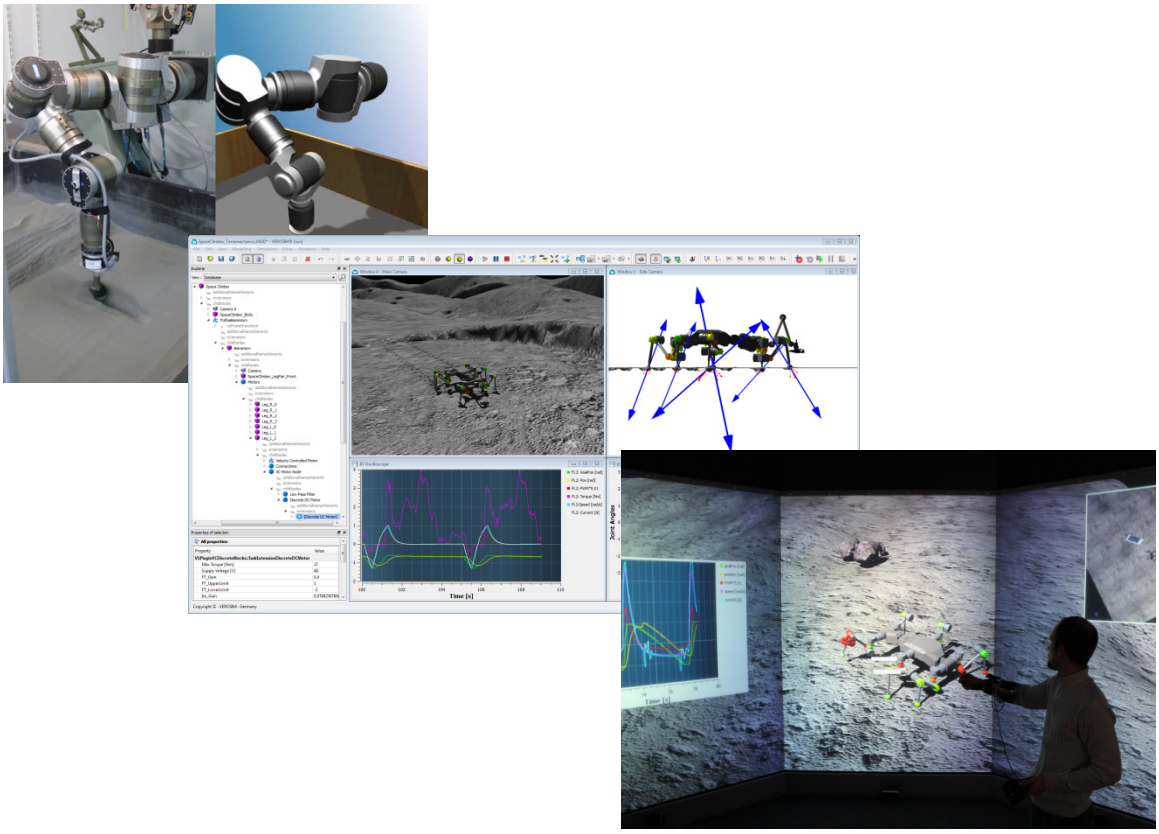
Virtual Crater

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

Schlussbericht

Laufzeit: 01.05.2009 – 31.08.2012

27.02.201301.03.2013



RIF Institut für Forschung und Transfer e.V.
Prof. Dr.-Ing. Jürgen Roßmann
Joseph-von-Fraunhofer Str. 20
D-44227 Dortmund

1 Inhaltsverzeichnis

2 Abbildungsverzeichnis

3 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 war der Abgleich einer virtuellen Testumgebung mit einer im DFKI im Rahmen des LUNARES Projektes (DLR Fkz: 50 RA 0706) aufgebauten realen, lunaren Kratertestumgebung. Durch die Realisierung der virtuellen Testumgebung „Virtual Crater“ in Ergänzung zur bestehenden realen Testumgebung ist es nun möglich, die projektierten Systeme zunächst virtuell zu planen, zu evaluieren, zu programmieren und zu testen, sie dann aber direkt in einer vergleichbaren Umgebung real weiter zu analysieren und zu evaluieren. Durch den ständigen Abgleich zwischen realer und virtueller Testumgebung entstanden gänzlich neue Möglichkeiten zur Erhöhung der Realitätsnähe der virtuellen 3D-Testumgebung. Diese Kombination von aufeinander abgestimmter realer und virtueller Testumgebung ist momentan einzigartig und ein wertvolles Werkzeug für die Entwicklung und Erprobung wie auch für die Demonstration neuer Konzepte für die planetarische Oberflächenexploration.

3.1 Reale und virtuelle Umgebung

Die Grundidee dieses Projektes bestand in der Bereitstellung einer neuartigen, umfassenden Testumgebung durch Ergänzung der vorhandenen **realen** Testumgebung durch eine simulationsgestützte **virtuelle** Testumgebung. Durch die Kombination einer virtuellen mit einer realen Testumgebung entstehen für den Entwicklungsprozess mobiler Robotikkomponenten völlig neue Möglichkeiten durch Nutzung unterschiedlicher Synergieeffekte:

- In der virtuellen Testumgebung können physikalische Effekte simuliert werden, die in einer realen Testumgebung **auf der Erde nicht nachgebildet** werden können (Gravitation, beschränkte Größe etc.).
- Eine 3D-Simulation bietet durch die Möglichkeit, z. B. Zwangskräfte im mechanischen Aufbau beobachten zu können, einen deutlich **tieferen Einblick** in das entwickelte System.
- Eine virtuelle Testumgebung kann auf jedem „normalen“ Arbeitsplatzrechner verwendet werden und steht somit jedem Entwickler als **persönliche Testumgebung** zur Verfügung.
- Durch vordefinierte, auf Knopfdruck abrufbare Testszenarien und die einfache Änderbarkeit des virtuellen Modells (Roboter, Steuerungssoftware, Umgebung etc.) fördert eine virtuelle Testumgebung deutlich **kürzere Entwicklungszyklen**.
- Durch eine offene Softwarestruktur können neue Sensoren oder Motoren schnell in das Simulationsmodell aufgenommen werden, so dass **unter normierten Bedingungen erste Vergleichstests** durchgeführt werden können.
- Auf der anderen Seite ist die Realitätsnähe realer Testumgebungen für eine Vielzahl von Effekten natürlich unübertroffen.
- Durch den direkten Vergleich einer normierten realen Testumgebung mit einer virtuellen Testumgebung können beiden Testumgebungen **wechselseitig optimiert und auf Plausibilität der Simulationsergebnisse überprüft** werden, was in der Summe ***schneller*** **deutlich bessere Entwicklungsergebnisse** bei deutlich **verringertem Hardware-Aufwand** erwarten lässt.
- Insbesondere wird hierdurch die **Validierung und iterative Weiterentwicklung** des virtuellen Modells möglich.

Am besten lässt sich die symbiotische Kombination von virtueller und realer Testumgebung an einem Beispiel veranschaulichen. Soll z. B. für einen Laufroboter ein neuer, energiesparender aber drehmomentärmerer Motor erprobt werden, stellt sich die Frage, wie hoch die Energieeinsparung unter lunaren Bedingungen tatsächlich ist und ob bestehende Laufstrategien mit dem geringeren Drehmoment auszuführen sind. Hierzu könnte im realen Testbed zunächst eine Drehmoment-

Leistungs-Kennlinie empirisch bestimmt werden. Anhand dieser Daten wird dann ein mathematisches Modell seines Verhaltens erzeugt und im Simulationssystem des virtuellen Testbeds implementiert. Hier können dann unterschiedliche Testszenarien „durchgespielt" werden und die Frage der Energieeinsparung auf Basis der Simulationsergebnisse beantwortet werden.

Neben dem Abgleich der virtuellen Testumgebung mit einer im Rahmen des LUNARES Projektes aufgebauten Kraterumgebung war eine anwendungsnahe Referenzmondmission ein Ziel dieses Projektes. Für dieses Ziel sollten Informationen aus Recherchen zu lunaren Kraterlandschaften herangezogen werden. Recherchen des DFKIs haben eine Auswahl von drei möglichen realen Mondkratern ergeben, in denen die meisten Anwendungs-Szenarien durchgeführt werden können. Die Daten zu einem dieser Mondkrater wurden verwendet um eine virtuelle Kraterlandschaft zu erstellen, die eine anwendungsnahe Referenzmondmission erlaubt.

Die virtuelle Umgebung wurde in den Räumen des DFKI verfügbar gemacht. Sie kann mehrfach instanziiert und damit – im Gegensatz zur realen Testumgebung – mehreren Anwendern gleichzeitig zur Verfügung gestellt werden. Die virtuelle Testumgebung kommt entsprechend sowohl bei den Entwicklern auf ihren „normalen" Arbeitsplatzrechnern für die parallele Entwicklung der Systeme sowie auf High-End-Simulations- und Visualisierungshardware für die weitergehende Analyse und Präsentation der Entwicklungsergebnisse zum Einsatz.

3.2 Anbindung der Robotersteuerung

Abbildung 1 stellt die reale und die virtuelle Testumgebung insbesondere vor dem Hintergrund der Integration der Robotersteuerung gegenüber. Hierbei ist es wichtig, 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.

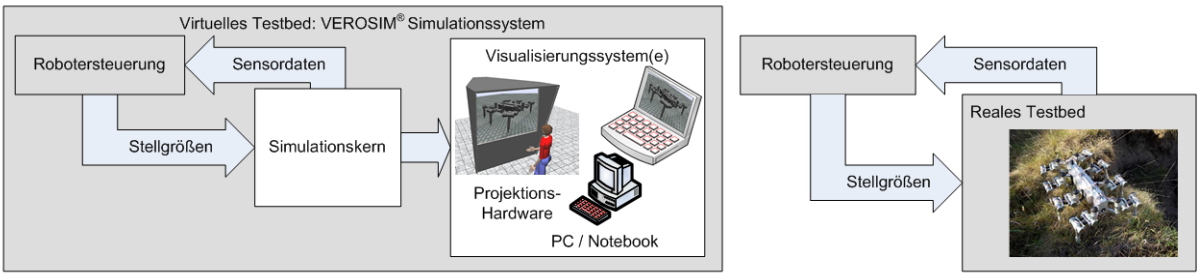


Abbildung 1: Anbindung einer Robotersteuerung an das reale und an das virtuelle Testbed

3.3 Testsystem-„Konfigurationen"

Das Simulationssystem selbst kann in unterschiedlichen „Konfigurationen" betrieben und somit in unterschiedlichen Anwendungsbereichen eingesetzt werden. Möglich sind u. a. folgende Anwendungsszenarien:

- Die virtuelle Testumgebung wird vollständig auf einem „normalen" **Arbeitsplatzrechner** als autarke Entwicklungsumgebung eingesetzt.
- Eingesetzt zusammen mit einer **Mehrschirm-Projektionsumgebung** (siehe Kapitel 3.4) können Testszenarien detaillierter evaluiert und analysiert werden.
- In einer Art **„Terminal-Struktur"** können besonders aufwändige Berechnungen auf besonders leistungsfähigen Rechnern ablaufen. Das Simulationsergebnis kann dann wiederum auf einem Arbeitsplatzrechner oder auf einer beliebigen Projektionsumgebung wiedergegeben werden.
- Zu Präsentationen oder zum Projektmarketing ist die Simulationsumgebung als **mobile virtuelle Testumgebung** auch auf einem Laptop lauffähig.

3.4 Stereoskopische Mehrschirm-Projektionsumgebung

Gerade die realitätsnahe stereoskopische Darstellung ermöglicht einen intuitiven Zugang zum simulierten System. So genannte Visualisierungsmetaphern geben auf Wunsch gleichzeitig detaillierte Einblicke in nicht direkt sichtbare Systemzustände. Eine Verbesserung des Eindrucks der „Immersion" in die virtuelle Welt ist möglich, wenn es gelingt, das Gesichtsfeld des Anwenders bei der Arbeit in der virtuellen Welt möglichst vollständig durch die künstlich erzeugte Umgebung abzudecken. Dies kann erreicht werden, indem die wirksame Fläche der Rückprojektionsbox derart vergrößert wird, dass das gesamte Blickfeld des Anwenders abgedeckt wird. Durch Mehrschirm-Rückprojektionssysteme kann eine vollständige Abdeckung des Gesichtsfeldes und damit eine optimale Stimulation des Sehapparates für den Anwender erreicht werden. Abbildung 2 illustriert in diesem Zusammenhang eine 5-Seiten Projektionsbox. Die im Projekt umgesetzte Installation basiert auf dieser Projektionstechnik.

Abbildung 2: Fahrsimulation in einer Mehrschirmprojektion

3.5 Weiterentwicklung der realen Testumgebung

Um zu einer möglichst realistischen Simulation des Roboterverhaltens zu gelangen, sollte ein Abgleich zwischen der realen und der virtuellen Umgebung durchgeführt werden.

Um auf den verschiedenen relevanten Untergründen physikalische Parameter, z.B. Reibungswerte empirisch zu ermitteln, sollten bzgl. des betrachteten Mechanismus einfache und in der virtuellen Testumgebung gut nachstellbare Testaufbauten erstellt werden (z.B. um ein in seinen physikalischen Parametern bekanntes Objekt mit einer definierten Kraft über den Untergrund zu ziehen und dies entsprechend in der virtuellen Testumgebung zu simulieren) um die nötigen Bodenparameter in der Realität zu messen und in die Simulation zu übertragen.

3.6 Validierung und Weiterentwicklung der virtuellen Testumgebung

Durch den ständigen Abgleich zwischen den aus der virtuellen Testumgebung gewonnenen Simulationsergebnissen mit dem realen Systemverhalten ist eine permanente Validierung sowie die iterative Weiterentwicklung sowohl des virtuellen Modells als auch des mechanischen Mockups möglich. Nachdem die Simulations- und Visualisierungshardware des DFKI in Betrieb genommen wurde sind in einem sukzessiven Verfahren die notwendigen 3D-Simulations- und -Visualisierungskomponenten eingesetzt worden. Zum Test wurden entsprechende empirische Daten, welche aus der realen Testumgebung anhand von Referenzexperimenten erhoben wurden, verwendet. Die Simulation wurde dabei solange verfeinert und parametrisiert bis die aus der Simulation gewonnenen Ergebnisse mit den im realen Testbed gewonnenen übereinstimmten.

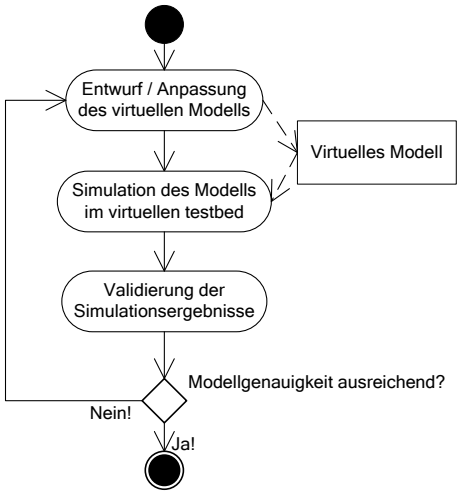


Abbildung 3: Das iterative Vorgehen zur Entwicklung und Validierung des Simulationsmodells

Zusätzlich sollten weitere **Simulationsmodelle für die in der realen Umgebung nicht simulierbaren Aspekte** erstellt und in die „Virtual Crater"-Simulation integriert werden. Um diese Modelle zu testen, sollten als Referenzsystem der zeitgleich entstehende Spaceclimber-Roboter sowie Systeme aus dem „LUNARES-Projekt" genutzt werden.

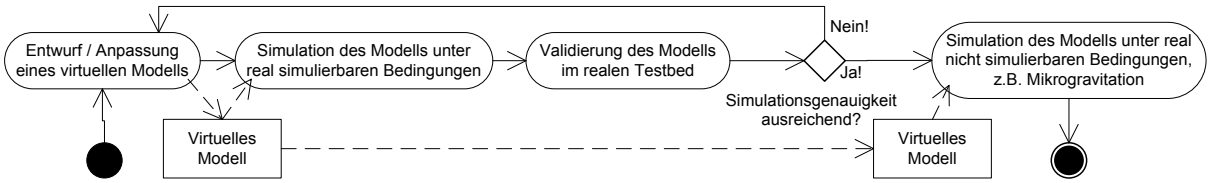


Abbildung 4: Eine möglichst weitgehende Parametrierung unter real simulierbaren Bedingungen ermöglicht eine deutlich verlässlichere Simulation des Systems unter nicht simulierbaren Bedingungen, z. B. Mikrogravitation

Neben der Bereitstellung optimaler Entwicklungswerkzeuge für das „LUNARES-Projekt" wurde durch den detaillierten Abgleich von realer und virtueller Testumgebung eine weitergehende „Kalibrierung" der eingesetzten Simulationskomponenten erzielt. Diese sollten dann eine ideale Basis auch für weitere Anwendungen (mobiler) Robotik im Weltraum, auf der Erde oder im Wasser bilden.

4 Voraussetzungen, unter denen das Vorhaben durchgeführt wurde

Das Projekt orientierte sich an den Bedürfnissen des DFKI hinsichtlich der Anforderungen an eine simulationsgestützte Entwicklungsumgebung für mobile Roboter. Um den Abgleich zwischen realen und virtuellen Testumgebungen durchführen zu können, legten insbesondere die Vorarbeiten des DFKI im Bereich realer Testumgebungen die Randbedingungen des Projekts fest.

4.1 Reale Testumgebung am DFKI

Das DFKI hat im Rahmen des Projektes LUNARES (Fkz: DLR 50RA076 & BIG INNO1036A) eine nachgebildete Mondlandschaft aufgebaut (Fertigstellung Dez. 2007), die der Demonstration rekonfigurierbarer Robotersysteme dient (siehe Abbildung 5 und Abbildung 6). Diese Umgebung kann insbesondere für Mobilitäts- und Funktionstests (z. B. von Funktionen wie Navigation, Lokalisation, Objekterkennung, Manipulation) verwendet werden. Bei der aktuellen Gestaltung liegt der Fokus auf einer **realistischen Landschafts- und Bodenmechaniknachbildung** sowie auf **kontrollierbaren Lichtverhältnissen**. Mittels einer vom DFKI beigestellten Portalkrananlage können

zudem Experimente mit **Gewichtsreduktion** durchgeführt werden. Ein **High-Speed-Tracking-System** mit drei Infrarotkameras erlaubt zusätzlich die Gewinnung genauester Trajektoriendaten aller in dieser Umgebung sich bewegenden Objekte.

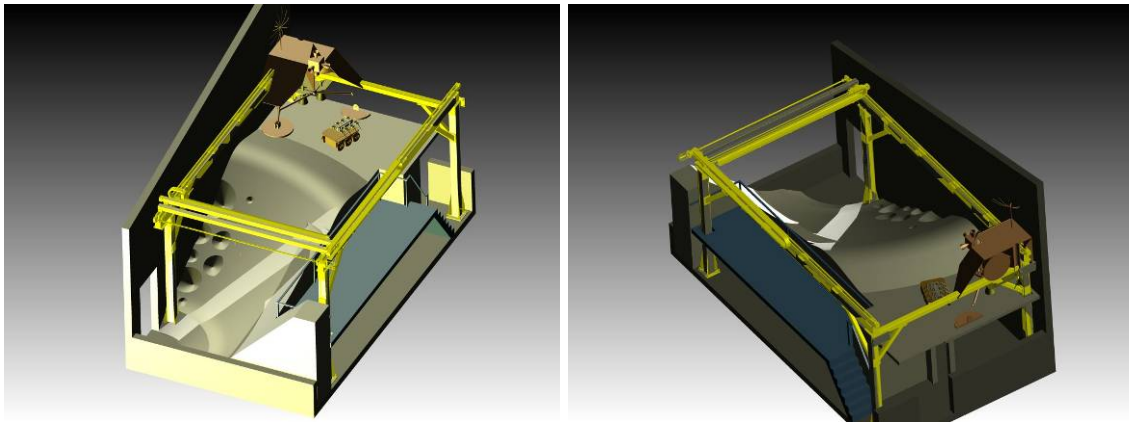


Abbildung 5: Ansichten der LUNARES Umgebung mit künstlichem Krater und unterschiedlichen Gefälle mit OHB-Lander-Mock Up, EADS-Astrium Rover und DFKI Scorpion Roboter sowie dem bereits installierten Flächenportal

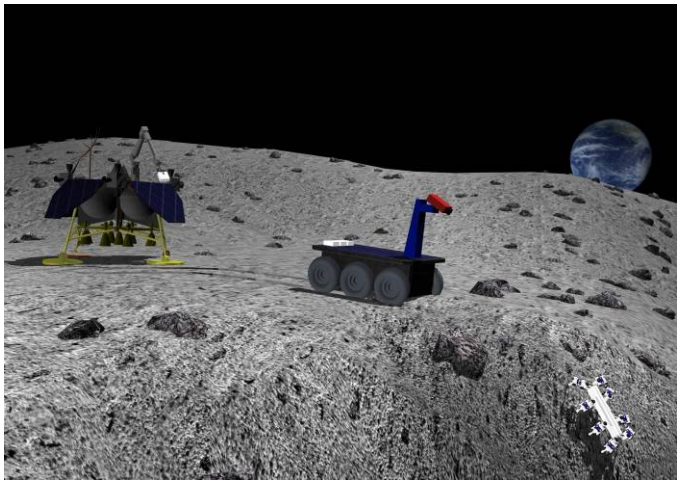


Abbildung 6: LUNARES Szenario

Aufgrund der begrenzten Fläche von 40qm für die LUNARES-Testumgebung plante das DFKI zum Zeitpunkt des Projektbeginns bereits eine weitere, größere Halle zu errichten, welche über eine Fläche von 250qm und eine Raumhöhe von ca. 9m verfügen sollte. Erste Darstellungen dieser Halle sind in Abbildung 7 gezeigt. Dieser Plan ist mittlerweile umgesetzt worden.

In diesen Umgebungen sollte es möglich sein, komplexe Missionen sehr realistisch zu erproben und das Verhalten von Robotern auf Untergründen, die denen auf extraterrestrischen Körpern (hier vornehmlich Mars und Mond) ähneln unter **kontrollierten Bedingungen** zu testen. Darüber hinaus sollten diese Anlagen auch für den Test terrestrischer Robotersysteme (z.B. Rettungsrobotersysteme) eingesetzt werden können.

