichtigen Aspekt erweitern und daher das Virtual Prototyping und seine Aussagekraft signifikant verbessern.

Die Modularität des Simulationssystems ließe sich ebenso erhöhen, so dass eine Annäherung an das Ziel möglich würde, Roboter bei laufender Simulation modular aus Einzelkomponenten zusammensetzen, zu modifizieren und zu evaluieren.

Solche Erweiterungen würden auch die wirtschaftliche Attraktivität des Systems erhöhen, weil industrielle Systementwicklungen sowohl für terrestrischen als auch für extraterrestrischen Einsatz durch noch genauere Vorhersagen der Simulationen sowie effizientere Systementwicklungsabläufe und durch die sich daraus ergebenden Zeit- und Kostenersparnisse von dem Simulationssystem profitieren könnten.

Im Folgenden werden tabellarisch Verwertungsmöglichkeiten des Projekts "Virtual Crater - Entwicklung einer virtuellen Simulations- und Demonstrationsumgebung zur planetarischen Exploration mit Fokus auf extraterrestrische Krater" dargestellt.

lfd.	Bezeichnung	Zeithorizont
1	Einsatz des Systems im Rahmen des geplanten Projekts TransTerra als Leitstand zur intuitiven Kontrolle von Robotern	kurzfristig
2	Nutzung des Systems zur Simulation von Capture-Technologien zur Beseitigung von Space Debris im Rahmen des geplanten Projekts RTES	mittelfristig
3	Weiterentwicklung des Systems in Folgeprojekten, insbesondere im Hinblick auf Bodenmechanik, Temperaturdynamik und interaktive Konstruktion robotischer Systeme	mittelfristig
4	Einsatz des Systems für Anwendungen die über den Space-Bereich hinausgehen wie z.B. Entwicklung von Fahrzeug-Prototypen	kurzfristig und langfristig

3.4 Forschungsergebnisse Dritter

Stand der Technik heute

VR-Umgebungen

Ein Problem von VR-Umgebungen ist ihre räumliche Beschränktheit. Einem Betrachter bzw. Nutzer einer solchen Umgebung sind durch die Projektionsflächen Grenzen in seiner Bewegung gesetzt. Er kann sich nicht frei bewegen und wird so immer wieder daran erinnert, dass er sich in einer virtuellen Umgebung befindet. Um diesen Umstand zu ändern wird versucht, dem Nutzer einen nach Möglichkeit grenzenlosen Bewegungsraum zu geben. Ein Ansatz hierfür ist ein omnidirektionales Laufband. Hierbei befindet



sich der Nutzer der VR-Umgebung auf einem Laufband, dass sich gleichzeitig in zwei Richtungen bewegen kann. Dies wird kombiniert mit einem Tracking des Nutzers. Das Laufband wird dann so angesteuert, dass der Nutzer sich nur wenig aus dem Zentrum des Laufbandes bewegt. Ein anderer Ansatz, um die räumliche Beschränkung aufzuheben ist es den Nutzer in eine Kugel zu platzieren. Der Nutzer läuft hierbei in einer Kugel, ähnlich wie ein Hamster in einer Lauf-Kugel. Zur Darstellung der Umgebung trägt der Nutzer ein Head-Mounted-Display, dass über Funk angesteuert wird. Die Bewegungen der Kugel werden zur Steuerung der Kamera in der virtuellen Welt eingesetzt. Der Nachteil der beiden genannten Lösungen ist, dass nur ein Nutzer sie zur Zeit nutzen kann. Es müsste zur kooperativen Nutzung jedem Nutzer eine solche Kugel oder ein Laufband zur Verfügung gestellt werden.

Das Thema Interaktion wird zur Zeit sehr stark durch die Kinect von Microsoft geprägt. Die Kinect ist zwar nicht der erste Sensor dieser Art, aber durch ihren sehr niedrigen Preis ist sie für einen sehr großen Personenkreis erschwinglich. Die Kinect bzw. ein ähnlicher Sensor könnte in Zukunft das heute sehr aufwändig zu installierende und in der Anschaffung teure Tracking, sei es nun optisch oder anders umgesetzt, ersetzen. Für typische 4-Wand CAVE-Aufbauten wurde dies in [27] beschrieben. Jedoch reicht die Genauigkeit der Kinect noch nicht aus, um die Bewegung des Kopfes mit ihr so genau umzusetzen, wie dies mit optischen Tracking-Systemen möglich ist. Zudem ist das Volumen, welches von der Kinect erfasst werden kann, noch zu klein für größere Installationen, wie sie am DFKI RIC zu finden ist. Hier zeigen aber Arbeiten wie [38] dass es möglich ist mehrere Kinects zu kombinieren. Dies kann zu verschiedenen Zwecken genutzt werden. Zum einen kann so ein größerer Raum abgedeckt werden, oder aber das Problem der Verdeckung reduziert werden. Interessant sind auch die Arbeiten, welche versuchen nicht nur die Arme, Beine und den Kopf zu erfassen, sondern einzelne Finger erkennen können. Dies funktioniert zur Zeit jedoch nur in einem sehr kleinen Bereich direkt vor der Kinect. Sollte es aber gelingen den Sensor selbst zu verbessern und die Methoden zur Erkennung der Finger, so könnte die Kinect auch Datenhandschuhe ersetzen. So würde auch das Problem der Kalibrierung der Handschuhe auf die jeweilige Hand entfallen.

Physik-Simulation

Es zeigt sich schon seit einigen Jahren, dass sich die Taktraten von Prozessoren nur mit erheblichen Problemen und auf Kosten einer immer aufwendiger werdenden Kühlung steigern lassen. Um dennoch dem Nutzer mehr Leistung bieten zu können, werden mehrere Rechenkerne auf einen Prozessor gesetzt. Dies steigert aber nur die Ausführungsgeschwindigkeit von Software, die auch mehrere Rechenkerne nutzt. Es ist aber nicht jede Software gleich gut zu parallelisieren. Die Algorithmen einer Starrkörper-Physik-Simulation sind nur sehr schwer zu parallelisieren, da hierfür durchzuführende Berechnungen von einander unabhängig sein müssen. Sofern diese Berechnungen nicht unabhängig sind, müssen nach einer separaten Berechnung von unterschiedlichen Abschnitten diese Ergebnisse wieder miteinander verrechnet werden. Dies ist sehr aufwendig macht den Vorteil der parallelen Berechnung wieder zunichte.



Aus diesem Grunde konzentrieren sich die Arbeiten in diesem Bereich darauf unabhängige Bereiche der Physik-Simulation zu identifizieren. Bereiche sind unabhängig von einander, sofern es zwischen ihnen keine Kollisionen gibt. Es ist also wichtig die Kollisionsberechnung zu beschleunigen und dies am besten durch Parallelisierung zu erreichen.

Durch die Entwicklung im Bereich der Prozessoren hin zu immer mehr Kernen und durch die Entwicklung im Bereich der Grafikkarten weg von starren Pipelines hin zu einfachen programmierbaren Rechenkernen wird die Entwicklung in Richtung Parallelisierung noch weiter vorangetrieben. So wurde in [33] eine Starrkörper-Simulation komplett in einem Grafikprozessor umgesetzt. Die Ergebnisse zeigen, dass unter optimierten Bedingungen und bei steigender Objekt- bzw. Kollisionsanzahl die GPU Vorteile gegenüber einer CPU hat.

Lunar Surface Operations Simulator (LSOS) des NASA JPL:

LSOS [34] wurde aufbauend auf früheren Simulationssystemen am Jet Propulsion Laboratory (JPL) entwickelt. Die wichtigsten Simulationskomponenten in LSOS beziehen sich auf Umgebungssimulationen und Rover-Systeme, wodurch die Simulation lunarer Missionen ermöglicht wird. Neben der Mechaniksimulation z.B. für Fahrzeuge hat LSOS auch Modelle für missionskritische Aspekte wie Energie, Stauraum, Signalstärke der Kommunikation sowie für Temperatur. Zwei Langstreckentests wurden in LSOS simuliert, eine 3 km lange Strecke am Black Point Lava Flow in Arizona und eine 570km lange Strecke vom Shackleton Krater bis zum Malapert Mountain und zurück zum Shackleton Krater nahe des Südpols des Mondes.