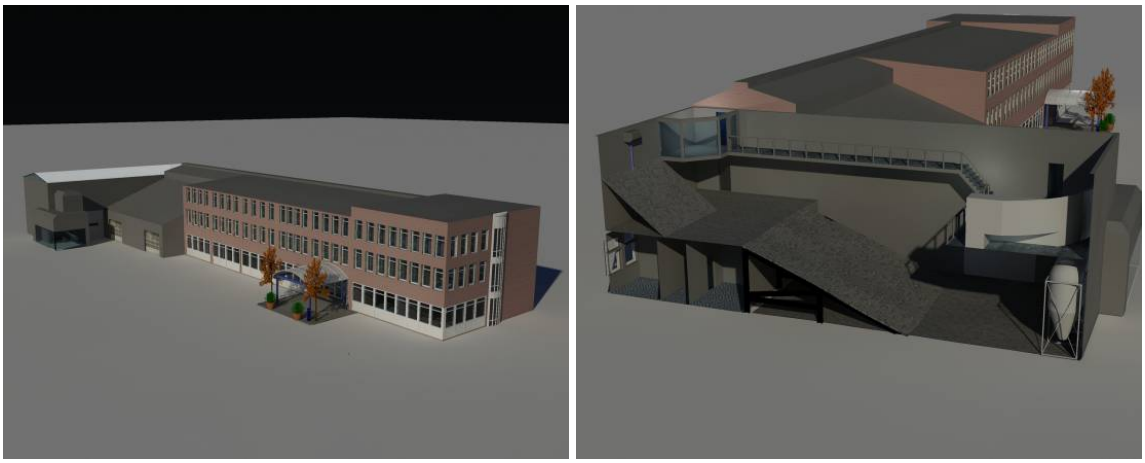


Abbildung 7: Ansichten der Robotics Surface Exploration Halle

Somit können viele relevante Funktionen von zukünftigen und aktuellen Explorationsrobotern hier frühzeitig getestet und auch komplette Missionsabläufe evaluiert werden. Als Referenzsysteme stehen im DFKI momentan der ARAMIES Roboter (DLR Fkz: 50JR0561, ESA Contract: 18116), der Scorpion Roboter und der parallel zu Virtual Crater entwickelte Spaceclimber (DLR Fkz:50RA0705, ESA Contract No. 18116) zur Verfügung (siehe auch Abbildung 8).

Abbildung 8: Robotersysteme des DFKI: ARAMIES (links) , Scorpion (rechts)

Faktoren die diese Umgebungen allerdings nur begrenzt oder gar nicht nachbilden können sind:

- Temperatur
- Atmosphäre
- Strahlungsspektrum, -intensität
- Gravitation (insbesondere bzgl. der Staubdynamik des Untergrundes (z.B. Regolith) über den sich ein mobiler Roboter bewegt)
- Größere Areale, z. B. Shackelton Crater

Bis auf die Gravitationsproblematik (die auf der Erde nicht vollständig gelöst werden kann) sind diese Einschränkungen in erster Linie aufgrund der sonst entstehenden unverhältnismäßig hohen Kosten gegeben.

4.2 Simulationssystem als Grundlage für Virtuelle Testumgebungen

Die im letzten Abschnitt genannten Faktoren sollten aber zu großen Teilen mittels eines leistungsfähigen 3D-Simulationssystems kostengünstig betrachtet werden können. Das 3D-Simulationssystem VEROSIM des RIF – und hier insbesondere die Virtual Reality-Variante – ist durch die Kombination seines leistungsfähigen Simulationskerns mit neuartigen Komponenten zur Physiksimulation und Hardwareintegration, eine ideale Basis hierzu. Der VEROSIM-Simulationskern ist kompatibel zu den Entwicklungen des DLR im Rahmen der Projekte ROTEX, CIROS, VITAL-I bis VITAL-IV, GETEX, ROKVIS, TECSAS/DEOS. Die im Rahmen dieses Projektes eingesetzte Softwareversion ist eine kommerziell verfügbare Softwareplattform, deren Nutzung die Kompatibilität dieses Projektes mit Industriestandards sicherstellt und so eine stabile Basis für die Projektentwicklung bildete.

5 Planung und Ablauf des Vorhabens

Abbildung 9 zeigt den ursprünglichen zeitlichen Phasenplan des Gesamtprojektes, unterteilt in die 39 Arbeitspakete. Ein Schwerpunkt liegt in den Arbeitspaketen zum Thema Bodenmechaniksimulation, da es sich hierbei um eines der aktuellsten Forschungsgebiete handelt, die das Projekt Virtual Crater behandelt und dieses gleichzeitig von besonderer Bedeutung für die Realitätsnähe der Simulation und damit für den Erfolg dieses Projektes ist.

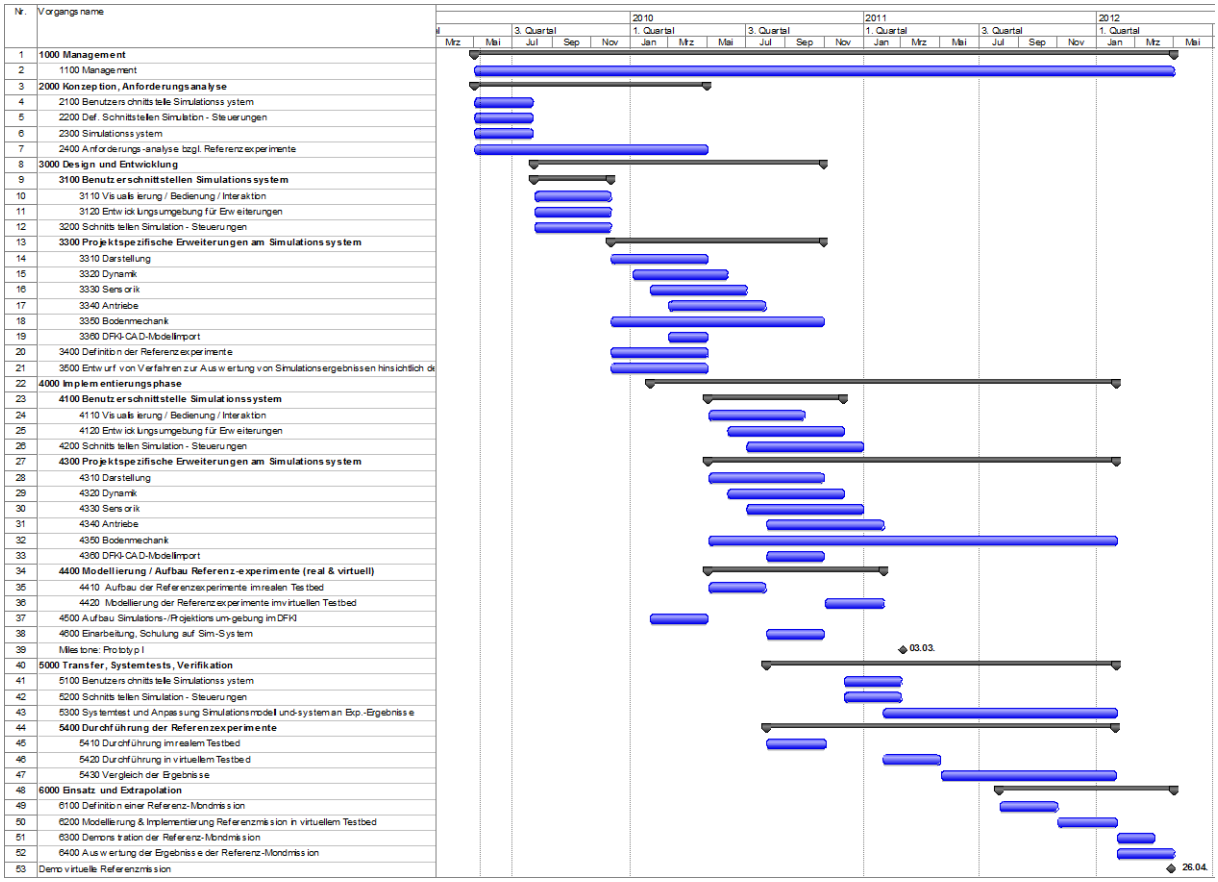


Abbildung 9: Ursprünglicher zeitlicher Phasenplan

Durch den Umzug des DFKI-RIC in ein neues Gebäude hat sich die Installation der 5-fach-Stereoprojektion dort um etwa 4 Monate verzögert. Da die virtuellen Referenzexperimente aber auch auf Arbeitsplatzrechnern durchgeführt werden konnten (abgesehen von der abschließenden Referenzmission), wurde der Gesamtzeitplan dadurch kaum beeinträchtigt. Seitens des MMI

ergaben sich Verzögerungen bei der Implementierungsphase, die u.a. darauf zurück zu führen waren, dass nicht im ursprünglich geplanten Umfang studentische Hilfskräfte gewonnen werden konnten, so dass entsprechende Arbeiten durch die wissenschaftlichen Mitarbeiter ausgeführt werden mussten. Am DFKI ergaben sich ebenfalls Verzögerungen, da einige Referenzexperimente aus technischen Gründen erst später durchgeführt werden konnten oder wiederholt werden mussten. In Abstimmung mit dem DFKI wurde daher eine kostenneutrale Verlängerung des Projektes um 4 Monate beantragt und vom DLR genehmigt. Die verlängerte Projektlaufzeit wurde im Wesentlichen für die Unterstützung des DFKI bei der Implementierung der Bodenmechaniksimulation, den Abgleich von realen und virtuellen Referenzexperimenten und die Verbesserung des Laufzeitverhaltens der ganzheitliche Simulation verwandt (siehe auch Erfolgskontrollbericht, Einhaltung der Zeitplanung).

Der angepasste zeitliche Phasenplan mit der 4-monatigen kostenneutralen Verlängerung ist in Abbildung 10 dargestellt.

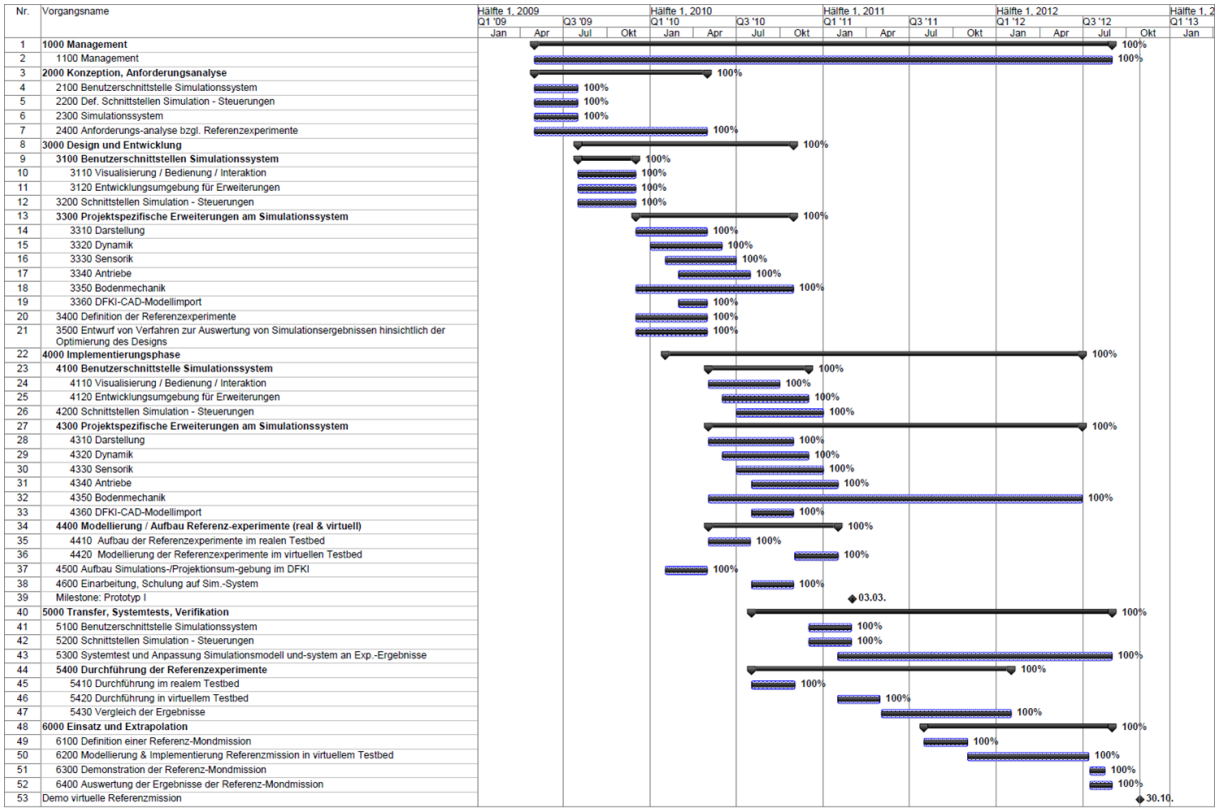


Abbildung 10: Finaler zeitlicher Phasenplan

Wichtige Meilensteine waren:

- Milestone Prototyp I: Nach etwa 22 Monaten der Projektlaufzeit und Abschluss der Implementierungsphase sollte ein erster Prototyp verfügbar sein, der bereits die vollständige Infrastruktur des Projekts abdeckt und erste Ergebnisse liefern kann. Dieser Prototyp wurde im Zuge des Review-Meetings zwischen Kooperationspartnern und DLR am 23.04.2011 am DFKI vorgestellt.
- Das Projekt wurde mit der Demonstration der virtuellen Referenz-Mondmission am 30.10.2012 abgeschlossen.

6 Stand der Wissenschaft und Technik

Nachfolgend werden zunächst maßgebliche Simulationssysteme aufgeführt, die in ihrem ganzheitlichen Ansatz dem als Grundlage für Virtual Crater ausgewählten Simulationssystem ähneln. Anschließend wird der Stand der Technik zu Beginn des Projekts bezüglich der beiden zentralen Simulationskomponenten Dynamiksimulation und Bodenmechaniksimulation dargelegt.

6.1 Artverwandte, ganzheitliche Simulationssysteme

Im Bereich der Simulation mobiler Systeme existieren unterschiedliche auf einzelne Anwendungsbereiche spezialisierte Simulationswerkzeuge.

6.1.1 ROAMS

Ähnliche Ziele wie die des Projekts Virtual Crater verfolgt die Entwicklung des Simulationssystem „ROAMS“ („Rover Analysis, Modeling and Simulation“) [Yen99], 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 [Jai04]. Es konnte gezeigt werden, auf diesem Wege vielversprechende Optimierungen der Simulationsmodelle möglich sind.

In [Soh05] 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 [Soh05]. Exemplarisch seien hier nur einige genannt: Es basiert auf dem „DARTS & Dshell“ Framework [Bie99] für die Raumfahrzeugsimulation. Für die Kollisionserkennung kommt SWIFT++ zum Einsatz, für das 3D-Rendering OpenInventor und für Raytracing-Aufgaben POVRAY.

6.1.2 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 Simulationen 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 sind hier nicht möglich.

Entsprechend seiner Ausrichtung werden hardwareseitig ausschließlich populäre Plattformen wie LEGO Mindstorms™ und Sony Aibo™ unterstützt.

6.1.3 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.

Abbildung 11 beschreibt die grundlegende Struktur einer Player-Architektur. Player selbst fungiert als ein TCP/IP-Server. Es definiert eine Reihe abstrakter Interfaces, die die Schnittstellen zu üblichen Komponenten eines mobilen Robotersystems widerspiegeln. In der Abbildung wurde exemplarisch ein Laserscanner gewählt. „Devices" implementieren die abstrakten Interfaces und stellen unter Zuhilfenahme eines konkreten Treibers die Verbindung mit einer konkreten Hardware oder auch einer entsprechenden Simulation, im Beispiel einem simulierten Laserscanner her.

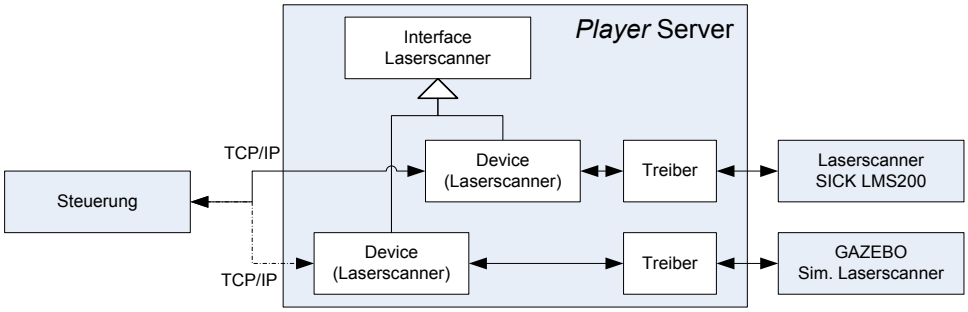


Abbildung 11: Schematische Darstellung einer Player-Architektur

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 [OGRE], als Dynamiksimulationssystem die Open Dynamics Engine [ODE] verwendet.

Player liefert im Hinblick auf die Abstraktion von Hardware oder Simulation einen interessanten Ansatz, gleichzeitig durch die Vorgabe bestimmter Infrastrukturen und eines begrenzten derzeitigen Realisierungsstatus jedoch auch große Einschränkungen. Der Einsatz spezieller Hardware würde die Implementierung eigener Treiber erfordern, ebenso müssten Interfaces erweitert und neue angelegt werden. Erweiterungen der Darstellungsverfahren und neu zu realisierende Verfahren der physikalischen Simulation erfordern Eingriffe in praktisch alle Dritt-Bibliotheken des Systems. Somit kann dieses Modell zwar als Muster dienen, ist als Grundlage für die Realisierung von Virtual Crater jedoch ungeeignet.

6.2 Dynamiksimulation

Im Bereich der interaktiven Echtzeitanwendungen waren aktuelle Verfahren der Starrkörperdynamiksimulation unter Berücksichtigung von Kollisionen, Reibung und Gelenken eine geeignete Ausgangsbasis für die Entwicklungen in Virtual Crater. Zudem war die Simulation flexibler Elemente für eine realitätsnahe, ganzheitliche Simulation mobiler Robotersysteme wünschenswert, um z.B. Feder-Dämpfersysteme oder Biegeelemente simulieren zu können. Mit einer ausgefeilteren Realisierung sollten sich allerdings neben der großen Menge Starrkörper-basierter Anwendungen auch solche mit flexiblen Elementen realisieren lassen.

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. Sieht 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 [Hahn88] und Brian Mirtich [Mir96]. Ebenfalls zu den impulsbasierten Verfahren wird „Stewarts Methode" gezählt [Ste95]. Sie war Grundlage für viele weitere Arbeiten, u.a. von Anitescu [Ani97] Zuletzt stellte Jan Bender [Ben07] eine praxistaugliche Realisierung [IBDS] des eines impulsbasierten Verfahrens mit einigen Optimierungen vor. Daneben nutzt auch die Open Source Dynamiksimulationsbibliothek „Bullet" [Bul] ein impulsbasiertes Verfahren.

Die Methode mit Lagrange Multiplikatoren wurde bereits 1990 in [Wit90] 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 [Bar94] 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 [Cott68] gelöst. Die Open Dynamics Engine [ODE], ebenfalls eine Open Source Dynamiksimulationsbibliothek, nutzt diese Methode.

Das Simulationssystem des 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. Auch Erweiterungen, die **über die reine Starrkörperdynamik hinausgehen,** wie die Simulation flexibler Bauteile, sind auf Basis dieses Verfahrens geplant und bereits in prototypischen Modellen realisiert.

6.3 Bodenmechaniksimulation

Die Bodenmechaniksimulation ist im Rahmen von Virtual Crater im Hinblick auf die Interaktionseffekte zwischen Laufroboter und Boden von Interesse. Bisher durchgeführte Untersuchungen in diesem Bereich beziehen sich wiederum vor allem auf **radgebundene** mobile Roboter.

Die Mehrheit der Untersuchungen verfolgt dabei einen makroskopischen Ansatz, d.h. es werden vereinfachte mathematische Modelle der bei der Interaktion wirkenden Kräfte entworfen und durch Experimente validiert und parametriert. Ein mikroskopischer Ansatz, bei dem das dynamische Verhalten z.B. einzelner Sandkörner und ihrer Interaktionen untersucht wird, verursacht demgegenüber einen erheblichen Rechenaufwand und scheint zum heutigen 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" [Bek59]. Diese Theorie beschreibt mathematische Performancemodelle für die Mobilität und Lokomotion von radbetriebenen Robotern. In diesen Modellen werden unter anderem Bodenköhäsion, Gravitation und Reibungswinkel als Eingabeparameter verwendet. Ausgabewerte der Modelle sind „Drawbar Pull", „Soil Thrust" und „Motion Resistance". Zwei planetarische Rover-Exploration-Simulatoren, RMPET (Rover Mobility Performance Evaluation Tool) [Patel04] und RCET (Rover Chassis Evaluation Tools) [Mich04], 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 [Bau05] 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 [ASE03] hergeleitet. In [Bau05b] wird schließlich beschrieben, wie die Ergebnisse dieser Untersuchungen in ein Simulationssystem auf Basis von Matlab / Simulink einfließen.

Kazuya Yoshida und Hiroshi Hamano entwickeln in [Yos02] 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 vergleichen, 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 bisherige Arbeiten sehr stark auf den Bereich der radgebundenen Roboter bezogen sind, so dass die Ergebnisse nicht ohne weitere, anwendungsspezifische Untersuchungen auf die Simulation von Laufrobotern übertragbar waren.

7 Zusammenarbeit mit anderen Stellen

Das Projekt wurde als Kooperationsprojekt gemeinsam mit der Forschungsgruppe Robotik des Deutschen Forschungszentrums für künstliche Intelligenz in Bremen bearbeitet.

DFKI GmbH
Forschungsgruppe Robotik
Prof. Dr. Frank Kirchner
Robert-Hooke-Str. 5
28359 Bremen

RIF hat in Abstimmung mit dem DLR einen Teil der Arbeitspakete als Auftrag an das Institut für Mensch-Maschine-Interaktion (MMI) der RWTH Aachen vergeben. Das MMI hat die Abstimmung der Anforderungsdefinition mit dem DFKI (APs 2000) sowie einen Teil der Entwicklungsphase (APs 3100, 3200, 3310-3350) und der Implementierungsphase (APs 4100, 4200, 4310-4350) übernommen.

Institut für Mensch-Maschine-Interaktion (MMI)
RWTH Aachen
Prof. Dr.-Ing. Jürgen Roßmann
Ahornstr. 55
52074 Aachen

8 Verwendung der Zuwendung, Ergebnis, Gegenüberstellung der vorgegebenen Ziele

Im Folgenden werden die in Kapitel 3 aufgeführten Ziele den Projektergebnissen gegenübergestellt. Für Details wird dabei auf die im beigefügten *Technischen Schlussbericht* dokumentierten Arbeitspakete (APs) verwiesen.

Ziel 1: Nutzung von Synergieeffekten durch parallele Entwicklung in realer und virtueller Umgebung

Ergebnis 1: Das DFKI hat eine Reihe von Referenzexperimenten definiert und gemeinsam mit RIF und MMI durchgeführt. Zum Teil hat RIF/MMI die virtuellen Anteile modelliert (Motormodell, Bodenmechanik-Modell, AP4420). Einige weitere Experimente sind vollständig in hybrider Form, d.h. gleichzeitig real und virtuell von RIF/MMI durchgeführt worden (Kraft-Drehmoment-Sensor,

IMU, PMD-Tiefenkamera, AP5400). Anhand der Daten dieser Referenzexperimente konnte eine kalibrierte virtuelle Testumgebung realisiert werden, auf Basis derer eine Referenzmission geplant, virtuell durchgeführt und bewertet werden konnte (AP6200, AP6300). Durch Implementierung von Darstellungs- und Analysewerkzeugen sowie neuer Visualisierungsmetaphern (AP4110, AP4310) fördert die Simulation ein tiefer gehendes Verständnis der Zusammenhänge in komplexen Systemen. So konnten durch die Referenzexperimente auch wertvolle Rückschlüsse auf die Funktionsweise der realen Komponenten gezogen werden.

Ziel 2: Anbindung der Robotersteuerung, so dass dieselbe Steuerung für realen und simulierten Roboter verwendet werden kann

Ergebnis 2: Es wurde eine generische Schnittstelle implementiert, welche sich gegenüber dem Roboter-Controller MONSTER des DFKI (Generierung von Laufmustern) so verhält wie der reale Roboter (AP4200, AP5200). Dabei wird zwischen der Simulation und dem Controller eine synchrone Kommunikation aufgebaut, die die Sollwerte der Gelenke auf die simulierten Motormodelle aufschaltet und die Messwerte der virtuellen Sensoren an den Controller zurückkoppelt. Damit kann die Robotersteuerung vollkommen transparent mit einem realen oder einem simulierten Roboter verbunden werden.

Ziel 3: Frei konfigurierbare Testsysteme

Ergebnis 3: Das Virtuelle Testbed steht allen Entwicklern gleichzeitig auf ihren Rechnern (PC-Arbeitsplatz oder Laptop) vollständig als lauffähige Arbeitsgrundlage zur Verfügung (AP4120) und kann gleichzeitig in der Stereoprojektion zur Präsentation und Demonstration genutzt werden (AP6300). Controller, Simulation und Visualisierung können auf unterschiedlichen Rechnern laufen (AP5200, AP5300, AP6200). Die Verteilung der Simulationsalgorithmen selbst auf verschiedene Rechner ist prinzipiell möglich, bietet aber in diesem Anwendungsfall aufgrund der hohen Taktraten, mit denen die Simulationskomponenten synchronisiert werden müssen, keinen großen Vorteil.

Ziel 4: Stereoskopische Mehrschirm-Projektionsumgebung

Ergebnis 4: Es wurde eine Projektionsumgebung bestehend aus 5 Stereoprojektionswänden am DFKI in Bremen aufgebaut, die eine Abdeckung des gesamten horizontalen Blickwinkels ermöglicht (AP4500). In dieser Projektionsumgebung erfolgte die Durchführung der Referenzmission und die Demonstration der Projektergebnisse gegenüber dem DLR am Milestone (23.04.2011) und bei der Abschlusspräsentation (30.10.2012).

Ziel 5: Weiterentwicklung der realen Testumgebung

Ergebnis 5: Die realen Testaufbauten fielen in die Zuständigkeit des DFKI. Zum Teil konnten auf Basis des im Rahmen des ebenfalls vom DLR geförderten Projekts FastMap (DLR-Förderkennzeichen 50RA1034) aufgebauten Mockups auch bei RIF/MMI zusätzliche reale bzw. hybride Tests durchgeführt werden (siehe auch Ergebnis 1).

Ziel 6: Validierung und Weiterentwicklung der Virtuellen Testumgebung

Ergebnis: Ausgehend von dem in Kapitel 3.6 dargestellten iterativen Vorgehen wurden zwei grundsätzliche Vorgehensweisen entworfen. Erstens ein Ziel-Workflow für den Nutzer des Virtuellen Testbeds, der auf dieser Grundlage einen neuen Laufroboter testen oder eine Mission planen möchte. Zweitens ein Arbeitsablauf zur Entwicklung und Validierung des Virtuellen Testbeds selbst. Letzteres ist zur Entwicklung von Virtual Crater umgesetzt worden (siehe auch Ergebnis 1). Virtual Crater selbst bietet damit nun die Möglichkeit, genau den zu Anfang definierten Ziel-Workflow zur effizienten simulationsgestützten Entwicklung von Raumfahrtanwendungen zu nutzen (vgl. Zusammenfassung des *Technischen Schlussberichts*).

Ziel 7: Weitere Anwendungen, Transfer

Ergebnis: Die in Virtual Crater erarbeiteten Ergebnisse, insbesondere zur Infrastruktur Virtueller Testbeds lieferten wertvolle Erkenntnisse für aktuell vom DLR geförderte Projekte wie beispielsweise FastMap (DLR-Förderkennzeichen 50RA1034), SELOK (DLR-Förderkennzeichen 50RA0911) oder iBOSS-2 (DLR-Förderkennzeichen 50RA1203). So wurde auf der Abschlusspräsentation am 30.10.2012 die Nutzung der in SELOK entwickelten Algorithmen zur Selbstlokalisation des DFKI-Roboters SpaceClimber in lunarer Kraterumgebung demonstriert. Aktuell wird im Rahmen von FastMap daran gearbeitet, die aus dem planetaren Anflug abgeleiteten Informationen ebenfalls mit dem in VirtualCrater betrachteten Referenz-Szenario zu verknüpfen. Auf diese Weise entsteht ein in seinem Funktionsumfang immer umfassenderes Werkzeug zur ganzheitlichen Planung von Raumfahrtmissionen und zur simulationsgestützten Entwicklung der notwendigen Technologien. Zudem könnte zukünftig die Einsatzfähigkeit des VR-Systems im Anwendungsbereich Robotik auf Basis der entstandenen Kooperation zwischen RIF e.V. und der DFKI GmbH weiter verbessert und auf zusätzliche Bereiche wie z.B. die Unterwasserrobotik ausgeweitet werden.

Verwendung der Zuwendung: Zur Erarbeitung dieser Ergebnisse sowie zur Erreichung der beschriebenen Ziele war die Aufwendung der in Kapitel 9 aufgelisteten Mittel notwendig.

9 Wichtigste Positionen des zahlenmäßigen Nachweises

Personalkosten	202.284,53 €
Forschungsanteil des Instituts für Mensch-Maschine-Interaktion der RWTH Aachen	298.315,00 €
Gegenstände > 400 €	19.939,39 €
sonstige Ausgaben	19.400,63 €
Gesamt	539.939,55 €

10 Notwendigkeit und Angemessenheit der geleisteten Arbeit

Im Sinne des definierten Projekt-Gesamtziels wurden im Projektverlauf immer wieder kleine Korrekturen des Projektplans vorgenommen. Diese sind im beigefügten Erfolgskontrollbericht und im technischen Schlussbericht dokumentiert. Insgesamt waren alle Arbeiten zur Erreichung der Projektziele unabdingbar notwendig und konnten im Rahmen der zu Beginn kalkulierten Kosten durchgeführt werden.

11 Nutzen und Verwertbarkeit des Ergebnisses

Der Nutzen des Projektergebnisses ist ausführlich im beigefügten technischen Bericht dokumentiert und im fortgeschriebenen Verwertungsplan zusammengefasst.

12 Fortschritt auf dem Gebiet des Projekts bei anderen Stellen

ROAMS selbst ist seit [Soh05] und einer für ROAMS erfolgten Modellierung und Kalibration einer Kamera [Mad05] nicht weiterentwickelt worden. Die Arbeiten des NASA JPL auf diesem Gebiet in den letzten Jahren sind in einem „Lunar Surface Operation Testbed“ [Tre12] gemündet. Der Hauptteil dieser Arbeiten besteht in einem Hardware-Mockup mit einer Reihe von Komponenten, die für Lander und Explorationsrover typisch sind. Softwareseitig wurde ein „Robot Sequencing and Visualization Program“ (RSVP) entwickelt, welches zur Visualisierung von Terrain, Kommandosequenzen und zur Darstellung und Analyse numerischer Daten dient. Es verfügt außerdem über eine rein kinematische Simulation der Roboterbewegung.

Cyberbotics Webots™ (www.cyberbotics.com) und vergleichbare Systeme wie z.B. V-Rep (www.vrep.eu) oder Microsoft Robotics Studio (www.microsoft.com/robotics) haben sich insgesamt in ihrem Funktionsumfang (z.B. unterstützte Sensortypen) und ihrer Benutzerfreundlichkeit weiterentwickelt. Der Fokus auf die phänomenologische abstrahierte Beschreibung von Funktionsweisen als Plattform für Lehre und Grundlagenforschung im Bereich KI ist aber geblieben (siehe z.B. [Maa12]).

Veröffentlichungen rund um Player / Stage / Gazebo entwickeln sich aktuell hauptsächlich in Richtung Service-Robotik, Objekterkennung, Greifen und Mensch-Roboter-Interaktion (z.B. [Lim13]). Weltraumrobotik spielt dabei keine Rolle. Nichtsdestotrotz bleibt die Infrastruktur interessant und beobachtenswert.

Insgesamt ist Virtual Crater als ganzheitliches Werkzeug zur simulationsgestützten Entwicklung, Planung, Verifikation und Demonstration lunarer Explorationsmissionen zurzeit einzigartig. Insbesondere die flexible Mehrkörperdynamiksimulation als Simulationsgrundlage, die Erweiterbarkeit um detaillierte Simulationsmodelle, wie z.B. Antriebe und Motormodelle, sowie die konsequente Umsetzung der Kalibration von Simulationsmodellen anhand von Referenzexperimenten heben Virtual Crater von vergleichbaren Projekten ab.

13 Veröffentlichungen

- Roßmann, J., Schluse, M., Sondermann, B., Emde, M., Rast, M.: Advanced Mobile Robot Engineering with Virtual Testbeds. In: Proceedings for the Conference of ROBOTIK 2012, 7th German Conference on Robotics, May 21-22, Munich, pp. 331-336, ISBN 978-3-8007-3418-4, VDE Verlag GmbH Berlin
- Jung, T., Rast, M., Guiffo Kaigom, E., Roßmann, J.: Fast VR Application Development Based on Versatile Rigid Multi-Body Dynamics Simulation. In: Proceedings of the ASME 2011 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (IDETC/CIE), August 28-31,2011, Washington DC, http://www.asmeconferences.org/IDETC2011/SearchPaperSchedule.cfm
- Roßmann, J., Rast, M., Jung, T., Guiffo Kaigom, E.: Modellierung und Synchronisation von Simulationsaufgaben für Virtuelle Testbeds. In: Gausemeier, J. / Grafe,M. / Meyer auf der Heide, F. (Hrsg.): Wissenschaftsforum 2011 Intelligente Technische Systeme - 10. Paderborner Workshop Augmented & Virtual Reality in der Produktentstehung, 19.-20. Mai 2011, Heinz Nixdorf Institut, Universität Paderborn, Bd. 295, pp. 145-156, ISBN 978-3-942647-14-4
- Roßmann, J., Jung, T., Rast, M.: Developing virtual testbeds for tasks in research and engineering. In: Proceedings of the ASME 2010 World Conference on Innovative Virtual Reality (WINVR 2010), MAY 12-14, 2010, Ames, Iowa, USA, ISBN 978-0-7919-3869-3
- Roßmann, J., Jung, T., Rast, M.: Entwicklung Virtueller Testbeds mit Dynamik- und Bodenmechaniksimulation für Aufgaben in Forschung und Entwicklung. In: Gausemeier, J./ Grafe, M. (Hrsg.): "9. Paderborner Workshop Augmented & Virtual Reality in der Produktentstehung (ARVR 2010), 10-11 Juni, Paderborn, Bd. 274, HNI-Verlagsschriftenreihe, pp. 173-187, ISBN 978-3-939350-93-4 *(Ausgezeichnet mit dem Best-Paper-Award)*
- Roßmann, J., Jung, T., Rast, M.: Developing Virtual Testbeds for Mobile Robotic Applications in the Woods and on the Moon. In: The IEEE/RSJ 2010 International Conference on Intelligent Robots and Systems(IROS 2010), Taipei, Taiwan, October 18-22, 2010, Conference Proceedings, pp. 4952-4957, ISBN: 978-1-4244-6675-7 / ISSN: 2153-0866
- Yoo, Y.-H. , Jung, T., Römmermann, M., Rast, M., Kirchner, F., Roßmann, J.: Developing a Virtual Environment for Extraterrestrial Legged Robots with Focus on Lunar Crater

Exploration. In: The 10th International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS 2010), August 29 - September 1, 2010, Sapporo, Japan, PP. 206-213

14 Literatur

[Bek59] M. Bekker, “Theory of Land Locomotion: Mechanics of Vehicle Mobility,” University of Michigan Press, Ann Arbor, USA, 1959.

[Cott68] R. W. Cottle und G. B. Dantzig,: Complementary Pivot Theory of Mathematical Programming. Linear Algebra and Appl. 1, pp. 103-125, 1968.

[Hahn88] J. K. Hahn: Realistic Animation of Rigid Bodes. Proceedings of the 15th annual conference on Computer graphics and interactive techniques, 1988.

[Wit90] A. Witkin, M. Gleicher, und W. William: Interactive Dynamics. Computer Graphics, Symposium on Interactive 3D Graphics, 24, 11-21, 1990,.

[Bar94] D. Baraff: Fast Contact Force Computation for Nonpenetrating Rigid Bodies. Proceedings of the 21st annual conference on Computer graphics and interactive tech-niques, 1994.

[Ste95] D. Stewart und J. C. Trinkle: An Implicit Time-Stepping Scheme For Rigid Body Dynamics With Inelastic Collisions and Coulomb Friction. International Journal of Numerical Methods in Engineering, eingereicht 1995.

[Mir96] B. V. Mirtich: Impulse-based dynamic simulation of rigid body systems. PHD-Thesis, University of California, Berkeley, 1996.

[Ani97] M. Anitescu und F. A. Potra: Formulating dynamic multi-rigid-body contac-problems with friction as solvable linear complementarity problems. Nonlinear Dynamics 14: 231–247, 1997.

[Bie99] J. Biesiadecki, D. Henriquez, und A. Jain: A reusable, real-time spacecraft dynamics simulator. In 6th Digital Avionics System Conference, 1999.

[Yen99] J. Yen, A. Jain und J. Balaram: ROAMS: Rover Analysis, Modeling and Simulation. Proceedings of the Fifth International Symposium on Artificial Intelligence, Robotics and Automation in Space (iSAIRAS), 1999.

[Yos02] K. Yoshida und H. Hamano: Motion Dynamics of a Rover with Slip-Based Traction Model. Proceedings of the 2002 IEEE International Conference on Robotics and Automation, ICRA 2002, May 11-15, 2002, Washington, DC, USA.

[ASE03] AESCO, “Matlab/Simulink Module AESCO Soft Soil Tyre Model (AS2TM) User’s Guide”, 2003.

[Jai04] A. Jain, J. Balaram, J. Cameron, J. Guineau, C. Lim, M. Pomerantz, G. Sohl: Recent Developments in the ROAMS Planetary Rover Simulation Environment. IEEE 2004 Aerospace Conf., Big Sky, Montana, March 6-13, 2004.

[Patel04] N. Patel, A. Ellery, und G. Scott, “Application of Bekker Theory to Wheeled, Tracked and Legged Vehicles”, SPACE 2004, San Diego, California, USA. (September 2004)

[Mich04] S. Michaud, R. Richter, N. Patel, T. Thuer, T. Huelsing, L. Joudrier, R. Siegwart, und A. Ellery, “Rover Chassis Evaluation Tool (RCET)”, ESA-ASTRA, European Space Agency, ESTEC, Noordwijk, The Netherlands. (November 2004)

[Bau05] R. Bauer, W. Leung und T. Barfoot: Experimental and Simulation Results of Wheel-Soil Interaction for Planetary Rovers. Proceedings of the International Conference on Intelligent Robots and Systems (IROS), 2005.

[Bau05b] R. Bauer, W. Leung und T. Barfoot: Development of a Dynamic Simulation Tool for the Exomars Rover. Proceedings of the Fifth International Symposium on Artificial Intelligence, Robotics and Automation in Space (iSAIRAS), 2005.

[Soh05] G. Sohl and A Jain: Wheel-Terrain Contact Modeling in the Roams Planetary Rover Simulation. 5th ASME International Conference on Multibody Systems, Nonlinear Dynamics and Control, Long Beach, CA, September 2005.

[Ben07] B. Jan: Impulsbasierte Dynamiksimulation von Mehrkörpersystemen in der virtuellen Realität. Dissertation, Universität Karlsruhe, 2007.

[Bul] Bullet Physics Library. WWW: http://www.bulletphysics.com/Bullet/ (Zugriff: Feb, 08).

[IBDS] J. Bender: Impulse Based Dynamic Simulation. WWW: http://www.impulse-based.de (Zugriff: Feb, 08).

[ODE] R. Smith: Open Dynamics Engine. WWW: http://www.ode.org (Zugriff: Feb, 08).

[OGRE] Open Graphics Engine. WWW: http://www.ogre3d.org (Zugriff: Mrz, 08).

[Mad05] R. Madison, M. Pomerantz and A. Jain: Camera Response Modeling and Verification in ROAMS, i-SAIRAS 2005, September 2005.

[Tre12] A. Trebi-Ollennu, K. S. Ali, A. L. Rankin, K. S. Tso, C. Assad, J. B. Matthews, R. G. Deen, D. A. Alexander, R. Toda, H. Manohara, M. Wolf, J. R. Wright, J. Yen, F. Hartman, R. G. Bonitz, and A. R. Sirota: Lunar Surface Operation Testbed (LSOT), Proceedings of IEEE Aerospace Conference, Mar. 2012.

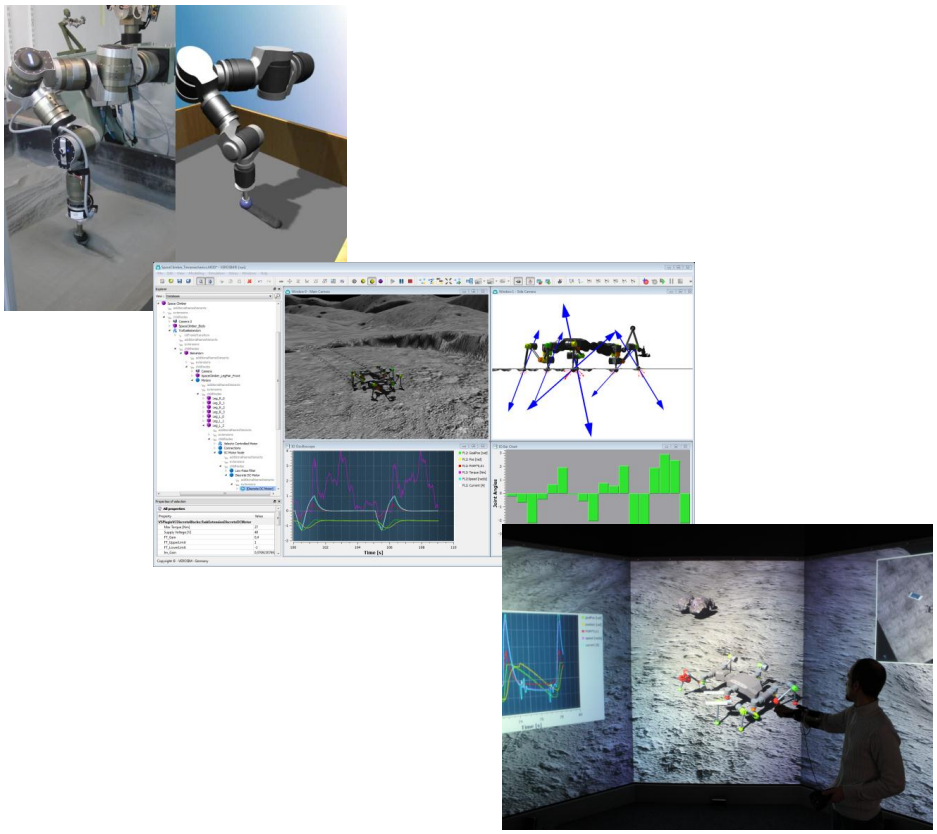
[Lim13] J. L. Lima, J.A. Goncalves, P.G. Costa and A. P. Moreira: Humanoid Gait Optimization Resorting to an Improved Simulation Model, International Journal of Advanced Robotic Systems, 2013, Vol. 10, DOI: 10.5772/54766

[Maa12] R. Maas and E. Maehle: An Easy to Use Framework for Educational Robots, ROBOTIK2012, Munich, Germany

Virtual Crater
Technischer Abschlussbericht

RIF Institut für Forschung und Transfer e.V.
Förderkennzeichen: 50 RA 0913
Projektleiter (RIF): Prof. Dr.-Ing. Jürgen Roßmann
Laufzeit: 1.5.2009 – 31.8.2012

25. Februar 2013



RIF Institut für Forschung und Transfer e.V.
Prof. Dr.-Ing. Jürgen Roßmann
Joseph-von-Fraunhofer Str. 20
44227 Dortmund
Germany

Inhaltsverzeichnis

1 Zusammenfassung

Im Rahmen des Projekts *Virtual Crater* wurde eine virtuellen Testumgebung entwickelt, die es ermöglicht, Robotersysteme kostengünstig in einer realitätsnah simulierten, lunaren Kraterlandschaft zu programmieren, zu testen und zu optimieren. Das Projekt wurde in einer Kooperation des Deutschen Forschungszentrums für Künstliche Intelligenz (DFKI GmbH) mit dem RIF Institut für Forschung und Transfer e.V. bearbeitet.

Virtual Crater ist eine umfassende Simulationsumgebung, die es ermöglicht, Missionen zur Erforschung der Mondoberfläche zu programmieren und zu testen sowie neue Konzepte vorzuführen. Entwicklerteams, die einen neuen Laufroboter entwickeln oder gar eine ganze Mission planen möchten, können jetzt auf Basis von *Virtual Crater* durch den in Abb.1 dargestellten Arbeitsablauf schnelle parallele Entwicklungszyklen anhand virtueller Prototypen durchlaufen und so die Entwicklungszeiten verkürzen und reale Prototypen einsparen - bei gleichzeitiger Steigerung von Qualität und Robustheit der Implementierung.