3DROV: Planetare Simulationsumgebung der ESA für Rover:

3DROV [17] ist eine von der Abteilung „Automation and Robotics“ der ESA und Trasy Space entwickelte Software zur Simulation von planetarischen Roversystemen. Es stellt Modelle planetarischer Umgebungen, mechanische, elektrische und thermale Subsysteme der Rover, generische on-board Controller und ein Bodenstationsmodul bereit. Die Systemarchitektur beinhaltet ein Simulationsframework, die Kontrollstation, generische Controller, Rover s/s Block, Umgebungskomponenten und eine Visualisierungsumgebung. Das Simulationsframework stützt sich auf ESAs SIMSAT und sorgt für die korrekte Planung und Ausführung der Simulation. Die auf A-DREAMS [16] basierende Kontrollstation ermöglicht die Erstellung von Missionsszenarien über eine grafische 3D Umgebung. Diese können dann auf den Rover geladen werden, während die on-board Telemetriedaten des Rovers wiederum in der Kontrollstation dargestellt werden. Während des Betriebs werden alle Rover Operationen vom generischen Controller, der als SIMSAT Komponente umgesetzt wurde, kontrolliert. Der Rover s/s Block beinhaltet Modelle der Sensoren und der wissenschaftlichen Instrumente. Die Umgebungskomponente kümmert sich um Ephemeriden und Zeitmessung, um die Erzeugung von Gelände und Witterungsbedingungen und das Tracken der Roverposition. Schließlich dient die Visualisierungsumgebung als Echtzeitfrontend zu 3DROVs



Simulation.

Bodenmechanikmodell der Universität von Surrey:

Das Surrey Space Centre (SSC) der Universität von Surrey hat basierend auf [39] ein Bodenmechanikmodell für granulare Untergründe und Laufroboter entwickelt. Dieses Modell verwendet ein validiertes Modell für die Zugkraft basierend auf der Arbeit von Grisso, ein validiertes Kraftmodell das von Bekker vorgeschlagen wurde [6] sowie ein Schubmodell für den Boden das von Terzaghi entwickelt wurde [45]. Alle Experimente mit diesem Modell waren auf langsame einfache Bewegungsmuster beschränkt. Ein auf MATLAB basierendes Werkzeug, genannt Legged Performance and Traction Prediction Tool (LPTPT), beinhaltet das Bodenkontaktmodell. Ergebnisse bezüglich der Echtzeitfähigkeit des Systems liegen nicht vor. Es wurde jedoch eine Einführung gegeben in die Boden-Simulanten, die in dieser Studie verwendet wurden, sowie in den Prozess der eingesetzt wurde um deren mechanische Eigenschaften zu bestimmen.

SCM (Soft Soil Contact Model) des DLR:

Das Simulationsmodell "Soil Contact Model (SCM)" [28] wurde am DLR Institut für Robotik und Mechatronik entwickelt und stellt ein Interface zwischen der klassischen Bodenmechanik Theorie von Bekker und den Fähigkeiten eines Mehrkörperssystem-Simulators dar und erreicht damit eine allgemeine 3D Simulation von Bodenkontakten. Das System wird hauptsächlich zur Simulation von Rovern auf sandigen Oberflächen genutzt, es liegen keine Ergebnisse bezüglich der Echtzeitfähigkeit des Systems vor. Das SCM behandelt typische Phänomene der Bodenmechanik wie einen steigenden Rollwiderstand durch Erhebungen des Bodens vor dem Rad, laterale Führung innerhalb von Furchen, Varianzen des "Drawbar Pull" Effekts wenn Schlupf vorliegt und den Effekt von mehrfachen Kontakten der gleichen Stelle des Bodens mit Rädern, wenn diese in einer Linie hintereinander fahren. Das "Soil Contact Model (SCM)" ist in SIMPACK verfügbar und kann in allen Designphasen der Roverentwicklung eingesetzt werden.