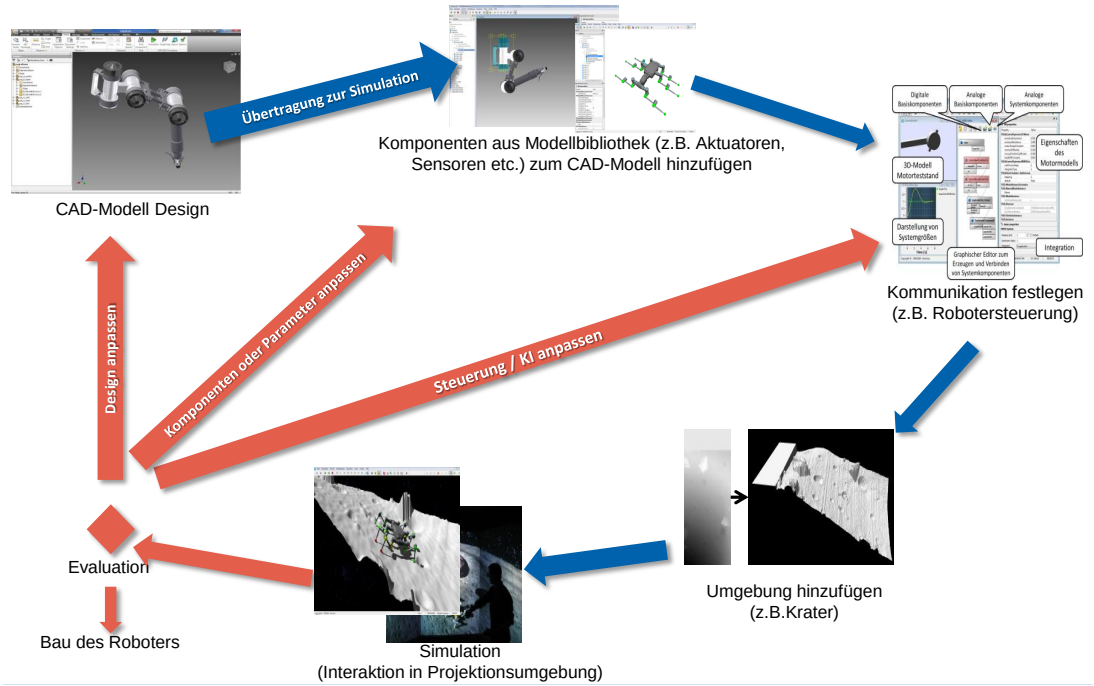


Abbildung 1: *Entwicklungszyklus auf Basis von Virtual Crater.*

Damit sich diese Testumgebung so realitätsnah wie möglich verhält, wurden umfassende physikalische Experimente durchgeführt und mit analogen simulierten Experimenten verglichen. Diese Referenzexperimente und die Kalibration der einzelnen detaillierten Simulationsmodelle wurden im Rahmen des Projektes parallel zur Entwicklung des ganzheitlichen Simulationssystems vorangetrieben und im späteren Projektverlauf zusammengeführt (siehe Abb. 2).

Die Simulation wiederum kann dann - gegebenenfalls zusammen mit speziellen Optimierungswerkzeugen - unter anderem dazu verwendet werden, um Hardware zu optimieren, Explorations-Szenarien aufzubauen und entsprechende Missionen zu simulieren. Hierbei war es wichtig, 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, siehe Abb. 3) sowie auf einem normalen Arbeitsplatzrechner als autarke Entwicklungsumgebung eingesetzt.

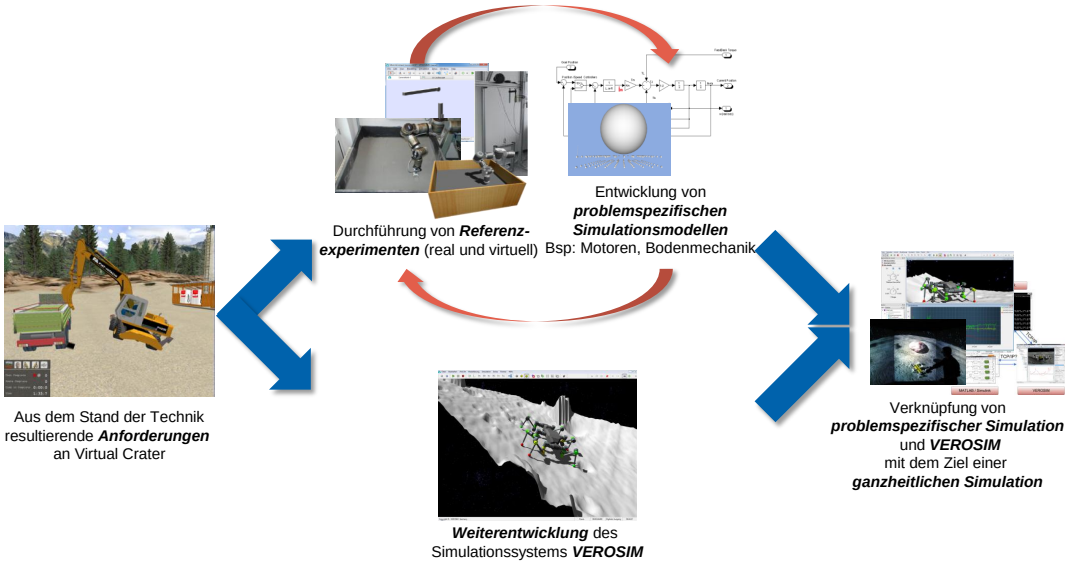


Abbildung 2: *Entwicklung des Virtuellen Testbeds Virtual Crater.*

Abbildung 3: *Links: CAVE-System in der ersten Ausbaustufe mit 5 Stereoprojektionsschirmen Rechts: Interaktionsbeispiel in der CAVE*

Das Gesamtkonzept des Projektes wurde in einer gemeinsamen Veröffentlichung der beiteiligten Partner auf der i-SAIRAS (International Symposium on Artificial Intelligence, Robotics and Automation in Space) 2010 vorgestellt [YJR+10]. Die gesammelten Erfahrungen insbesondere im Hinblick auf den Integrationsaspekt des Virtuellen Testbeds wurden in [RSS+12b] veröffentlicht. Darüber hinaus wurden Einzelaspekte in diversen weiteren Veröffentlichungen präsentiert.

1.1 Administratives

Um die im ersten Meeting zwischen den Kooperationspartnern (13.06.2009) definierten Ziele umzusetzen, wurden die entsprechenden Arbeitspakete während der Konzeptphase und Entwicklungsphase zwischen den Projektpartnern aufgeteilt (siehe Abb. 4). Ein Großteil der Anforderungsanalyse (APs 2000) fand auf Seiten des DFKIs statt, da sie hauptsächlich auf die Simulationsanforderungen für Missionen und Szenarien im SpaceClimber- und LUNARES-Projekt basiert. Die Design-Phase (APs 3000) wurde in enger Abstimmung zwischen den Kooperationspartnern durchgeführt und die Inhalte sind größtenteils gemeinsam erarbeitet worden. Die Implementierungsphase (APs 4000) oblag hauptsächlich RIF. Hier war es ein erklärtes Ziel des Projekts, dem DFKI eine vollständige Entwicklungsumgebung für VEROSIM®zur Verfügung zu stellen, sodass auch dort

Bodenmechanik- und Motormodelle direkt in VEROSIM®implementiert werden konnten. Systemtests und Verifikation (APs 5000) wurden über die zweite Projekthälfte iterativ wechselseitig von den Kooperationspartnern bearbeitet. Während der vom DFKI koordinierten Extrapolationsphase (APs 6000) ergaben sich immer wieder zusätzliche Anforderungen, die dann wiederum in möglichst kurzen Zyklen von RIF erfüllt wurden.

Die Abstimmung der Anforderungsdefinition (APs 2000), ein Großteil der Entwicklungsphase (APs 3100, 3200, 3310-3350) und der Implementierungsphase (APs 4100, 4200, 4310-4350) wurden in Abstimmung mit dem Projektträger von RIF als Unterauftrag an das Institut für Mensch-Maschine-Interaktion (MMI) der RWTH Aachen vergeben.

In diesem Bericht werden die Hauptbeiträge von RIF zum Projekt Virtual Crater dokumentiert, d.h. die in Abb. 4 rot markierten Arbeitspakete. Da das Projekt in sehr enger Kooperation durchgeführt wurde, werden aber auch teilweise zu den vom DFKI koordinierten APs Beiträge dokumentiert und umgekehrt werden im technischen Bericht des DFKI auch Beiträge zu Arbeitspaketen des RIF aufgeführt.

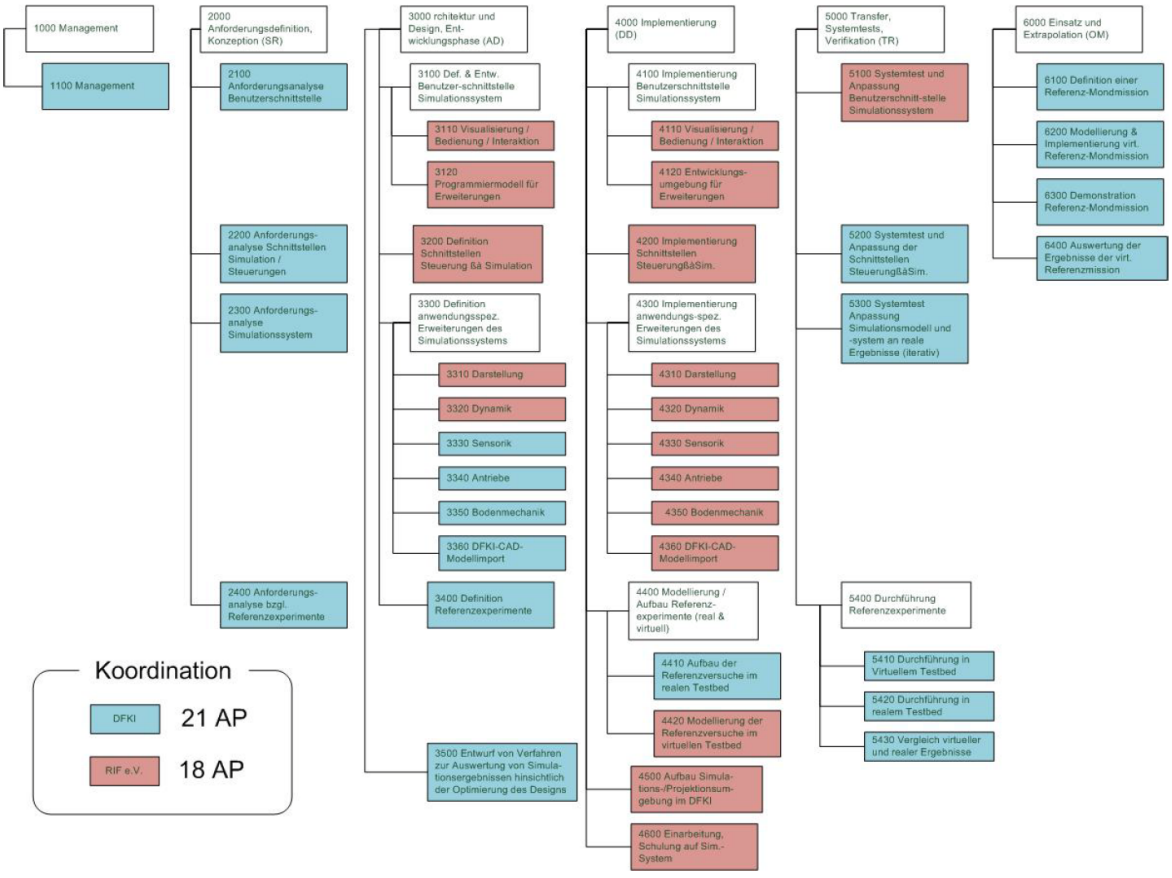


Abbildung 4: *Strukturierung und Zuordnung der Arbeitspakete*

Am 01.10.2009 fand ein Review-Meeting mit den Kooperationspartnern und dem Projektträger statt. In diesem Meeting haben die Kooperationspartner in einer gemeinsamen Präsentation und Demonstration dem Projektträger die Arbeitsergebnisse und die weitere Arbeitsplanung vorgestellt und zur Diskussion gestellt.

Das RIC des DFKI hat aufgrund der aktuellen Expansion ein neues Gebäude angemietet, in das einige Mitarbeiter und Projekte Anfang Mai 2010 umgezogen sind. Dies betraf auch die Simulationsabteilung und damit auch das Projekt Virtual-Crater. Daher wurde entschieden, auch die unten genannte Simulations- und Projektionsumgebung in diesem neuen Gebäude aufzubauen. Hierfür musste eine ausreichende Stromversorgung und Klimatisierung sicher gestellt werden. Da das für die Projektionsumgebung verwendete Simulationssystem

VEROSIM® auch auf Standard-Rechnersystemen lauffähig ist, ergaben sich aus der Verzögerung des entsprechenden Arbeitspaketes 4500 keine negativen Einflüsse auf den geplanten Projektablauf. Alle geplanten virtuellen Experimente (bis auf die ganzheiliche Simulation der Referenzmission) konnten zunächst auch ohne die Projektionsumgebung durchgeführt werden. Am 27.04.2010 fand ein Treffen zwischen MMI und DFKI-RIC statt, in dem die Implementierung des bisher erarbeiteten Simulationsdesigns geplant wurde.

Am 23.04.2011 fand ein Review-Meeting mit den Kooperationspartnern und dem Projektträger statt. In diesem Meeting haben die Kooperationspartner in einer gemeinsamen Präsentation und Demonstration dem Projektträger die Arbeitsergebnisse und die weitere Arbeitsplanung vorgestellt.

Die bisherigen Ergebnisse des Projektes wurden auf der 2. Nationalen Konferenz zur Raumfahrt-Robotik am 6. und 7. März 2012 in Berlin und auf einer Stereoprojektion in der Begleitausstellung im Deutschen Museum in Bonn präsentiert.

Entsprechend des von den Projektpartnern gemeinsam gestellten Antrags wurde die Projektlaufzeit um 4 Monate bis zum 31.08.2012 kostenneutral verlängert. Infolgedessen wurde der Projektplan für die verbleibenden Arbeitspakete entsprechend angepasst.

Das Projektergebnis wurde dem Auftraggeber am 30.10.2012 am DFKI in Bremen präsentiert. Dabei wurden sowohl die erreichten Ziele in einer Präsentation dargestellt, als auch der praktische Nutzen anhand einer Demonstration des Arbeitsablaufes sowohl am PC-Arbeitsplatz als auch in der Stereoprojektionsumgebung veranschaulicht.

Abbildung 5 zeigt den abgearbeiteten Projektplan zum Projektende.

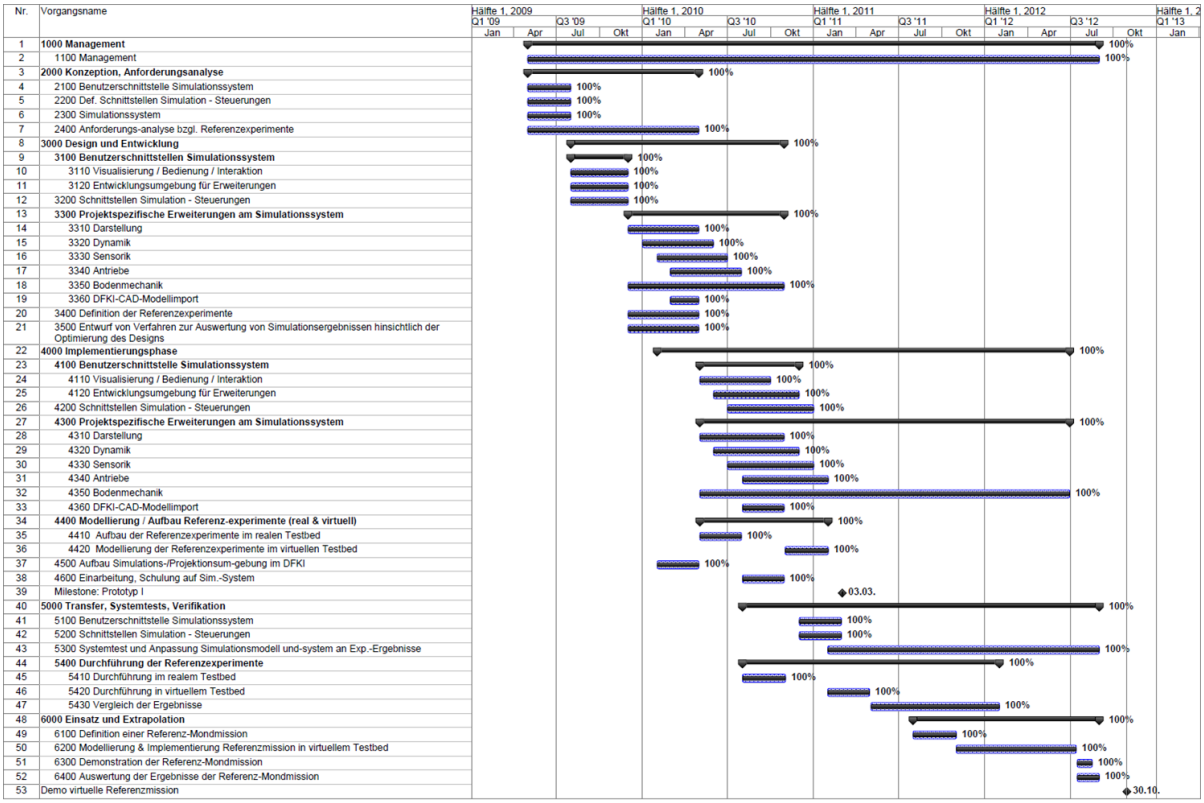


Abbildung 5: *Projektzeitplan zum Projektende*

2 Technische Dokumentation der inhaltlichen Arbeiten

AP3100: Definition und Spezifikation der Benutzerschnittstelle zwischen Simulationssystem und Simulationsanwender

AP3110: Visualisierung / Bedienung / Interaktion

Mit dem DFKI wurde ein Szenario abgestimmt, welches die gewünschte Interaktion mit dem Simulationssystem VEROSIM® definiert. In diesem Szenario wird zunächst der Laufroboter mit Hilfe des Datenhandschuhs und des Trackingsystems in der 3D-Umgebung platziert. Hierbei ist ein 6DOF Tracking essentiell, da so die Lage des Roboters einfach durch die Lage der getrackten Hand beeinflusst werden kann. So wird es möglich, den Roboter auf einen 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.

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 zu integrieren. 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.

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 Flexibität 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 2009 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 (siehe AP4600: Einarbeitung in das und Schulung auf dem Simulationssystem).

Damit steht am DFKI jetzt die Schnittstelle mit größtmöglicher Flexibilität zur Verfügung. Im Gegenzug bedeutet das direkte Entwickeln gegen die C++-Schnittstelle auch die größte Anfälligkeit gegen mögliche Entwicklungsfehler - dennoch scheint dieser Weg der sinnvollste zu sein.

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

Mit dem DFKI wurde ein Kommunikationsprotokoll für das Roboterbetriebssystem MONSTER abgestimmt. Die Kommunikation zwischen der Robotersteuerung und der Simulation soll 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 die Kommunikation auf Grundlage der Socketverbindung synchron implementiert werden. Dies bedeutet, dass die Simulation und auch die Robotersteuerung blockiert wird, bis eine Aktualisierung der jeweiligen Aktuator- und Sensordaten erfolgt ist. Die Datenübertragung soll 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 die bisher definierten Sensoren ist in der Tabelle 1 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 die bisher definierten 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 1: *Socket Kommunikationsprotokoll für die bisher definierten Sensordaten.*

AP3300: Definition der projektspezifischen Erweiterungen des Simulationssystems

AP3310: Darstellung

Mit dem DFKI wurden visuelle Ergänzungen für VEROSIM® definiert, welche das intuitive Verständnis bei der Durchführung einer Simulation verbessern. Diese Ergänzungen werden nachfolgend aufgezählt und kurz beschrieben.

Geometrisches Modell vs. physikalisches Ersatzmodell Es soll möglich sein, zwischen dem geometrischen Modell und dem physikalischen Ersatzmodell umzuschalten. Dies erlaubt eine Vorführung oder Videoaufnahme, während das geometrische Modell aktiv ist. Die Visualisierung des physikalischen Ersatzmodells ermöglicht ein besseres Verständnis des Verhaltens des simulierten Roboters.

Transparenz Es soll möglich sein, die Transparenz der simulierten Komponenten beliebig zu verändern. Dadurch kann eine größere Anzahl an Einzelkomponenten gleichzeitig im Blick behalten werden.

Ghost-Modus Im Ghost-Modus soll eine semi-transparente Version des Roboters aus einem vorherigen Lauf eingeblendet werden, was den visuellen Vergleich zweier Läufe ermöglicht. Es soll entweder der jeweils vorherige Lauf oder ein durch den Benutzer angegebener vorheriger Lauf dargestellt werden können.

Kartendarstellung Während der Steuerung des Roboters in der Simulation soll eine Übersichtskarte der Umgebung eingeblendet werden können, was die gezielte Steuerung des Roboters vereinfacht. Die Karte soll die Möglichkeit bieten, verschiedene Zoom-Stufen einzustellen.

Kraftvektoren Kräfte, die zwischen zwei Komponenten und innerhalb von Motoren und Gelenken auftreten, sollen durch überlagertes Rendern von Vektoren visualisiert werden.

Farbkodierung Einzelne Komponenten des Roboters sollen entsprechend frei wählbarer Parameter eingefärbt werden können. So können Strukturelemente entsprechend ihrer Belastung oder Motoren entsprechend ihrer Stromaufnahme eingefärbt werden. Auf diese Weise können sonst nicht sichtbare Vorgänge visualisiert und z.B. eine zu große Belastung des realen Roboters vermieden werden.

Zustandsgrößen Durch diese Ergänzung soll es möglich sein, interne Zustandsgrößen des Roboters durch eine visualisierte Zahl mit einstellbarere Beschriftung anzeigen zu lassen. Zu den Zustandsgrößen gehören Werte wie Struktur-Belastung, Stromverbrauch, Drehzahl und Sensorwerte.

Diagramme Die genannten Zustandsgrößen sollen auch kombiniert in Form eines Diagramms (Balkendiagramm, Plot) visualisiert werden können.