Neue Ansätze der Aktuatorsimulation:

Es wird zunehmend wichtig Aktuatoren in ihrer Entwicklungsphase zusammen mit ihren Controllern in einer Simulation zu testen. In der letzten Zeit wurden verschiedene Simulationsmodelle zur Analyse von Aktuatoren vorgestellt. Diese Modelle basieren auf mathematischen Modellen in Form von Zustands-Raum-Gleichungen [14] [47], gewöhnlichen Differentialgleichungen [30] [15], Finite-Elemente-Methoden (zur Simulation der inneren Motorkomponenten) [22][29] sowie teilweise auch auf Fuzzy Logic [14][44]. Ein interessanter Ansatz nutzt gewöhnliche Differentialgleichungen (ODEs) als einen speziellen Fall von diskreter Ereignis-System-Spezifikation (DEVS) um ein System zu modellieren und zu simulieren indem atomare DEVS Modelle in C++ definiert werden



die dann hierarchisch miteinander verknüpft werden können um komplexere Systeme zu erstellen.

F. Bergero und E. Kofman [9] entwickelten ein Werkzeug das automatisch die verbundenen Modelle in einen C++-Code übersetzt der die Simulationen innerhalb von Scilab durchführt [12]. Dieses Modell wurde zur Modellierung und Simulation eines kleinen DC Motors und seines PI-Geschwindigkeits-Controllers eingesetzt.

Dieser Ansatz hat den Vorteil dass er für eine hybride Systemsimulation und Systemkontrolle eingesetzt werden kann, da Petri-Netze, Zustandsdiagramme, Event Graphen und zeitdiskrete Systeme als Sonderfälle der DEVS gesehen werden können. Dennoch erscheint der Ansatz für komplexere Systeme nur begrenzt echtzeitfähig zu sein (in einem DC-Motor-Modell könnte es höchstens eine PWM-Frequenz von weniger als 1,7 kHz simulieren). Es ist ferner schwierig und aufwendig alle Systemmodule direkt in C++ zu modellieren.

Um den modellierten Aktuator zu evaluieren und seine Controller zu optimieren bevor er aufgebaut wird und auch um die Lücke zwischen Simulation und Realität zu verringern ist es wichtig, verschiedene digitale Steuerungsmechanismen in dem System vollständig zu testen um Engpässe und Schwachstellen bezüglich der Interrupt-Latenz, Rechengeschwindigkeit, Speicherleistung, Interfaces und Kommunikation rechtzeitig vor dem tatsächlichen Systemzusammenbau zu finden. Die Methode der Hardware-in-the-Loop-Simulation (HIL) ist dazu ein mächtiges Werkzeug. Der modellierte Aktuator und sein Controller werden in Echtzeit simuliert und durch eine Menge von Eingängen und Ausgängen (IOs) mit dem Echtzeit-Simulator verbunden [23].

Solche Simulationen der Aktuatoren und ihrer Controller werden benötigt um die Entwicklungs- und Testphasen zu beschleunigen indem Echtzeitsimulationen eingesetzt werden bevor Tests mit realen physischen Prototypen durchgeführt werden, weshalb sie intensiv untersucht und genutzt werden [37][18][36][21]. Zu den leistungsstärksten Werkzeugen die für HIL-Simulationen eingesetzt werden zählen MATLAB/Simulink und dSpace [35][40].

3.5 Veröffentlichungen

Die Ergebnisse der Arbeiten im Rahmen des Projekts Virtual Crater wurden regelmäßig publiziert, sowohl durch das DFKI als auch durch die Projektpartner MMI/RIF. Es folgt in Tabelle 20 eine Auflistung der Veröffentlichungen des DFKI zum Projekt Virtual Crater.

Ferner wurde das Projekt Virtual Crater am 06.03.2012 auf der zweiten nationalen Konferenz zur Raumfahrt -Robotik in Berlin präsentiert.



2009
Y.-H. Yoo, L. Abdenebaoui, M. Rohn, M. Langosz and F. Kirchner, "Modelling and Simulation of Mobile Six-Legged Robot with Leg-Soil Contact", ISTVS-09, Bremen, Oct 2009
M. Langosz, D. Kuehn, F. Cordes, Y.-H. Yoo and F. Kirchner, "Concept Evaluation of Modeling Terrain Mechanics by a Neural Network", ISTVS-09, Bremen, Oct 2009.
L. Abdenebaoui, M. Langosz, Y.-H. Yoo and F. Kirchner, "Optimization of terrain-vehicle simulation using evolutionary algorithms", ISTVS-09, Bremen, Oct 2009
2010
Y.-H. Yoo, M. Ahmed, S. Bartsch and F. Kirchner, "Realistic Simulation of Extraterrestrial Legged Robot in Trade-off between Accuracy and Simulation Time", IECON-2010, Glendale, AZ, USA, November 7-10 2010.
Y.-H. Yoo, T. Jung, M. Langosz, M. Rast, F. Kirchner and J. Rossmann "Developing a Virtual Environment for Extraterrestrial Legged Robot with Focus on Lunar Crater Exploration", i-SAIRAS 2010, August 29 - September 1, 2010, Sapporo, Japan
M. Ahmed, Y.-H. Yoo and F. Kirchner, "A Cosimulation Framework for Design, Test and Parameter Optimization of Robotic Systems", ISR Robotik-2010, Munich, Germany, 07.6.-09.6.2010
M. Ahmed, Y.-H. Yoo, F. Kirchner, "Measurement and Control of the Contact Forces between Walking Robot legs and its Environment", ISTVS-10, 27.9.-30.9.2010, Sapporo, o.A., Sep/2010.
2011
M.Langosz, M. Ahmed, L. Quack and Y. Kassahun, "Modeling of Leg Soil Interaction using Genetic Algorithms", ISTVS-11, 18.9.-22.9.2011, Blacksburg, VA, o.A., Sep/2011.
M. Ahmed, L. Quack, M. Langosz, Y.-H. Yoo, "Development of a Real and Simulation Testbed for Legged Robot Soil Interaction", ISTVS-11, 18.9.-22.9.2011, Blacksburg, Virginia, o.A., Sep/2011.
2012
Y.-H. Yoo, L. Quack, "Towards a Real-Time Capable Realistic Soil Contact Model Using a Particle System", In The 2nd Joint International Conference on Multibody System Dynamics , (IMSD-12), 29.5.-01.6.2012, Stuttgart, o.A., Jun/2012.

Tabelle 20: Veröffentlichungen des DFKI zum Projekt Virtual Crater



Literaturverzeichnis

- [1] AHMED, Mohammed ; KIRCHNER, Frank: A Simulation Environment to Be Utilised in the Design and Test Process of the HEVs and EVs BLDC Drive and Its Control. In: *Proceedings of the 11th European Regional Conference of the International Society of Terrain-Vehicle Systems. 11th European Regional Conference of the International Society of Terrain-Vehicle Systems (ISTVS-09), October 5-8, Bremen, Germany, ISTVS'09, 10 2009*
- [2] ANITESCU, M. ; POTRA, F.A.: Formulating dynamic multi-rigid-body contact problems with friction as solvable linear complementarity problems. In: *Nonlinear Dynamics* 14 (1997), Nr. 3, S. 231–247
- [3] BARAFF, D.: Fast contact force computation for nonpenetrating rigid bodies. In: *Proceedings of the 21st annual conference on Computer graphics and interactive techniques ACM*, 1994, S. 23–34
- [4] BAUER, R. ; LEUNG, W. ; BARFOOT, T.: Development of a dynamic simulation tool for the ExoMars rover. In: *Proceedings of the 8th International Symposium on Artificial Intelligence, Robotics and Automation in Space Citeseer*, 2005
- [5] BAUER, R. ; LEUNG, W. ; BARFOOT, T.: Experimental and simulation results of wheel-soil interaction for planetary rovers. In: *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on IEEE*, 2005, S. 586–591
- [6] BEKKER, M.G.: Introduction to Terrain-Vehicle Systems. In: *University of Michigan Press, Ann Arbor* (1960)
- [7] BEKKER, M.G.: *Theory of land locomotion: the mechanics of vehicle mobility*. University of Michigan Press, 1962
- [8] BENDER, J.: *Impulsbasierte Dynamiksimulation von Mehrkörpersystemen in der virtuellen Realität*. Univ.-Verlag Karlsruhe, 2007
- [9] BERGERO, Federico ; KOFMAN, Ernesto: PowerDEVS: a tool for hybrid system modeling and real-time simulation. In: *SIMULATION* (2010). <http://dx.doi.org/10.1177/0037549710368029>. – DOI 10.1177/0037549710368029
- [10] BIESIADECKI, J.J. ; HENRIQUES, DA ; JAIN, A.: A reusable, real-time spacecraft dynamics simulator. In: *Digital Avionics Systems Conference, 1997. 16th DASC., AIAA/IEEE Bd. 2 IEEE*, 1997, S. 8–2
- [11] BROOKS, Rodney A.: Artificial Life and Real Robots. In: *Proceedings of the First European Conference on Artificial Life*, MIT Press, 1992, S. 3–10
- [12] CAMPBELL, Stephen L. ; CHANCELIER, Jean-Philippe ; NIKOUKHAH, Ramine: *Modeling and Simulation in Scilab, Scicos*. Springer, 2006
- [13] COTTLE, R.W. ; DANTZIG, G.B.: COMPLEMENTARY PIVOT THEORY OF MATHEMATICAL PROGRAMMING. / DTIC Document. 1967. – Forschungsbericht