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 6 links). Die Felsen sind dabei je nach Alter unterschiedlich stark durch Regolith bedeckt. Das führt dazu, dass Felsschichten direkt unter einer feinen Regolith-Schicht liegen können. Dadurch können sich weiche Regolith-Untergründe übergangslos mit harten Steinuntergründen abwechseln. VEROSIM® muss dazu in die Lage versetzt werden, RGB-Höhenbilder einzulesen und daraus die Kraterumgebungen zu generieren. Die Höhenbilder werden aus Laserscans der am DFKI modellierten Krater 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 6 rechts). 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. Die Böden 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.

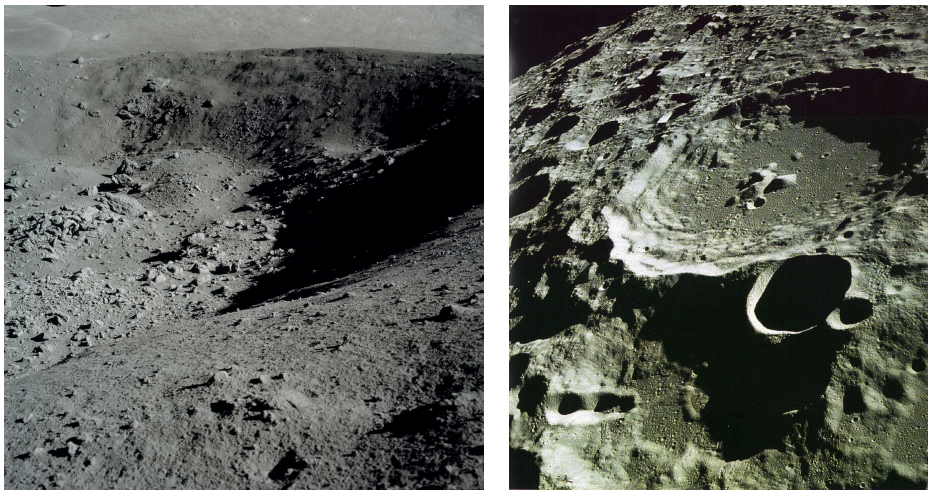


Abbildung 6: *Oberflächenstruktur und Lichtverhältnisse auf dem Mond*

AP3320: Dynamiksimulation

Es bestand schon zu Beginn des Projektes ein ausreichender Funktionsumfang der Starrkörperdynamik, um einfache Mehrkörpermodelle zu simulieren. Grundlagen und Modellierung der Starrkörperdynamiksimulation in VEROSIM® sind beispielsweise in [RJR10b] dokumentiert. Im Verlauf des Projekts kamen spezielle Anforderungen an die Simulation auf in dessen Folge die Starrkörpersimulation durch diverse kleinere Anpassungen erweitert wurde, die in AP4320: Dynamik und den APs 5000 dokumentiert sind.

Spezifikation von Verfahren zur Berücksichtigung von bodenmechanischen Effekten im Gesamtkonzept der Dynamiksimulation

Wenn sich zwei starre Körper in der Dynamiksimulation berühren, entspricht dies einem harten Kontakt ohne Deformation. Daher muss eine Verbindung zwischen der Standard-Starrköperdynamik und der Bodendynamik in VEROSIM® definiert werden. Im Folgenden werden einige notwendige Aspekte einer Schnittstelle zwischen Starrkörper- und Bodendynamiksimulation aufgeführt:

- Veränderliche Boden-Geometrie: Die Boden-Oberfläche sollte ähnlich einer plastischen Verformung veränderbar sein, so dass z.B. ein Oberflächenpolygon am Boden dem Kontakt entsprechend verändert werden kann.
- Auftretende Kontaktkräfte: Natürlich müssen Kontaktnormalen- und Kontaktreibungskräfte ausgetauscht werden können.
- Mehrfache Kontakte: Treten an komplexen Kontaktgeometrien mehrfache Kontaktpunkte auf, sollen diese berücksichtigbar sein.
- 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.

AP3330: Sensorsimulation

In diesem Arbeitspaket wurden die in der Simulation benötigten Sensoren definiert und das - entsprechend den Anforderungen aus AP2300 - benötigte Sensor-Framework spezifiziert. Das Sensor-Framework soll es ermöglichen, alle benötigten Sensoren auf Basis der Referenzexperimente in der Simulation umzusetzen. Hierbei ist es möglich für jeden Sensor die Stärke und Art seines Rauschens anzugeben. Die Auswahl der Sensoren beschränkt sich auf Sensoren, die im Spaceclimber eingesetzt werden oder sich am DFKI verbreitet im Einsatz befinden:

- Inertialnavigationssystem (IMU = Inertial Measurement Unit)
- Gelenk-Winkel-Sensor
- Kraft-Drehmoment-Sensor
- Kontakt- / Drucksensor
- Laserscanner
- Kamera
- 3D-Kamera (PMD = Photonic Mixer Device)

Kamera-Simulation: Die Simulation von Video- und Fotokameras soll auf Grundlage einer Mehrfenster-Rendertechnologie realisiert werden. Diese Technologie schafft zunächst die Möglichkeit, eine Simulationsszene aus mehreren Blickwinkeln gleichzeitig zu betrachten. Im nächsten Schritt sollen die zusätzlichen Blickwinkel dann dazu genutzt werden, Kamerabilder einer simulierten Kamera zu synthetisieren. Abbildung 7 zeigt einen Screenshot der Simulation des Scarabeus Roboters in VEROSIM®, die gleichzeitig aus vier Blickwinkeln betrachtet wird.

Als Teil des vom DLR geförderten Projekt FastMap (DLR-Förderkennzeichen 50RA1034) sollen im Arbeitspaket FastMap AP 3.6.1 Ergebnisse von FastMap zu VirtualCrater transferiert werden. Im Rahmen der Vorarbeiten zu FastMap wurden erste Überlegungen zur Kamerasimulation angestellt. Man benötigt zur Simulation

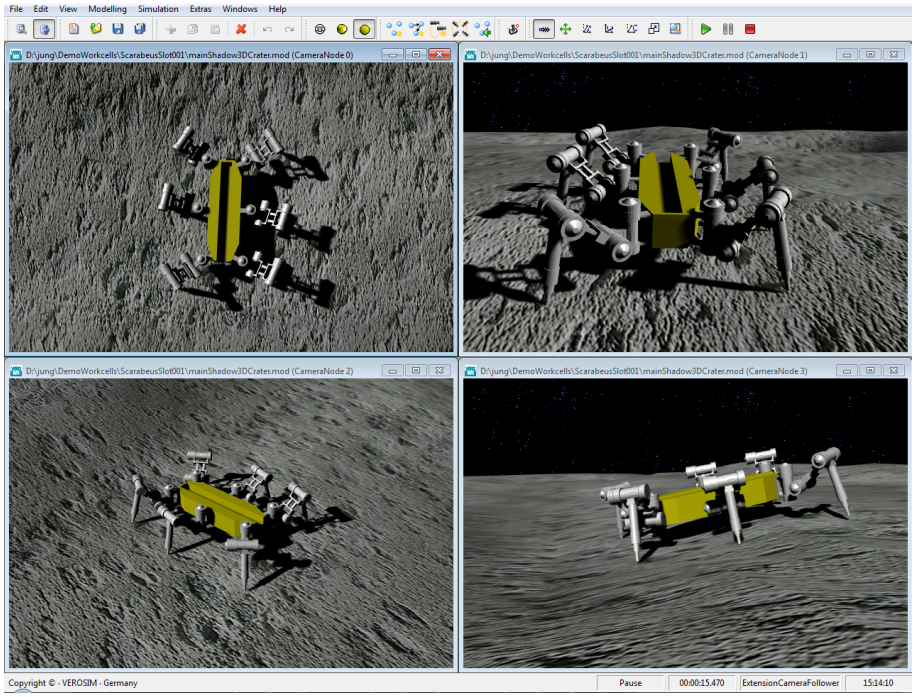


Abbildung 7: *Mehrfenster-Ansicht der selben Simulation in VEROSIM®*

der Kamera die intrinsischen Parameter, die durch eine Kalibrierung bestimmt oder vom Hersteller geliefert werden [Tsa86]. Bei CCD- oder CMOS-Bildsensoren zu berücksichtigende Effekte sind temperaturabhängiges Rauschen, Hotpixel und Übersprechen benachbarter Pixeln. Je nach Anforderung und benötigter Genauigkeit kann ein Schrot- oder Photonenrauschen durch einen Photonmapper simuliert werden. Es muss überprüft werden ob dieses physikalisch korrekte Verhalten anforderungsgerecht in Echtzeit zu realisieren ist [Joz02]. Desweiteren sind optische Effekte, wie Reflektionen und Verzerrung des Lichts im Objektiv zu berücksichtigen. Blendenflecken bzw. Lens flares können mit moderner Computergraphik effizient physikalisch plausibel dargestellt werden. Dies gilt auch für das Überstrahlen („Bloom“), das auftritt wenn sich eine Lichtquelle hinter einem Teil eines Objekts oder einem kleinen Objekt befindet [YIMS08].

Laserscanner-Simulation: Die Simulation eines Laserscanners innerhalb einer 3D-Simulationsumgebung geschieht typischerweise mit sog. Raytracing-Algorithmen. Um diese äußerst rechenaufwändigen Algorithmen möglichst effizient ausführen zu können, wurde im Rahmen des vom DLR geförderten Projekts SELOK (DLR-Förderkennzeichen 50RA0911) auf der Grundlage von VEROSIM® eine Implementierung entwickelt, mit der besonders effizient entsprechende Tiefeninformationen aus der Simulation gewonnen werden können.

AP3340: Simulation der Antriebe

Das Roboterbetriebssystem MONSTER soll in einen kompletten Regelkreis eingebettet werden, bei dem die von MONSTER vorgegeben Sollgelenkwinkelstellungen auf das zu entwickelnde Motormodell aufgeschaltet werden und sämtliche notwendigen Sensorinformationen an MONSTER zurück gekoppelt werden. Offen ist dabei insbesondere noch die Frage, wie Motormodell und Starrkörperdynamiksimulation miteinander verknüpft werden können. Um die Machbarkeit zu untersuchen und erste Erfahrungen zu sammeln wurde ein Regelkreis aus PID-Regler, Gleichstrommotor und der Starrkörperdynamiksimulation in VEROSIM® implementiert (siehe Abb. 8und 9). Prinzipiell können Regler und Motormodell hierbei aber beliebig komplex werden. Die kraftbasierte Dynamiksimulation erfordert deutlich kleinere Zeitschritte, sodass die Simulation wie erwartet nicht mehr echtzeitfähig ist. Die Komponenten "Regleründ "Motorßind mit generischen Ein- und Ausgängen versehen und können über die graphische Benutzeroberfläche von VEROSIM® mit der Dynamik verschaltet werden.

Auf diese Weise können einzelne Komponenten flexibel ausgetauscht werden. Im nächsten Schritt wird eine konkrete Teilkomponente des SpaceClimbers in VEROSIM® modelliert und die Simulationsergebnisse mit Realexperimenten abgeglichen.

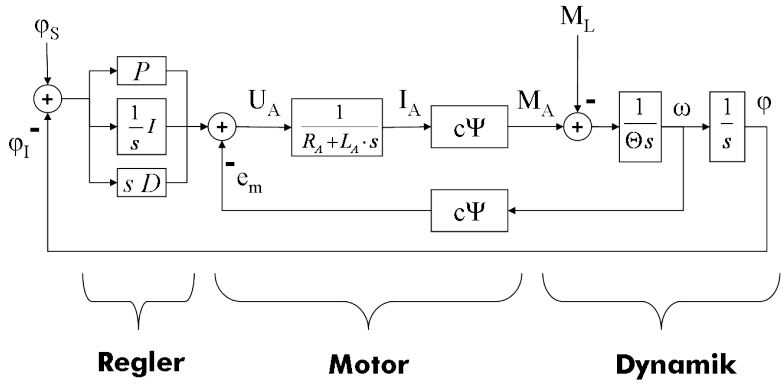


Abbildung 8: *Regelkreis aus PID-Regler, einfachem Gleichstrommotor und Starrkörperdynamiksimulation.*

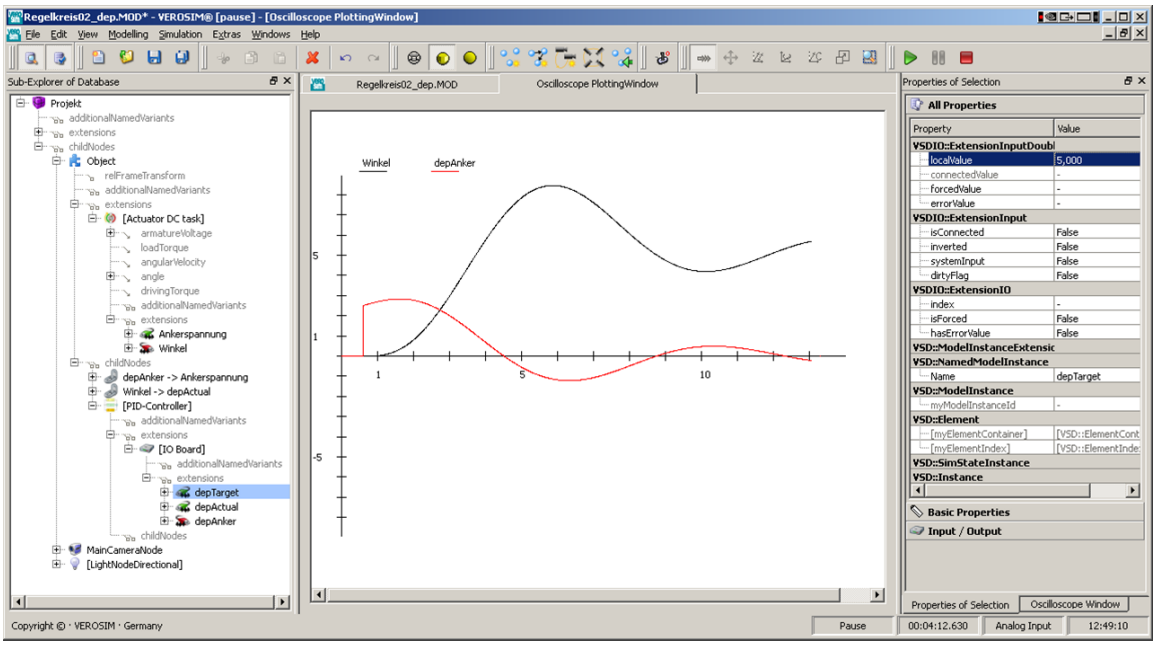


Abbildung 9: *Modellierung des Regelkreises aus Abb. 8 in VEROSIM®.*

AP3350: Bodenmechanik

Im Bereich der radgetriebenen Systeme existieren viele Arbeiten, die sich mit der Bodenmechanik-Simulation beschäftigen (z.B. [PSE04], [MRP+04], [BLB05] oder [YH02]). Die Simulation der Bodenmechanik für Laufroboter ist derzeit noch ein relativ offenes Feld. Daher sollen zunächst verschiedene Ansätze parallel verfolgt und später verglichen werden:

- Ansatz 1: Fuß-Boden-Kontakt-Modell basierend auf der Bekker Theorie [Bek96]
- Ansatz 2: Bodenmechanik basierend auf Neuronalen Netzen [LKC+09]
- Ansatz 3: Bodenmechanik basierend auf zellulären Automaten [RJR10a]

Die ersten beiden Ansätze sollen auf Seiten des DFKI zur Modellierung der Bodenmechanik näher betrachtet werden. Bei RIF/MMI soll eine Schüttgutsimulation basierend auf zellulären Automaten (siehe [RSJR09]) weiterentwickelt und dahin gehend untersucht werden, ob sie auch als Untergrund für einen Laufroboter dienen kann. Außerdem muss die Frage untersucht werden, wie unterschiedliche Bodenmechanikmodelle mit der Starrkörperdynamik, die zur Simulation des eigentlichen Roboters eingesetzt wird, verkoppelt werden können.

Bodenmechanik basierend auf zellulären Automaten

Das hier eingesetzte Verfahren benutzt zelluläre Automaten als Simulationsmechanismus. Die Boden-Oberfläche wird als „Heightfield“ (deutsch: Höhenfeld) abgebildet, also als kontinuierliche Höhenwerte über einem zweidimensionalen Netz aus äquidistanten Knoten. Jeder dieser Knoten agiert als zellulärer Automat und interagiert nur mit seinen direkt benachbarten Knoten, in dem er einen Satz Regeln befolgt. Diese Regeln sichern die Einhaltung eines maximalen Böschungswinkels und einer maximalen Krümmung der Sand-Oberfläche. Wenn die Höhendifferenz zwischen zwei benachbarten Knoten größer ist als es der maximale Böschungswinkel erlaubt, wird „Material“ zwischen den Knoten ausgetauscht, so dass die Regel wieder erfüllt ist. Analog dazu gilt, ist die Krümmung entlang drei benachbarter Knoten größer als es die maximale Krümmung erlaubt, wird sie durch „Materialaustausch“ reduziert.

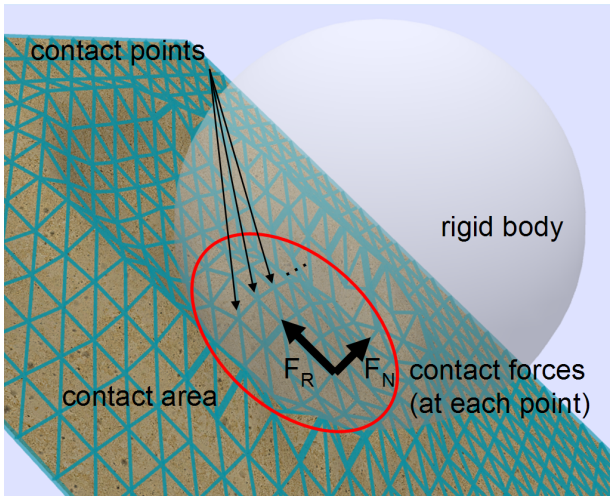


Abbildung 10: *Darstellung des Bodenmechanikmodells auf Basis zellulärer Automaten*

Die Menge der notwendigen Berechnungen wird minimal gehalten, indem immer eine Liste der aktiven Knoten vorgehalten wird und alle inaktiven Knoten während der Simulation ignoriert werden. Befindet sich das System in einem gültigen Zustand, d.h. maximaler Böschungswinkel und maximale Oberflächenkrümmung werden an jeder Stelle der Oberfläche eingehalten, sind gar keine Berechnungen notwendig. Erst wenn das System lokal durch externe Einflüsse aus der Ruhelage gebracht wird, werden alle betroffenen Knoten aktiviert. Wird der Materialinhalt eines Knotens (= seine Höhe) durch einen benachbarten Knoten verändert, wird auch er aktiviert. Ist ein aktiver Knoten stabil, d.h. die Regeln sind für diesen Knoten erfüllt, so wird er deaktiviert.

Diese Implementierung ist äußerst effizient im Hinblick auf die notwendige CPU-Zeit und bietet ob seiner lokalen Sichtweise einen natürlichen Weg zur Parallelisierung für zukünftige Performanzoptimierungen. Einwirkende Kräfte werden durch sog. „kraftäquivalente Höhen“ auf jedem Knoten berücksichtigt, wodurch interne Regelverletzungen und externe Störgrößen sehr einfach und gleichzeitig berücksichtigt werden können. Das physikalische Verhalten eines Kontaktes wird durch Abfragen lokaler Reibungs- und Kontaktnormalenkräfte in allen Knoten entlang einer Kontaktfläche approximiert (vgl. Abbildung 10). Diese Kräfte werden zur Integration dieser Bodenmechaniksimulation mit der Starrkörperdynamik herangezogen.

Kopplung von Bodenmechanik- und Starrkörperdynamiksimulation

Eine realitätsnahe Bodenmechaniksimulation ist für das Projekt Virtual Crater von essenzieller Bedeutung. Doch Bodenmechaniksimulation allein reicht nicht aus, um einen Laufroboter in lunarem Gelände zu simulieren, es muss ein gemeinsames Modell für Starrkörperdynamik und Bodenmechanik formuliert werden.

Da im Simulationssystem VEROSIM® bereits eine Komponente für die Starrkörperdynamik existiert, erscheint es sinnvoll, eine Möglichkeit zu schaffen, um nahezu beliebige Bodenmechanikmodelle mit der Starrkörperdynamik zu koppeln. Eine solche Möglichkeit scheinen nach dem derzeitigen Wissensstand sogenannte „Gluing“-Strategien („Aneinanderkleben“) zu sein.

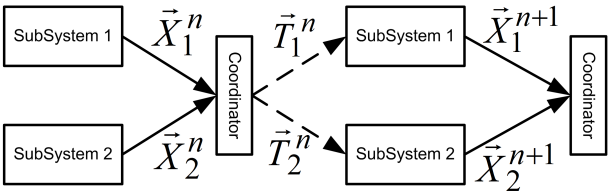


Abbildung 11: *Darstellung einer klassischen T-T-Gluing Strategie*

Gluing bedeutet das Zusammenfügen von zwei Simulationsmodellen durch einen iterativen Algorithmus, der direkt die nativen Ein- und Ausgabegrößen der jeweiligen Verfahren berücksichtigen kann. Die bis heute eingeführten Strategien unterscheiden sich in der Art der ausgetauschten Daten. Dabei werden alle kinematischen Informationen, also Positionen und Orientierungen, Geschwindigkeiten sowie Beschleunigungen als „X-Type“-, alle Kraft-Größen als „T-Type“-Informationen bezeichnet. Eine mögliche Variante ist eine „T-T“-Strategie, Abbildung 11 zeigt eine schematische Darstellung des Ablaufs. Dieses Verfahren läuft wie folgt ab: Zunächst werden beide Subsysteme einmal unabhängig voneinander gelöst. Das Ergebnis ist jeweils eine kinematische Konfiguration, also je eine X-Type Information, im Beispiel wären das die Höhe der Sand-Oberfläche und die Position eines Fußes. Eine koordinierende Instanz, in der Abbildung der Koordinator, nimmt beide kinematischen Informationen entgegen und wertet Zwangsbedingungen aus, die zwischen beiden Systemen erfüllt sein sollen. Im Beispiel wäre das: Fuß und Sand können nicht den gleichen Raum einnehmen, Sand muss verdrängt werden. Abhängig vom Ergebnis der Auswertung der Zwangsbedingung bestimmt der Koordinator eine Kraftinformation, die gleichermaßen in beide Subsysteme eingebracht wird. Das Prinzip Actio = Reactio erfordert, dass auf beide Systeme der gleiche Vektor nur mit unterschiedlichem Vorzeichen aufgebracht wird. Nun werden beide Subsysteme erneut gelöst und die nächste Iteration beginnt. Die Iteration endet, wenn die vom Koordinator aufgebrachten Kräfte konvergieren bzw. die Auswertung der Zwangsbedingung ergibt, dass sie in ausreichendem Maße erfüllt ist.