- [14] CUNKAS, Mehmet ; AYDOGDU, Omer: REALIZATION OF FUZZY LOGIC CONTROLLED BRUSHLESS DC MOTOR DRIVES USING MATLAB/SIMULINK. In: *Mathematical and Computational Applications* 15 (2010), S. 218–229
- [15] DAMODHARAN, P. ; VASUDEVAN, K.: Sensorless Brushless DC Motor Drive Based on the Zero-Crossing Detection of Back Electromotive Force (EMF) From the Line Voltage Difference. In: *Energy Conversion, IEEE Transactions on* 25 (2010), sept., Nr. 3, S. 661 –668. <http://dx.doi.org/10.1109/TEC.2010.2041781>. – DOI 10.1109/TEC.2010.2041781. – ISSN 0885–8969
- [16] DIDOT, F. ; KAPELLOS, K. ; SCHIELLE, A.: A-DREAMS: An Advanced Ground Control Station for Teleoperation and Telemanipulation. In: *in 8th ESA Workshop on Advanced Space Technologies for Robotics and Automation (ASTRA'04), Noordwijk, The Netherlands* (2004)
- [17] ESA: 3DROV planetary rover system simulation environment. In: <http://www.esa.int>, 2009 (2009)
- [18] GRUBIC, S. ; AMLANG, B. ; SCHUMACHER, W. ; WENZEL, A.: Hardware-in-the-Loop Simulation for the Development of an Experimental Linear Drive. In: *Industrial Electronics, IEEE Transactions on* 57 (2010), april, Nr. 4, S. 1208 –1216. <http://dx.doi.org/10.1109/TIE.2009.2037656>. – DOI 10.1109/TIE.2009.2037656. – ISSN 0278–0046
- [19] HAHN, J.K.: Realistic animation of rigid bodies. In: *ACM SIGGRAPH Computer Graphics* Bd. 22 ACM, 1988, S. 299–308
- [20] HANSEN, N. ; OSTERMEIER, A.: Completely Derandomized Self-Adaptation in Evolution Strategies. In: *Evolutionary Computation* 9 (2001), Nr. 2, S. 159–195
- [21] HUANG, Sunan ; TAN, Kok K.: Hardware-in-the-Loop Simulation for the Development of an Experimental Linear Drive. In: *Industrial Electronics, IEEE Transactions on* 57 (2010), april, Nr. 4, S. 1167 –1174. <http://dx.doi.org/10.1109/TIE.2009.2038408>. – DOI 10.1109/TIE.2009.2038408. – ISSN 0278–0046
- [22] IONEL, D.M. ; POPESCU, M.: Finite-Element Surrogate Model for Electric Machines With Revolving Field – Application to IPM Motors. In: *Industry Applications, IEEE Transactions on* 46 (2010), nov.-dec., Nr. 6, S. 2424 –2433. <http://dx.doi.org/10.1109/TIA.2010.2073671>. – DOI 10.1109/TIA.2010.2073671. – ISSN 0093–9994
- [23] ISERMANN, R. ; SCHAFFNIT, J. ; SINSEL, S.: Hardware-in-the-loop simulation for the design and testing of engine-control systems. In: *Control Engineering Practice* 7 (1999), Nr. 5, 643 - 653. [http://dx.doi.org/10.1016/S0967-0661\(98\)00205-6](http://dx.doi.org/10.1016/S0967-0661(98)00205-6). – DOI 10.1016/S0967–0661(98)00205–6. – ISSN 0967–0661
- [24] JAIN, A. ; BALARAM, J. ; CAMERON, J. ; GUINEAU, J. ; LIM, C. ; POMERANTZ, M. ; SOHL, G.: Recent developments in the ROAMS planetary rover simulation environment. In: *Aerospace Conference, 2004. Proceedings. 2004 IEEE* Bd. 2 IEEE, 2004, S. 861–876