Um das beschriebene Verfahren im Kontext von Virtual Crater und VEROSIM® zu testen, wurde eine T-T-Gluing Strategie prototypisch implementiert. Sie verbindet die in VEROSIM® vorhandene Starrkörperdynamikkomponente mit der im Rahmen von Virtual Crater bereits experimentell realisierten Bodenmechaniksimulation auf der Basis zellulärer Automaten. Abbildung 13 zeigt den dynamisch simulierten Scarabeus-Roboter auf der Simulation einer Sandoberfläche. Die abgebildete Simulation ist noch nahe-echtzeitfähig, jedoch wird das Gluing hier mit nur einer einzigen Iteration ausgeführt. Um eine bessere Konvergenz des Gluing-Algorithmus und damit eine höhere Genauigkeit der Gesamtsimulation zu erhalten, werden mehrere Iterationen notwendig sein und die Echtzeitfähigkeit für die Simulation eines vollständigen Laufroboters kann u.U. nicht mehr erreicht werden. Das realisierte Verfahren basiert auf einer Besonderheit der

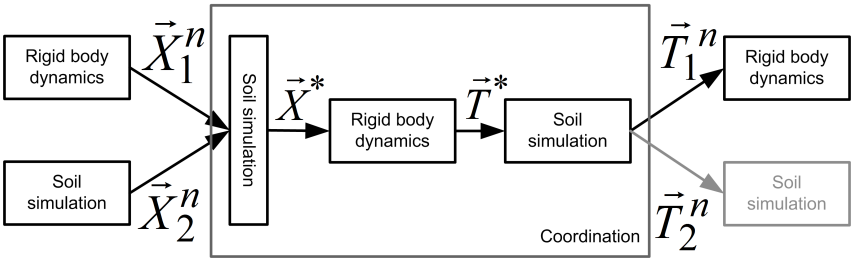


Abbildung 12: *Gluing Strategie zur Verbindung von Starrkörperdynamik und automatenbasierter Bodenmechaniksimulation*

Integration beider Verfahren: Da das Bodenmechanikmodell prinzipbedingt nur schwer gute Schätzungen für Kontaktkräfte bestimmen kann, ist unter Umständen beim Gluing eine hohe Anzahl an Iterationen notwendig. Um diese zu minimieren, nutzt der Gluing-Ansatz als eine erste Schätzung der Kontaktnormalenkräfte die Kontaktkräfte, die bei einem starren Kontakt zwischen Oberfläche und Fuß auftreten würden. Abbildung 12 beschreibt das Vorgehen analog zu Abbildung 11. Weitergehende Details des Verfahrens werden in [RJR10c] beschrieben.

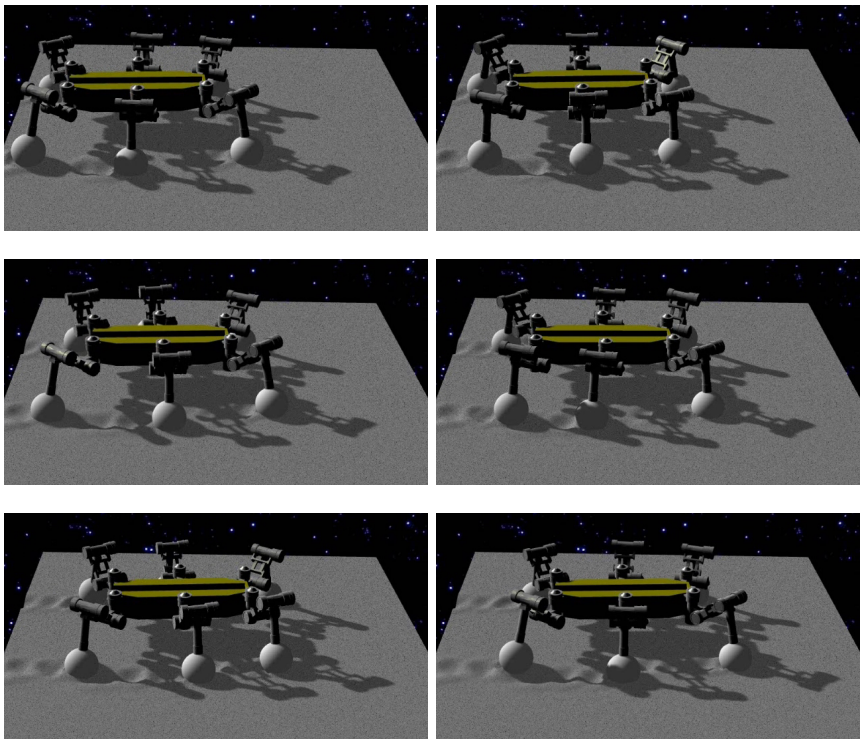


Abbildung 13: *Vorläufiges Ergebnis der Umsetzung einer Gluing Strategie: Ein Starrkörper-Dynamikmodell des Scarabeus-Roboters läuft in VEROSIM® über eine Sandbodensimulation auf Basis zellulärer Automaten*

Gluing scheint - gerade im Hinblick auf die experimentelle Analyse von Simulationsverfahren - ein äußerst flexibler und leistungsfähiger Ansatz zu sein. Besonders interessant ist er, weil auch völlig neue und sehr unterschiedliche Modelle und Formulierungen zur Bodenmechaniksimulation mit der bestehenden Starrkörperdynamik verknüpft werden können. Die prototypisch entwickelte Gluing-Schnittstelle zur Starrkörperdynamiksimulation wurde dann für die weiteren Entwicklungen weiterentwickelt und für den flexiblen Zugang offengelegt.

Konzept und prototypische Realisierung wurden in [RJR10c] veröffentlicht.

AP3360: DFKI-CAD-Modellimport

In diesem Arbeitspaket wurde ein Verfahren definiert, um ein CAD-Modell direkt aus der entsprechenden CAD-Software mit möglichst geringem Aufwand in die Simulationsumgebung in VEROSIM® zu portieren. Es wurde entschieden, dass die Umsetzung dieser Definitionen durch die Erweiterung eines AddIns direkt in der CAD-Anwendung am günstigsten zu realisieren ist. Die Partner haben sich auf Autodesk® Inventor® als Modellierungswerkzeug für Roboter-CAD-Daten geeinigt. Der Datentransfer zwischen Autodesk® Inventor®und VEROSIM® wird mittels eines Inventor®-AddIns realisiert. Dieses AddIn lädt VEROSIM® als Bibliothek zu Autodesk® Inventor® hinzu und stellt Funktionen für den Datentransport bereit. Gewünscht ist die Analyse und Abbildung von Inventor®-Constraints in kinematische und dynamische Strukturen in VEROSIM®, speziell für den Anwendungsfall „Virtual Crater“ und die hier verwendete Modellierungsstrategie. Inventor® bietet die Möglichkeit, Constraints zu formulieren, die nicht direkt technisch realisierbaren mechanischen Zusammenhängen entsprechen, z.B. „Ebenengelenke“ (zwei Körper liegen in einer unendlichen Ebene). In VEROSIM® hingegen werden typischerweise Sätze von Constraints gemeinsam definiert, so dass sie real existierenden Gelenktypen entsprechen. (z.B. Drehgelenk, Lineargelenk). Hier gilt es, einen Weg zu finden, die in diesem Projekt verwendeten Constraints aus Inventor® in VEROSIM® abzubilden. Dies könnte dadurch geschehen, dass die Inventor®-Constraint-Mengen dahingehend analysiert werden, welche real existierenden Gelenktypen sie darstellen. Alternativ könnten die abstrakten, in Inventor® üblichen Constrainttypen auch in VEROSIM® nachgebildet werden. Darüber hinaus sollten Beleuchtungseinstellungen aus Inventor®bestmöglich nach VEROSIM® übertragen werden.

AP4100: Implementierung Benutzerschnittstelle zum Simulationssystem

AP4110: Visualisierung / Bedienung / Interaktion

Visualisierung

Die Darstellung und Protokollierung von Sensorwerten und Simulationsgrößen, die innerhalb der Input/Output-Struktur von VEROSIM® zur Verfügung gestellt werden, wurde verbessert und vereinfacht. Im VEROSIM®-Database-Explorer kann eine neue Extension *IO data logger* oder *IO oscilloscope* erzeugt werden. Die Extensions verfügen über Properties *selectedOutputs* und *selectedSensors* (nur IO data logger), in die Verweise auf Outputs und Sensoren eingetragen werden können. Diese Outputs und die Outputs der eingetragenen Sensoren werden dann mit der Simulationszeit in eine *.csv-Datei geschrieben oder in einem Oszilloskop-Fenster dargestellt (siehe Abb. 14 und 15).

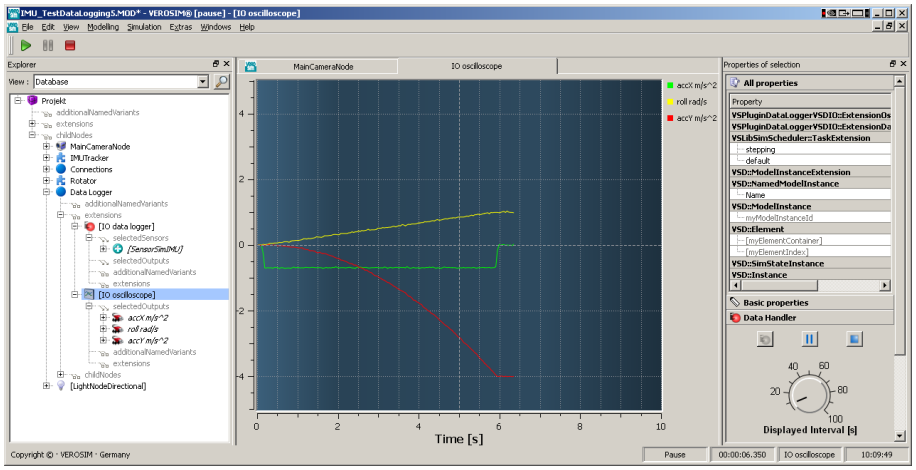


Abbildung 14: *Darstellung von Sensorwerten in einem Oszilloskop bei laufender Simulation.*

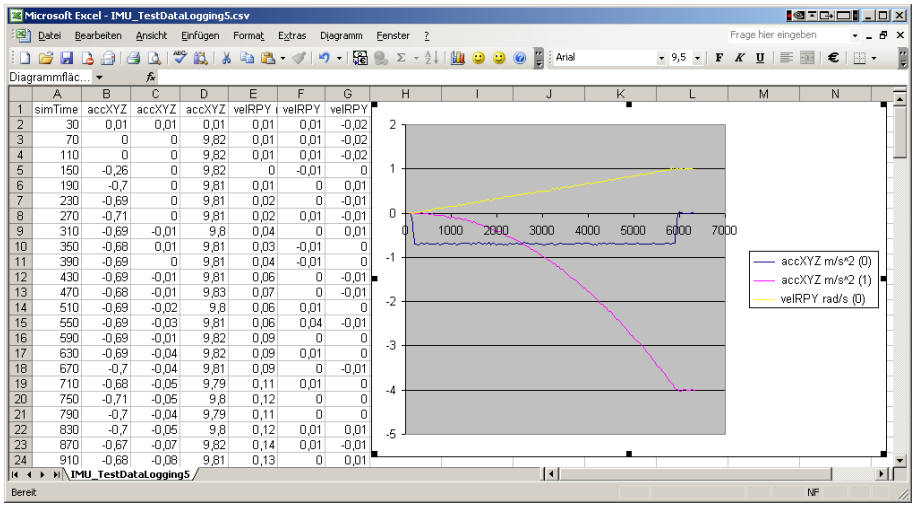


Abbildung 15: *In einer *.csv-Datei protokollierte Sensorwerte zur nachträglichen Analyse.*

Interaktion

Zur direkten Interaktion zwischen Anwender und laufender Simulation wurde eine Metapher „Elektromagnet“ realisiert. Hierbei handelt es sich um einen Körper innerhalb der Mehrkörpersimulation, der direkt durch den Anwender, z.B. via SpaceMouse oder über das Trackingsystem, gesteuert werden kann. Ist er aktiviert, bindet er jeden Körper, mit dem er kollidiert, durch ein in allen 6 Freiheitsgraden parametrierbares Feder-Dämpfer-System an sich. Diese Metapher erlaubt dem Benutzer damit die direktest mögliche Interaktion mit der Simulation. Abbildung 16 links zeigt, dass auch das gelenkgekoppelte System selber direkt manipuliert werden kann.

Der Datenhandschuh „Immersion CyberGlove II“, der am MMI vorhanden ist, wurde an VEROSIM® angebunden. Da der am DFKI verwendete Datenhandschuh vom gleichen Typ ist, ist die Schnittstelle auch dort verwendbar. Für den Handschuh können beliebige Gesten anhand minimaler und maximaler Gelenkwinkelstellungen definiert werden. Befindet sich der Handschuh im Toleranzbereich einer bestimmten Geste, wird ein digitaler Ausgang auf 1 geschaltet. In Verbindung mit der Metapher „Elektromagnet“ kann man damit also aktuell Starrkörper (Roboter, Steine, etc.) verschieben, greifen und loslassen (siehe Abb. 16 rechts). Dieses Werkzeug bietet jedem Benutzer die Möglichkeit, besonders intuitiv und direkt mit der laufenden Simulation zu interagieren. Weitere Gesten, wie z.B. das Zeigen auf eine Zielposition, können über die grafische Oberfläche definiert werden.

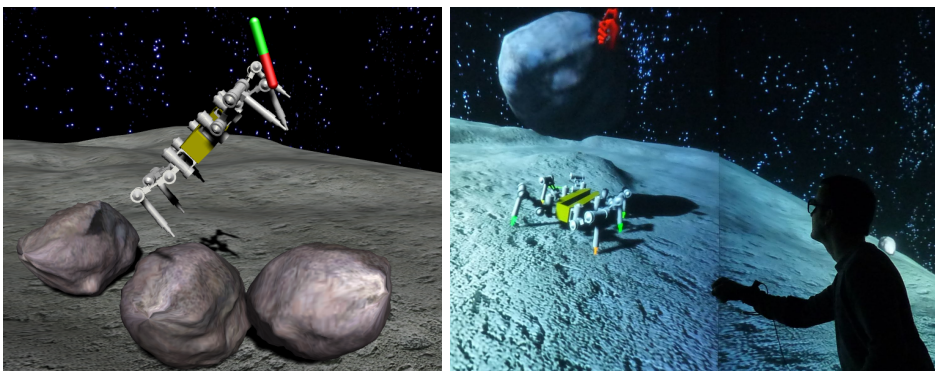


Abbildung 16: *Links: Direktes Manipulieren von Teilen der Mehrkörpersimulation mit der Magnet-Metapher*
Rechts: Direkute Interaktion mittels Datenhandschuh zwischen Experimentator und 3D-Modell in einer Stereoprojektionsumgebung

AP4120: Entwicklungsumgebung für Erweiterungen

Dem DFKI wurde eine Entwicklungsumgebung für VEROSIM® bereit gestellt. Diese besteht aus einem vorkompilierten VEROSIM®, den Header-Dateien der zentralen Bibliotheken und einigen Beispielplugins anhand derer der Zugriff auf die Datenbasis und die graphische Oberfläche von VEROSIM® demonstriert wird. Erleichtert wird die Einrichtung durch das weiter entwickelte Software-Werkzeug „SolutionExplorer“ (siehe Abb. 17). Damit lassen sich Abhängigkeiten und Einstellungen für die Kompilierung der verschiedenen Plugins verwalten und schnell „Solutions“ (*.sln) für Microsoft Visual Studio oder Projektdateien (*.pro) für Qt Creator erstellen.

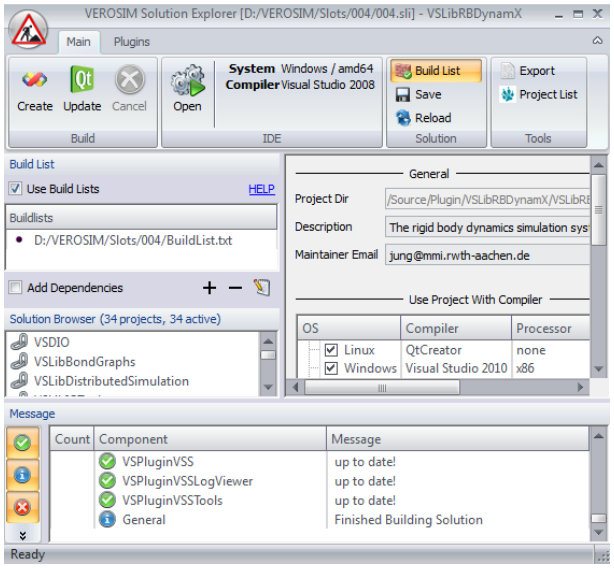


Abbildung 17: *SolutionExplorer zum Erstellen und Verwalten von Visual Studio Solutions*

Im gesamten Verlauf des Projekts konnte das DFKI anhand der zur Verfügung gestellten Entwicklungsumgebungen eine Reihe von Erweiterungen für das Simulationssystem implementieren (siehe Tab. 2).

Tabelle 2: *Liste von VEROSIM-Erweiterungen, die im Laufe des Projekts vom DFKI entwickelt wurden.*

VSPluginVCDiscreteBlocks	Plugin zur Simulation von Systemkomponenten basierend auf diskretisierten Differenzialgleichungen (konkret: Erster Entwurf des SpaceClimber-Antriebs)
VSPluginVCAutomatedExperiment	Plugin zur automatisierten Durchführung von virtuellen Versuchsreihen
VSPluginVCSignalGenerator	Generierung von Signalen zu Test- und Analysezwecken
VSPluginVCLookat	Steuerung des Sherpa-Scheinwerfers
VSPluginVCMultiResHeightMap	Management und Adaption veränderlichen Höhendaten mit variabler horizontaler Auflösung
VSPluginVCNeuroTerraDynamics	Bodenmechanikmodell auf Basis Neuronaler Netze
VSPluginVCRealTimeChecker	Spezialisiertes Performance Benchmarking Werkzeug für die ganzheitliche Simulation des Virtual Crater Szenarios
VSPluginVCSlippageSensor	Virtueller Schlupfsensor zur Analyse der Traktion des SpaceClimbers
VSPluginVCSCJoint	Modell für den Antrieb des SpaceClimbers

Bibliothek zur Modellierung technischer Systeme

Für VEROSIM® wurden ein neues Paar aus Bibliothek und Plugin (VSLibControlSystems, VSPluginControl-

Systems) zur Modellierung technischer Systeme implementiert. Ausgehend von einem allgemeinen Zustandsraummodell

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned}$$

(mit Zustandsvektor x, Eingangsgrößen u, Ausgangsgrößen y, Systemmatrix A, Steuermatrix B, Beobachtungsmatrix C und Durchgangsmatrix D) wurde eine Basisklasse für linear zeitinvariante MIMO[1]-Systeme erstellt, welche die zugrunde liegende Differentialgleichung zur Verfügung stellt. Dadurch können verschiedene Integrationsverfahren als separate Klassen angelegt und ausgewählt werden. Von der MIMO-Basisklasse können dann beliebige Systemkomponenten abgeleitet werden, wie z.B. ein PID-Regler oder ein Gleichstrommotor, deren Parameter dann auf die Matrizen des Zustandsraummodells abgebildet werden. Diese Komponenten sind in den graphischen Editor für I/O-Netzwerke integriert worden (siehe AP4310). Auf diese Weise können regelungstechnische Probleme auch direkt in VEROSIM® bearbeitet werden, ohne dass man bereits für erste Tests auf Code-Exporte anderer Simulationssysteme angewiesen ist.

VEROSIM®-Wrapper-Klassen für C-Code-Export externer Simulationssoftware

Das VEROSIM®-Plugin *VSPluginVCActuators*, welches bereits zur Implementierung von Motormodellen in VEROSIM® aufgesetzt wurde, wurde um eine Klasse *TaskExtensionSubModel20Sim* erweitert, die beliebige Simulations-Untermodelle, wie sie vom C-Code-Export des Simulatiossystems 20-sim (`http://www.20sim.com/`) erzeugt werden, als VEROSIM®-Extensions mit entsprechenden Inputs und Outputs instantiiert. Nun können in 20-sim entworfene und optimierte elektrische und mechanische Modelle schnell nach VEROSIM® portiert und dort in die Simulation eingebunden werden.

Modularisierung und Optimierung der API zur Starrkörperdynamiksimulation

Abb. 18 zeigt die Integration des Starrkörperdynamik-Moduls mit einem Bodenmechanik-Modul unter Ausnutzung der modularen Softwarestruktur und des Metadatensystems der Echtzeitdatenbank *VSD* von VEROSIM®.

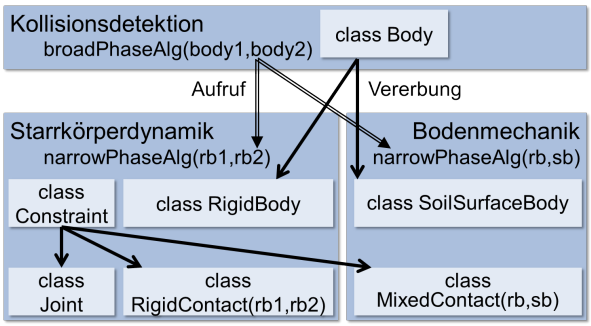


Abbildung 18: *Integration des neuen Softwaremoduls Bodenmechanik mit der bestehenden Starrkörperdynamik.*

Kollisionsdetektion und -Behandlung erfolgt üblicherweise in mehreren Phasen, so wird in der *Broad Phase* durch einfache geometrische Abfragen die Anzahl der potentiell kollidierenden Körper stark reduziert. In der *Narrow Phase* erfolgt die exakte Kollisionserkennung und die Bestimmung der Kontaktpunkte. Das Modul *Kollisionsdetektion* berechnet die *Broad Phase* für beliebige geometrische Körper und bietet die Möglichkeit, zur Laufzeit neue *Narrow Phase*-Algorithmen für abgeleitete Klassen zu registrieren, an die eine Paarung dann weiter gegeben wird. Hier werden Paarungen von Starrkörpern (engl. Rigid Bodies) vom Modul Starrkörperdynamik behandelt, während Paarungen von Starrkörper und dem neu definierten *SoilSurfaceBody* im neuen Bodenmechanik-Modul behandelt werden. Ausgehend von einer Basisklasse zur allgemeinen Formulierung von Zwangsbedingungen (engl. Constraints) lassen sich dann für die Kontaktsituation spezifische

[1] Multiple Input Multiple Output

Zwangsbedingungen aufstellen und geschlossen in einem LCP (Linear Complementarity Problem) lösen. Aus Stabilitäts- und Performance-Gründen ist es günstiger, geschwindigkeitsbasierte Zwangsbedingungen in das LCP einzuführen, statt einfach Kräfte auf einen Starrkörper aufzubringen. Wenn das Bodenmechanik-Modell rein dissipativ ist, ist dies möglich, indem man Zwangsbedingungen formuliert, die im Kontaktpunkt das Verschwinden der Relativgeschwindigkeit fordern. Die auf den Starrkörper wirkenden Kräfte lassen sich dann kontrollieren, indem sie als Maximalkräfte zur Einhaltung der Zwangsbedingung eingesetzt werden. Dabei werden tatsächlich immer die maximalen Kräfte aufgebracht, solange der Starrkörper nicht vollständig abgebremst wurde. Ist der Starrkörper in Ruhe, verschwinden auch die dissipativen Kräfte und es stellt sich ein Gleichgewicht zwischen Gewichtskraft und Kontaktkraft ein (ruhender Kontakt). Eventuelle vernachlässigbare Abweichungen sind nur auf die endliche Zeitschrittweite zurückzuführen.