- [25] JAKOBI, Nick ; HUSBANDS, Phil ; HARVEY, Inman: Noise and The Reality Gap: The Use of Simulation in Evolutionary Robotics. In: *Advances in Artificial Life: Proc. 3rd European Conference on Artificial Life*, Springer-Verlag, 1995, S. 704–720
- [26] JANOSI, Z. ; HANAMOTO: Analytical determination of drawbar pull as a function of slip for tracked vehicles in deformable soils. In: *In Proc. the First International Conference on Terrain-Vehicle Systems, Edizioni Minerva Tecnica, Torino.*, 1961
- [27] JUNG, Thomas ; KROHN, Stephan ; SCHMIDT, Peter: Ein Natural User Interface zur Interaktion in einem Cave Automatic Virtual Environment basierend auf optischem Tracking. In: *Workshop 3D-NordOst*, 2011
- [28] KRENN, R. ; HIRZINGER, G.: SCM – A SOIL CONTACT MODEL FOR MULTI-BODY SYSTEM SIMULATIONS. In: *Proceedings of the 11th European Regional Conference of the ISTVS 2009I, October 5-8, Bremen, Germany* (2009)
- [29] MAHMOUDI, A. ; RAHIM, N. A. ; HEW, W. P.: Axial-flux permanent-magnet motor design for electric vehicle direct drive using sizing equation and finite element analysis. In: *Progress In Electromagnetics Research* 122 (2012), S. 467–496
- [30] MENG, Zhaojun ; CHEN, Rui ; SUN, Changzhi ; AN, Yuejun: The mathematical simulation model of brushless dc motor system. In: *Computer Application and System Modeling (ICCA SM), 2010 International Conference on Bd. 12*, 2010, S. V12–625 –V12–629
- [31] MICHAUD, S. ; RICHTER, L. ; PATEL, N. ; THÜER, T. ; HUELSING, T. ; JOUDRIER, L. ; SIEGWART, R. ; ELLERY, A.: RCET: rover chassis evaluation tools. In: *Proceedings of the 8th ESA Workshop on Advanced Space Technologies for Robotics and Automation*, 2004, S. 2–4
- [32] MIRTICH, B.V.: *Impulse-based dynamic simulation of rigid body systems*, University of California, Diss., 1996
- [33] MONTE, Álvaro ; TORRES, Roberto ; MARTÍN, Pedro J. ; GAVILANES, Antonio: A FULLY GPU-IMPLEMENTED RIGID BODY SIMULATOR. In: *GRAPP*, 2008, S. 342–350
- [34] NAYAR, H. ; JAIN, A. ; BALARAM, J. ; CAMERON, J. ; LIM, C. ; MUKHERJEE, R. ; POMERANTZ, M. ; REDER, L. ; MYINT, S. ; SERRANO, N. ; WALL, S.: Recent Developments on a Simulator for Lunar Surface Operations. In: *AIAA SPACE 2009 Conference and Exposition, Pasadena, CA, September 14-17*, 2009
- [35] POHL, L. ; VESELY, L.: PMSM control framework with dSPACE real-time control platform. In: *Intelligent Engineering Systems (INES), 2012 IEEE 16th International Conference on*, 2012, S. 179 –183
- [36] SALEEM, Ashraf ; ISSA, Rateb ; TUTUNJI, Tarek: Hardware-In-the-Loop for on-line identification and control of three-phase squirrel cage induction motors. In: *Simulation Modelling Practice and Theory* 18 (2010), Nr. 3, 277 - 290. <http://dx.doi.org/10.1016/j.simpat.2009.11.002>. – DOI 10.1016/j.simpat.2009.11.002. – ISSN 1569–190X



- [37] SARIKAN, A. ; AYDEMIR, M.T.: Real time digital simulation (RTDS) software and hardware in the loop (HIL) architecture for brushless DC motors. In: *MELECON 2010 - 2010 15th IEEE Mediterranean Electrotechnical Conference*, 2010, S. 779–783
- [38] SCHOLZ, Alexander ; BERGER, Kai ; RUHL, Kai: Multiple Kinect Studies Technical Report 2011-09-15. 2011. – Forschungsbericht
- [39] SCOTT, G. P. ; M., Saaj C.: The development of a soil trafficability model for legged vehicles on granular soils. In: *Journal of Terramechanics, Elsevier, June–August 2012* (2009)
- [40] SHAO, Xuejuan ; ZHANG, Jinggang ; ZHAO, Zhicheng: The position servo system based on dSPACE. In: *Control and Decision Conference (CCDC), 2012 24th Chinese*, 2012, S. 2515–2519
- [41] SOHL, G. ; JAIN, A.: Wheel-terrain contact modeling in the ROAMS planetary rover simulation. In: *Proc. Fifth ASME International Conference on Multibody Systems, Nonlinear Dynamics and Control Citeseer*, 2005
- [42] SPENNEBERG, Dirk ; ALBRECHT, M. ; BACKHAUS, Till: M.O.N.S.T.E.R.: A New Behavior-Based Microkernel for Mobile Robots. In: *Proceedings of the ECMR 2005.*, 2005
- [43] STEWART, D.E. ; TRINKLE, J.C.: AN IMPLICIT TIME-STEPPING SCHEME FOR RIGID BODY DYNAMICS WITH INELASTIC COLLISIONS AND COULOMB FRICTION. In: *International Journal for Numerical Methods in Engineering* 39 (1996), Nr. 15, S. 2673–2691
- [44] TERKI, A. ; MOUSSI, A. ; BETKA, A. ; TERKI, N.: An improved efficiency of fuzzy logic control of PMLDC for PV pumping system. In: *Applied Mathematical Modelling* 36 (2012), Nr. 3, 934 - 944. <http://dx.doi.org/10.1016/j.apm.2011.07.042>. – DOI 10.1016/j.apm.2011.07.042. – ISSN 0307–904X
- [45] TERZAGHI, K.: Theoretical Soil Mechanics. In: *3rd Ed., John Wiley and Sons, Inc., London* (1943)
- [46] WITKIN, A. ; GLEICHER, M. ; WELCH, W.: *Interactive dynamics*. ACM, 1990
- [47] XIN, Xiaonan ; LUO, Yangrui ; XU, Hexiu ; ZHANG, Huijuan: A new method of brushless DC motor control system simulation. In: *Artificial Intelligence, Management Science and Electronic Commerce (AIMSEC), 2011 2nd International Conference on*, 2011, S. 3667–3670
- [48] YEN, J. ; JAIN, A. ; BALARAM, J.: ROAMS: Rover analysis modeling and simulation. In: *Artificial Intelligence, Robotics and Automation in Space, Proceedings of the Fifth International Symposium, ISAIRAS'99, held 1-3 June, 1999 in ESTEC, Noordwijk, the Netherlands. Edited by M. Perry. ESA SP-440. Paris: European Space Agency, 1999., p. 249 Bd. 440, 1999, S. 249*



- [49] YOSHIDA, K. ; HAMANO, H.: Motion dynamics of a rover with slip-based traction model. In: *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on* Bd. 3 IEEE, 2002, S. 3155–3160