AP4200: Schnittstellen zwischen realer Steuerung und Simulation

Interne Schnittstellen

Innerhalb von VEROSIM® wird für den Datenaustausch zwischen verschiedenen Simulationskomponenten ein generalisiertes Ein-/Ausgabe-Framework verwendet (VEROSIM®-Bibliothek VSDIO). Jede Komponente wird mit einer E/A-Karte ausgestattet, an die Ein- und Ausgänge für verschiedene Datentypen angehängt werden können. Diese können dann über einen grafischen Editor erzeugt und verbunden werden, um ganze E/A-Netzwerke zu erstellen.

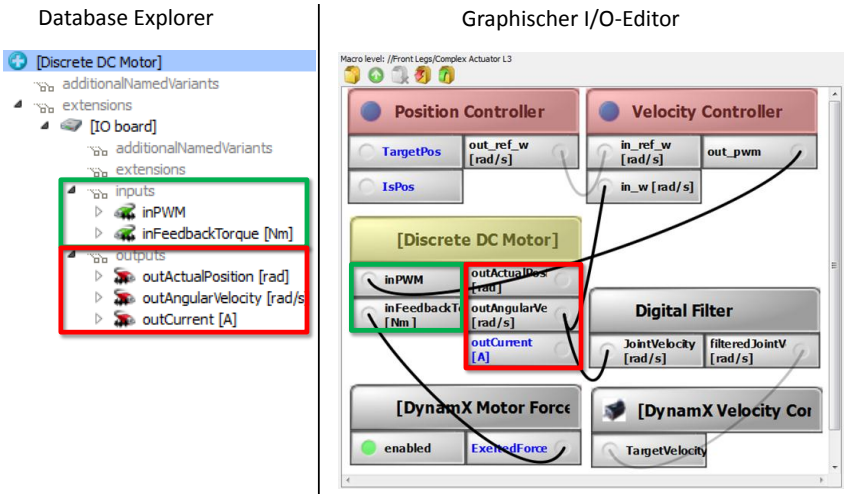


Abbildung 19: *Verwaltung von Schnittstellen innerhalb von VEROSIM®*

Abb. 19 zeigt beispielhaft links die Ansicht der Datenbasis mit einem Motormodell (*Discrete DC Motor*) der mit Eingängen (grün umrandet) und Ausgängen (rot umrandet) ausgestattet ist. Der grafische Editor auf der rechten Seite stellt Objekte mit Ein- und Ausgängen kompakt dar und ermöglicht die einfache Verwaltung von Verbindungen. Netzwerke können auf beliebig tiefen Makroebenen ineinander geschachtelt werden, so enthalten die Controller (rot hinterlegt) weitere E/A-Komponenten.

Externe Schnittstellen

Die Kommunikation mit einer externen Komponente lässt sich leicht in dieses E/A-Framework integrieren. Das Datenformat, in dem sich die Kommunikation mit verschiedenen externen Komponenten unterscheidet, wurde dabei vom Code der Schnittstelle separiert, sodass diese für verschiedenste externe Komponenten verwendet werden kann.

Für die Steuerung und Regelung der DFKI-Roboter SpaceClimber und Scarabaeus liegen verschiedene Versionen des vom DFKI entwickelten Roboter-Controller M.O.N.S.T.E.R. vor. Im einfachsten Fall kann der Scarabaeus durch Vorgabe von 18 Soll-Gelenkwinkeln und der Rückgabe der entsprechenden Ist-Gelenkwinkel

geregelt werden. Eine formal syntaktische Beschreibung des Kommunikationsprotokolls wird in Form einer XML-Datei hinterlegt. Da diese 18 Kanäle den gleichen Datentyp haben, können sie zu einer Kanalgruppe zusammengefasst werden. Erzeugt man nun im Simulationsmodell eine „ExtensionExternalComponent“ mit Verweis auf das Kommunikationsprotokoll, so werden automatisch die jeweils 18 Ein- und Ausgänge erzeugt (siehe Abb. 20).

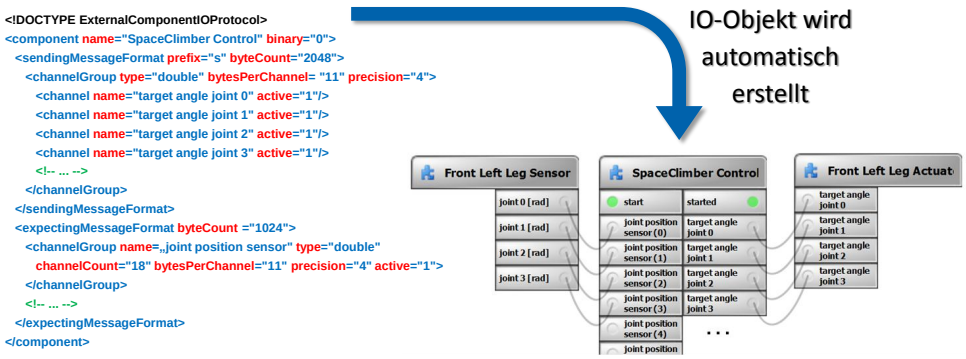


Abbildung 20: *Aus der Definition des Schnittstellenprotokolls in XML wird automatisch ein Objekt erstell, das die Kommunikation mit externen Komponenten auf das interne E/A-Konzept abbildet*

Während der Simulation werden nun die an den Eingängen anliegenden Werte entsprechend des Kommunikationsprotokolls formatiert und auf einen verallgemeinerten IO-Port geschrieben. Das kann z.B. wie in diesem Fall ein TCP-Socket sein. Der M.O.N.S.T.E.R.-Controller empfängt die Sensorwerte und gibt Sollwerte für die Gelenke zurück, die wiederum auf die Ausgänge im Simulationssystem geschrieben werden. Auf diese Weise erhält der Benutzer im Simulationssystem ein Objekt, das sich wie die angebundene Komponente verhält und dabei über das E/A-Konzept mit allen anderen Simulationskomponenten verbunden werden kann.

Co-Simulation

Dieselbe Schnittstelle kann auch für die Co-Simulation mit anderen Simulationssystemen wie z.B. Matlab/-Simulink verwendet werden (siehe Abb. 21).

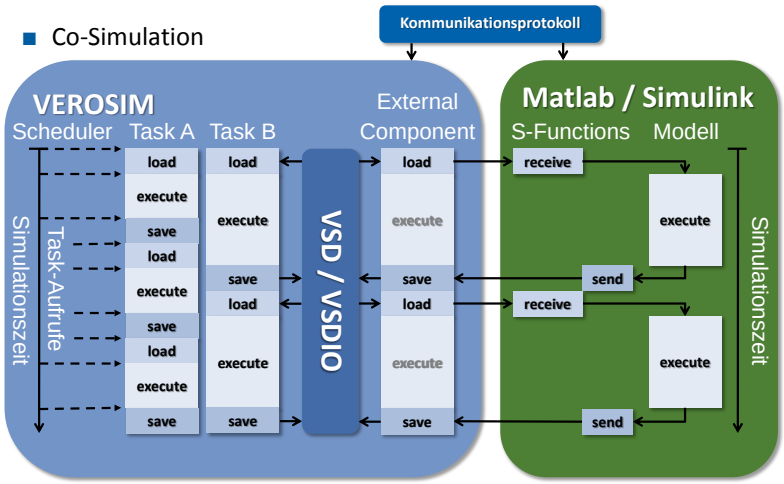


Abbildung 21: *Co-Simulation mit Matlab/Simulink*

In diesem Fall wurden für Simulink zwei so genannte „S-Functions“ zum Senden und Empfangen implementiert. Sowohl die S-Functions, als auch das VEROSIM®-Objekt “Extern“ verwenden dabei ein und dasselbe vom Benutzer zu spezifizierende Kommunikationsprotokoll. Die S-Function-Blöcke können dann in Simulink

mit beliebigen anderen Modellblöcken verbunden werden. Damit die Simulation deterministisch bleibt, müssen externe Simulationskomponenten synchronisiert werden. Im einfachsten Fall geschieht das, indem beide Simulationssysteme nach dem Versand von Daten auf die Antwort des jeweils anderen Systems warten. Im Load-Step des VEROSIM®-Tasks "Extern" werden Daten verschickt und im Save-Step die Antwort erwartet und in die Datenbasis (VSD), hier konkret das E/A-Framework (VSDIO), geschrieben. Somit können Simulationssystem und externe Komponente ihre Berechnungen parallel im verbliebenen Teil des Simulationstaktes (Calc-Step) ausführen. Des Weiteren muss natürlich die Taktrate des Tasks „Extern" genau dem Ein-/Ausgabe-Takt des externen Modells entsprechen.

Diese Arbeiten sind in [RRJGK11] veröffentlicht worden.

AP4300: Implementierung projektspezifischer Erweiterungen des Simulationssystems

AP4310: Darstellung

Es wurde eine Softwarekomponente „SimpleVisualizationMetaphors" entwickelt, die Farbkodierung von Geometrien und vektorielle Darstellung von Zustandsgrößen der Simulation (vgl. Anforderungsdefinition in AP3310: Darstellung) in Form sog. Extensions anbieten kann. Eine Extension wird hierzu an ein bestehendes Element des Modells, z.B. ein Glied eines Beines, „angehängt" und erweitert es dadurch um neue Funktionen, in diesem Fall die Fähigkeit, einen Vektor in der 3D-Visualisierung im Ursprung des Koordinatensystems des betreffenden Gliedes zu zeichnen. Die Richtung wird dabei durch die z-Achse des jeweiligen Koordinatensystems vorgegeben, der Betrag kann von einem beliebigen Output gelesen werden.

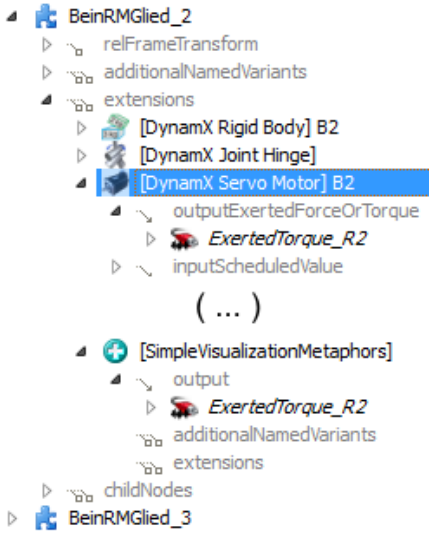


Abbildung 22: *Darstellung des Datenmodells zur Modellierung einer Komponente für eine einfache Visualisierungsmetapher in der grafischen Benutzeroberfläche von VEROSIM®*

Abbildung 22 zeigt den Ausschnitt der Datenbankrepräsentation aus einem Screenshot der grafischen Benutzeroberfläche von VEROSIM®. Hierin ist zu sehen, wie das Glied „BeinRMGlied_2" mit unterschiedlichen Extensions erweitert wird um die Funktionalitäten eines RigidBodys, eines Joints, eines Servo-Motors und der „SimpleVisualizationMetaphor". Letztere verweist auf den Ausgang „ExertedTorque_R2" des Motors, um dessen ausgeübtes Drehmoment zu visualisieren.

Abbildung 23 oben zeigt eine erste Demonstration des Tools. Alle Aktuatoren auf der linken Seite des Roboters werden durch die inverse Dynamik der Starrkörperdynamikkomponente realisiert, d.h. ihre Momente werden so bestimmt, dass die angetriebenen Gelenke exakt die vom Controller geforderten Soll-Winkel erreichen. Die

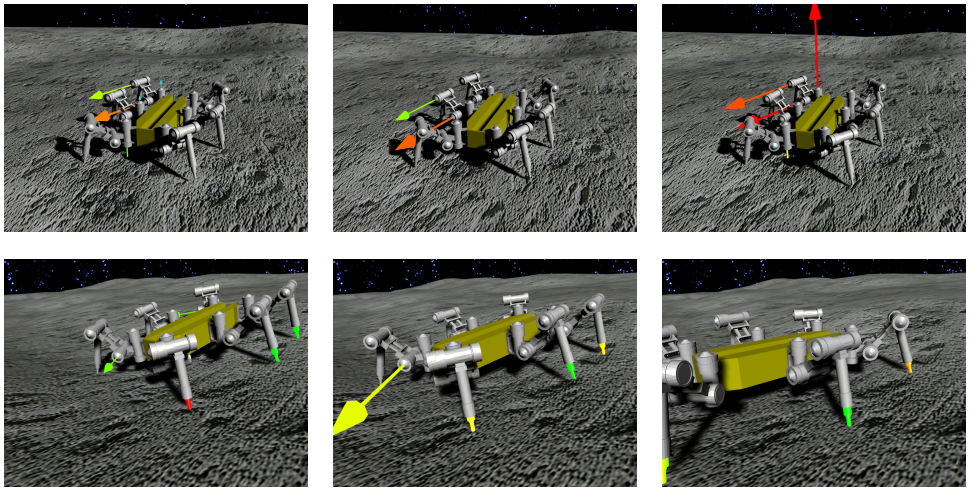


Abbildung 23: *Visualisierungsmetaphern für interne, physikalische Größen zur Simulationszeit in VEROSIM®: Motordrehmomente (vektoriell, oben) und Fußkontaktkräfte (Einfärben bestehender Geometrie, unten)*

entsprechenden Aktuator Modellkomponenten besitzen Ausgänge, an denen immer das gerade eingeprägte Moment anliegt. Und eben genau diese Ausgänge werden nun von den neuen Extensions als Beträge für die gezeichneten Momentvektoren herangezogen. Abbildung 23 unten zeigt eine kurze Sequenz der Scarabeus Simulation, in der Fußkontaktkräfte durch Verfärbungen (rot: hohe Belastung, grün: geringe Belastung) der entsprechenden Glieder visualisiert werden.

Des Weiteren wurde die Anforderung nach der Darstellung interner Größen auf sog. Billboards erfüllt. In Abbildung 24 sind entsprechend zwei der Gelenke mit solchen Billboards ausgerüstet. Auf den Billboards wird textuell die gegenwärtige Gelenkauslenkung angezeigt.

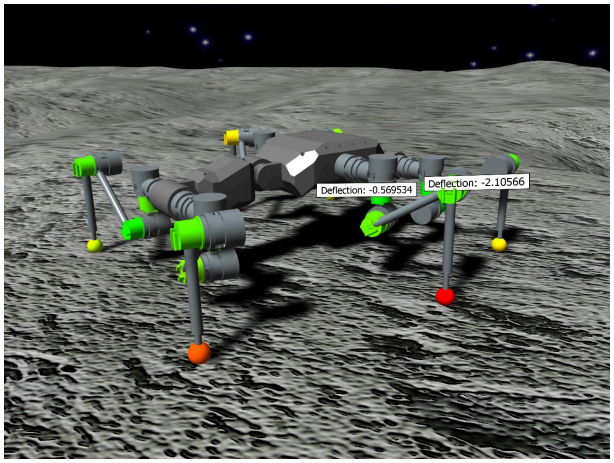


Abbildung 24: *Darstellung interner Größen auf Billboards. Hier: Gelenkauslenkung.*

Alternativ kann jede beliebige andere Größe, die über einen Ausgang bereitgestellt wird, angezeigt werden. Die Zuordnung zwischen Metapher und physikalischer Größe ist eine Frage der Modellierung und kann mit wenigen Mausklicks geändert werden. So werden z.B. in Abbildung 24 am Modell des SpaceClimbers jetzt neben den Fußkontaktkräften auch die Motordrehmomente durch Farbkodierung visualisiert.

E/A-Editor

Der graphische IO-Editor von VEROSIM® wurde verbessert. Damit lassen sich leichter Verbindungen zwi-

schen Untermodellen herstellen und verwalten. Des Weiteren lassen sich neben digitalen Komponenten (NOT, AND, OR, ...) für logische Netzwerke jetzt auch analoge Komponenten (ADD, SUB, MULT, ...) und Systemkomponenten (z.B. PID-Regler, Gleichstrommotor, siehe AP4120) über die Oberfläche im Modell erzeugen und verbinden (siehe Abb. 25). Auf diese Weise können über die Entwicklungsumgebung von VEROSIM® neue Systemkomponenten programmiert und leicht integriert werden.

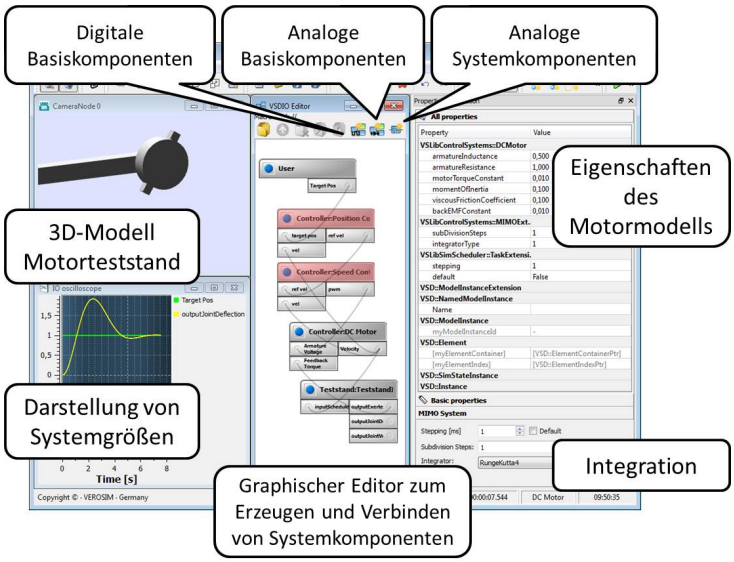


Abbildung 25: *Einbettung des graphischen IO-Editors (Bildmitte) in VEROSIM®*

Ghost-Modus

Zum Vergleich von zwei unterschiedlich parametrisierten Simulationsläufen wurde ein Ghost-Modus implementiert. Im Ghost-Modus wird eine semi-transparente Version des Roboters aus einem vorherigen Lauf eingeblendet, was den visuellen Vergleich zweier Läufe ermöglicht. Dabei können die zu protokollierenden Eigenschaften (z.B. Posen von Objekten) flexibel ausgewählt und in eine Log-Datei geschrieben werden. Diese kann dann auf einer halbtransparenten Ghost-Version des Roboters abgespielt werden (siehe Abb. 26).

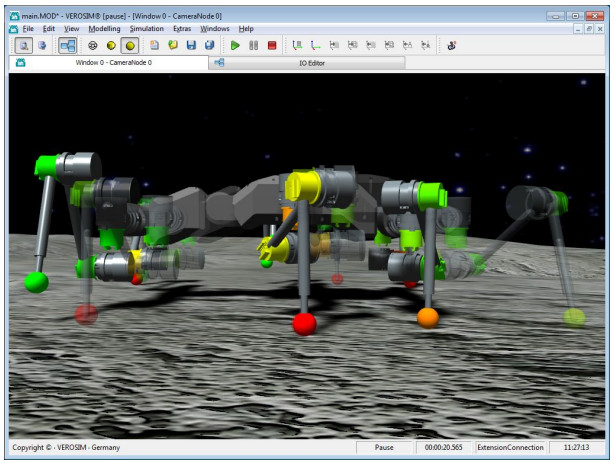


Abbildung 26: *Ghost-Modus zum Vergleich unterschiedlich parametrisierter Simulationsläufe*

AP4320: Dynamik

Das Referenzexperiment zur Kalibration des Motormodells beschreibt einen Motor, der über eine Seilwinde ein Gewicht bewegen kann. Da mit einer Starrkörperdynamik nur sehr unzulänglich Seile modelliert und simuliert werden können, wurde an dieser Stelle ein neuer Typ Zwangsbedingung realisiert, der das Verhalten der Seilwinde gut in die Starrkörperdynamik abbildet. Die Annäherung im virtuellen Experiment besteht aus einem Drehgelenk für den Motor und einem senkrecht angeordneten Lineargelenk für die Last. Um den Einfluss des Seiles nachzubilden, wurde ein neuer Differenzial-Zwangsbedingungstyp realisiert, der die Relativbewegung am Drehgelenk des Motors und im Lineargelenk der Last in ein Verhältnis setzt. Das Differenzial sorgt dafür, dass am Drehgelenk immer genau die selben Belastungen auftreten, als würde tatsächlich ein Gewicht über eine Seilwinde am Motor hängen. Je nach gewünschter Genauigkeit muss beim Aufbau des virtuellen Referenzexperiments noch dafür gesorgt werden, dass evtl. ein Seilgewicht, ein Windenträgheitsmoment und eine Lagerreibung in der Simulation berücksichtigt werden.

Realisierung der Interaktion zwischen Datenhandschuh und Mehrkörpersimulation

Im Projekt ist die Interaktion zwischen Anwender und Simulation über einen positions- und orientierungsgetrackten Datenhandschuh gefordert. Bereits in AP4110: Visualisierung / Bedienung / Interaktion wurde die Magnetmetapher beschrieben, die einen Körper der Simulation bewegt, indem sie die Ausgabewerte einer SpaceMouse als Sollwerte für die Geschwindigkeiten des Körpers interpretiert. Die Ausgabewerte des auch am DFKI eingesetzten Trackingsystems „InterSense“ sind jedoch nicht Geschwindigkeiten, sondern direkt Positionen und Orientierungen. Entsprechend war eine Ergänzung der Mehrkörpersimulation notwendig, um einen Körper der Simulation durch Nutzung von Zwangskräften an eine gewünschte Position im Raum bzw. in eine gewünschte Orientierung zu bewegen.

Auf Grundlage der Steuerung eines Starrkörpers durch direkte Vorgabe von Geschwindigkeiten ist eine solche Anforderung durch Einsatz eines Reglers einfach möglich. Regelgröße des Reglers ist die Abweichung zwischen Sollposition und Istposition bzw. zwischen Sollorientierung und Istorientierung des betroffenen Körpers.

Die Richtung der Geschwindigkeit zur Korrektur des Fehlers der Position entspricht genau der Richtung des Fehlers. Seien also $\vec{p}_{\text{soll}}$ der Sollwert der Position, $\vec{p}_{\text{ist}}$ ihr Istwert, dann ergibt sich die Richtung $\vec{v}, |\vec{v}| = 1$ der Korrekturgeschwindigkeit zu:

$$\vec{v} = \frac{\vec{p}_{\text{soll}} - \vec{p}_{\text{ist}}}{|\vec{p}_{\text{soll}} - \vec{p}_{\text{ist}}|}$$

Zur Bestimmung einer Winkelgeschwindigkeit zur Korrektur des Orientierungsfehlers wird wie folgt vorgegangen: Sei $\vec{q}_{\text{soll}}$ der Sollwert der Orientierung in Form eines Rotationsquaternions, $\vec{q}_{\text{ist}}$ das entsprechende Istwertquaternion. Gesucht ist eine Korrekturrotation $\vec{q}_{\text{korr}}$, die $\vec{q}_{\text{ist}}$ in $\vec{q}_{\text{soll}}$ überführt:

$$\begin{aligned} \vec{q}_{\text{korr}} \cdot \vec{q}_{\text{ist}} &= \vec{q}_{\text{soll}} \\ \vec{q}_{\text{korr}} &= \vec{q}_{\text{soll}} \cdot \vec{q}_{\text{ist}}^{-1} \end{aligned} \tag{1}$$

Eine Winkelgeschwindigkeit zur Korrektur des Fehlers ergibt sich durch Konvertierung dieses Rotationsquaternions in eine Drehvektor-Drehwinkel-Darstellung. Der Drehvektor dient dann als Richtung der Winkelgeschwindigkeit, der Drehwinkel als Betrag des Fehlers. Unter Einsatz eines P-Reglers können somit Geschwindigkeiten zur Korrektur des Positions- und des Rotationsfehlers bestimmt werden und in der „Magnetmetapher“ genutzt werden. Das Werkzeug ist in der Klasse `VSPluginRBDynamX::ExtensionRigidBodyDirectControl` implementiert. Als Sollwertvorgabe dient die Referenz „referenceNode“, die auf einen beliebigen VSD3D::Node innerhalb eines VEROSIM®-Modells verweist.

Das Ergebnis ist in Abbildung 16 dokumentiert.

SphereTrees zur automatisierten Kollisionshüllengenerierung

Um einen effizienten Workflow vom Inventor®-CAD-Modell hin zu einem VEROSIM®-Simulationsmodell zu ermöglichen, wird u.a. ein Verfahren benötigt, um aus einem detaillierten CAD-Modell eine Kollisionshülle zu erzeugen, die möglichst effizient für die Kollisionserkennung einzusetzen ist. Ein möglicher Weg könnten sog. SphereTrees sein. Die grundlegende Idee ist es, eine detaillierte Geometrie automatisiert durch eine Menge von Kugeln zu approximieren. Kugeln bieten sich an, weil sich an ihnen Kontaktparameter wie Kontaktpunktposition, Durchdringungstiefe und Kontaktnormale besonders leicht bestimmen lassen.

Ein SphereTree ist eine Baumstruktur, dessen Wurzel eine Kugel beschreibt, die die gesamte Geometrie umhüllt. Ihre Kinder sind kleinere Kugeln, die jede nur einen Teil der Geometrie umhüllen, in der Summe aber wiederum die gesamte Geometrie. Ein solcher SphereTree kann z.B. in Form eines Octree aufgebaut werden: Zunächst wird der kleinste achsenparallele Würfel bestimmt, der die gesamte Geometrie einhüllt. Die Wurzelkugel umhüllt diesen Würfel. Nun wird der Würfel in acht Teile zerlegt. Jeder dieser acht Teile, der noch Teile der Geometrie enthält, wird wieder mit einer Kugel umhüllt. Diese Kugeln bilden die Kinder der Wurzel. Der Prozess wird weiter wiederholt, bis die gewünschte Tiefe der Approximation erreicht ist.

Abbildung 27 zeigt die Sequenz der Approximation eines Beingliedes des SpaceClimbers der Tiefe 6 in VEROSIM®.

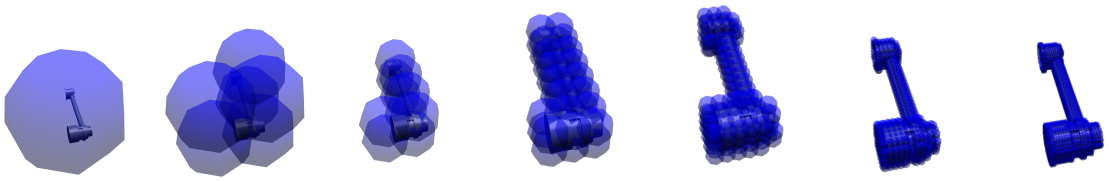


Abbildung 27: *Approximation eines Beingliedes des SpaceClimbers durch einen SphereTree der Tiefe 6 zur effizienten Kontaktpunktberechnung in der Dynamiksimulation*

Offenlegung einer Gluing-Schnittstelle

Das Gluing wurde bereits als ein mögliches Verfahren zur Kopplung von Bodenmechaniksimulationsverfahren an die VEROSIM®-Mehrkörpersimulation identifiziert. Eine mögliche Gluingschnittstelle wurde offengelegt. Zur Demonstration des Verfahrens wurde beispielhaft eine einfache Bodenmechaniksimulation implementiert und per Gluing an die Mehrkörpersimulation angebunden. Abbildung 28 zeigt einen entsprechenden Screenshot.

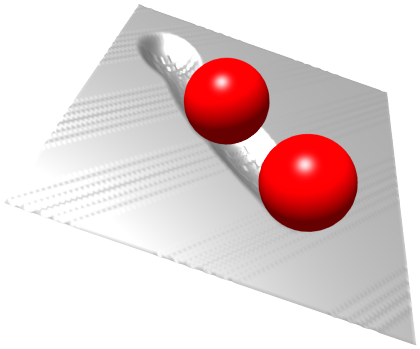


Abbildung 28: *Screenshot eines beispielhaft per Gluing angebundenem Bodenmechanikmodells*

Diese und einige weitere nützliche Optimierungen und Erweiterungen, die nicht im Rahmen dieses Projektes erarbeitet wurden, aber von denen Virtual Crater profitiert, sind in [JRKR11] veröffentlicht worden.

AP4330: Sensorik

Inertial Measurement Unit (IMU)

Unter Verwendung des I/O-Netzwerk-Konzepts von VEROSIM® wurde ein IMU-Modell implementiert. Dabei werden aus der Lage eines Geometrieknotens in den letzten drei Simulationsschritten Achsbeschleunigungen und Winkelgeschwindigkeiten in lokalen Koordinaten berechnet. Auf diese werden dann die Fehlermodelle aus den Abb. 29 und 30 aufgeschaltet. Diese Fehlermodelle sind frei parametrierbar und liefern zunächst ein plausibles Verhalten der IMU (siehe Abb. 31). Ein konkreter Abgleich mit einer realen IMU erfolgt in AP5400 Durchführung der Referenzexperimente.

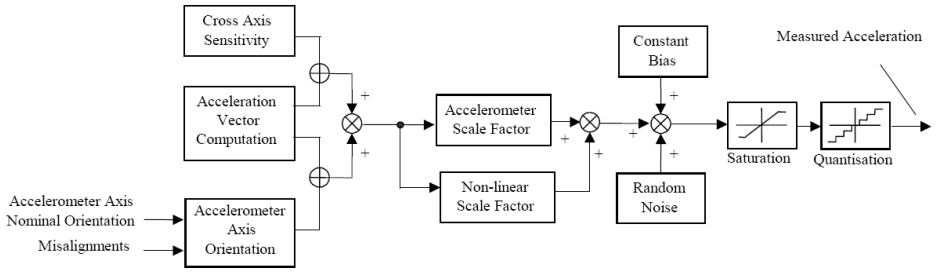


Abbildung 29: *Typisches Fehlermodell eines Beschleunigungssensors (entnommen aus [SMH+])*

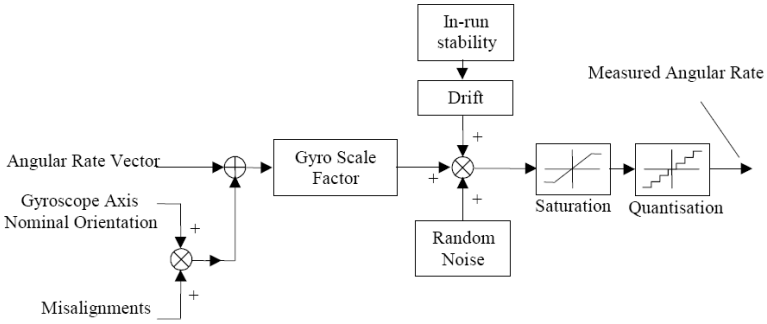


Abbildung 30: *Typisches Fehlermodell eines Gyroskops (entnommen aus [SMH+])*

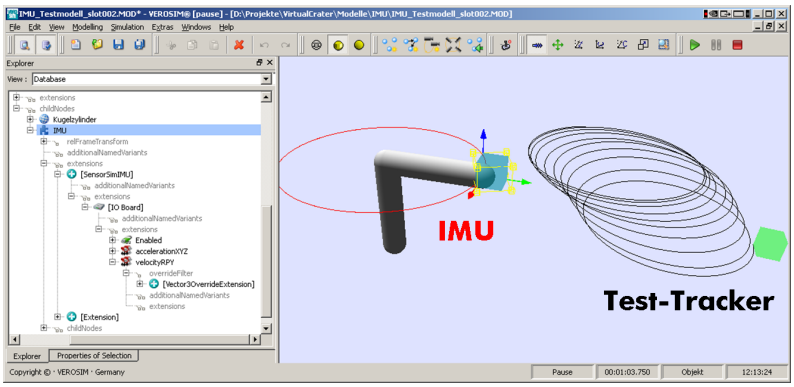


Abbildung 31: *Test einer simulierten IMU in VEROSIM®. Links bewegt sich die IMU auf der roten Kreisbahn, rechts bewegt sich ein Test-Tracker auf der aufgrund des Fehlermodells abdriftenden schwarzen Bahn.*

Kontakt- / Drucksensor

Ein wichtiger Sensor für die Simulation des Laufroboters ist ein Fußkontaktsensor, realisiert durch einen Drucksensor eingegossen in flexibles Material. Die Simulation eines solchen Sensors erfüllt die VEROSIM®-Extension „*ContactForce*“. Dieser Sensor erweitert einen Starrkörper der Simulation und liefert den Betrag aller auf den Körper einwirkenden Kontaktkräfte in einer dem Sensorframework entsprechenden Form.

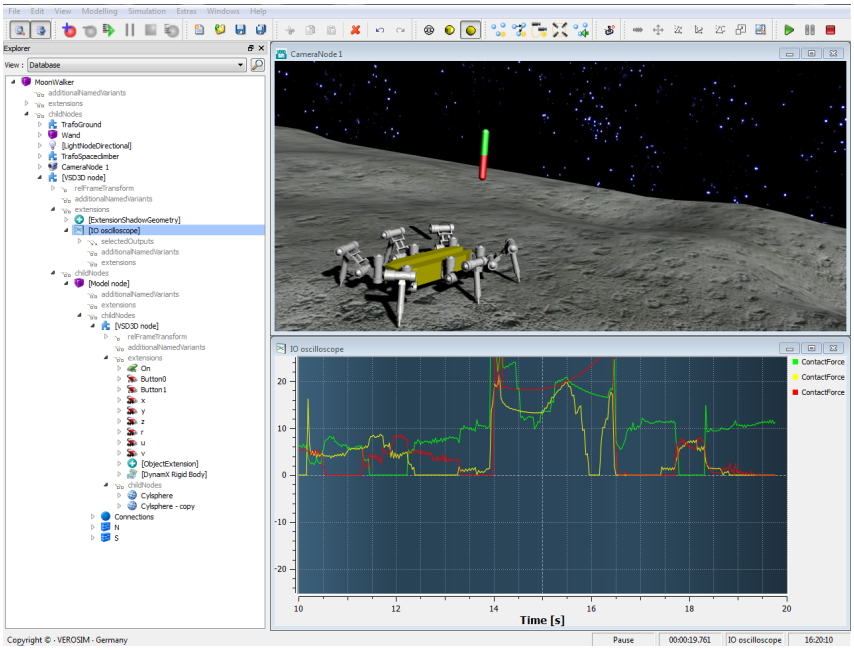


Abbildung 32: *Plot der Fußkontaktkraftsensoren während eines „Drückens“ mit Magneten auf den Roboter*

Als Parameter kann ein Richtungsvektor angegeben werden. Ist ein solcher Vektor definiert, werden nur Kontaktkräfte anteilig in Richtung des angegebenen Vektors berücksichtigt. Abbildung 32 zeigt den Screenshot eines „Experiments“ mit den Fußkontaktsensoren. Der Plot im unteren Bereich zeigt die Ausgabewerte der Fußkontaktkraftsensoren an den Beinen der linken Seite. Während der Simulation hat der Anwender mit dem Magneten auf den Roboter „gedrückt“ - am Plot ist schön zu erkennen, wie sich dies auf die Werte der Fußkontaktsensoren auswirkt (etwa bei Sekunde 14-16,5).

6DOF-Kraftsensor

Der in den Anforderungen beschriebene Kraftsensor für sechs Freiheitsgrade wurde auf Grundlage der Mehrkörpersimulation und unter Ausnutzung des VEROSIM®-Sensorframeworks implementiert. Der Sensor (VSPluginSensorSimRBDynamX::ForceTorque6DOF) wird in Form einer Extension an eine bestehende starre Verbindung (`VSPluginRBDynamX::ExtensionJointRigid`) des Mehrkörpersystems „gehängt“. Hier macht er nichts anderes, als die von der inversen Dynamik bestimmten Zwangskräfte und -momente, die zur Aufrechterhaltung der starren Verbindung notwendig sind, auszugeben. Damit dies sinnvoll funktionieren kann ist es notwendig, die Eigenschaft „forbidUnification“ des „Gelenks“ auf „true“ zu setzen, so dass die verbundenen Körper trotz der starren Verbindung von VEROSIM® als getrennte Instanzen simuliert werden. Andernfalls sorgt ein „Dynamische Rekonfiguration“-genannter Algorithmus in VEROSIM® dafür, dass starr verbundene Körper aus Sicht der Dynamiksimulation logisch vereint werden. Die Werte des Sensors stehen, dem Sensorframework entsprechend, auf Outputs zur Verfügung und können so z.B. geplottet werden oder als Eingabewerte für einen Controller dienen.

AP4340: Antriebe

Die Echtzeitfähigkeit der Starrkörperdynamiksimulation beruht unter anderem darauf, dass die meisten Berechnungen und die Formulierung der Zwangsbedingungen auf Ebene der Geschwindigkeit stattfinden. Dies erlaubt Zeitschrittweiten im Bereich von 10ms in der Integration. Kräfte und Momente extern in einem Motormodell zu berechnen und auf die Dynamiksimulation wirken zu lassen ist zwar möglich, bringt aber aufgrund der notwendigen 2-fachen Integration größere numerische Instabilität mit sich. Diese lässt sich nur durch Verringerung der Zeitschrittweiten auf etwa 1ms unter Verlust der Echtzeitfähigkeit kompensieren (vgl. AP3340: Simulation der Antriebe).

Bond-Graphen sind eine Möglichkeit zur domänenunabhängigen Beschreibung des dynamischen Verhaltens eines physikalischen Systems. Hierbei werden zwischen zwei Komponenten immer zwei physikalische Größen (allgemein *effort* und *flow* genannt) ausgetauscht, deren Produkt Leistung ist. Diese Leistungskopplung wurde bereits in [Yoo07] angewandt, um ein Eingabegerät mit haptischer Rückkopplung an ein Simulationssystem anzubinden. Auch hier spielte die Überbrückung verschiedener Zeitskalen und die Echtzeitfähigkeit eine wichtige Rolle.

Für den konkreten Fall der Integration von Motormodell und Starrkörperdynamiksimulation bietet die Leistungskopplung den entscheidenden Vorteil, dass als Stellgröße die Geschwindigkeit verwendet werden kann und die Dynamiksimulation das zur Einhaltung dieser Zwangsbedingung notwendige Moment als Lastmoment zurück koppelt (siehe Abb. 33, Leistung = Kraft x Geschwindigkeit).

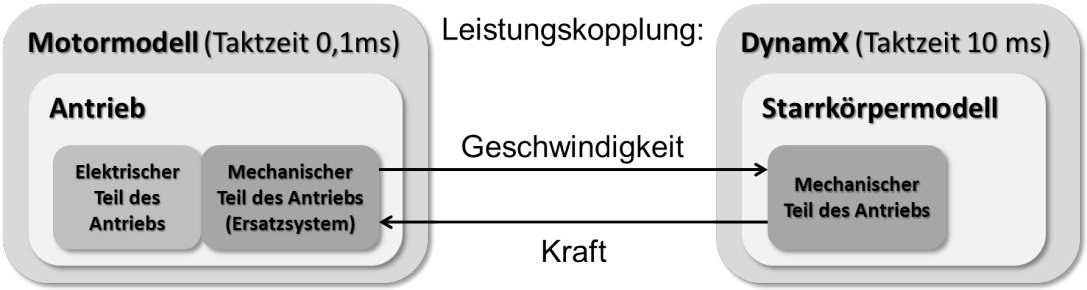


Abbildung 33: *Leistungskopplung von Motormodell und Starrkörperdynamiksimulation*

AP4350: Bodenmechanik

Es wurde eine auf die Modellierung der Bein-Boden-Wechselwirkung optimierte Geometrieklasse in VEROSIM® implementiert, welche die Höhendaten des Bodens in einem zweidimensionalen Raster von äquidistanten Punkten speichert und effizienten Zugriff für Geometrieberechnungen und -Manipulationen zur Laufzeit ermöglicht (siehe Abb. 34). Zur Visualisierung des Bodens wird dieses „Heightfield“ trianguliert. Für die Referenzexperimente kann das Modell der realen Testumgebung am DFKI auch direkt als Heightmap in VEROSIM® eingelesen werden.

Ansatz basierend auf zellulären Automaten

Der auf zellulären Automaten basierende Ansatz zur Simulation von Bodenmechanik wurde in folgenden Punkten weiterentwickelt:

- Die Kontaktgeometrie wird exakt eingehalten.
- Statische Kontaktsituationen wurden auf mechanischer Ebene stabilisiert, indem mehr Kontakte als echte Zwangsbedingungen in das zu lösenden Differentialgleichungssystem eingeführt werden.
- Die Kontaktreibung wird nicht mehr als externe Kraft aufgebracht, sondern genau wie die Normalenkraft als Zwangsbedingung. Dadurch wird das Kontaktverhalten numerisch stabiler.

Damit ist es jetzt möglich, das Referenzexperiment mit dem Schunk-Arm mit diesem Ansatz durchzuführen und das Verhalten qualitativ nachzubilden (siehe Abb. 35).

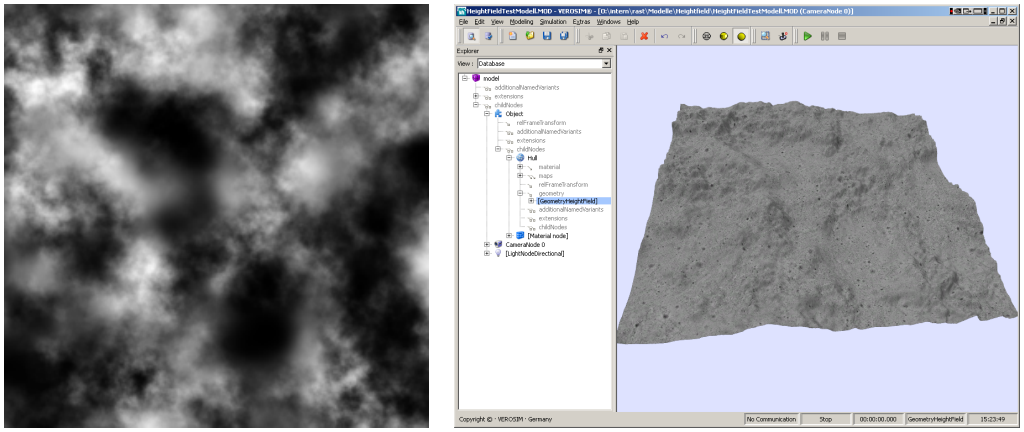


Abbildung 34: *Links: Höhendaten in Form einer Heightmap (Graustufenbild)*
Rechts: Darstellung der Höhendaten in VEROSIM®

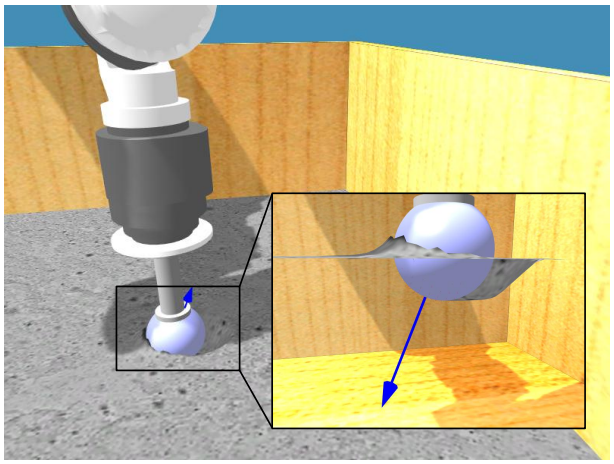


Abbildung 35: *Virtuelles Referenzexperiment mit dem Schunk-Arm.*

Auch die ganzheitliche Simulation des SpaceClimbers auf diesem Bodenmechanikmodell ist möglich. Abb. 36 zeigt dieses virtuelle Referenzexperiment mit mehreren Ansichten der Szene, Visualisierungsmetaphern zur Darstellung der gesampleten Kontaktkräfte (rote Pfeile), der kombinierten Kontaktkräfte, die an die Starrkörperdynamik übergeben werden (blaue Pfeile). Im unteren Bereich sind Analysewerkzeuge zu sehen, z.B. zur Anzeige der Zustandsgrößen eines Motors (Mitte unten) oder zur Darstellung aller Gelenkwinkel zur Überprüfung des Laufmusters.

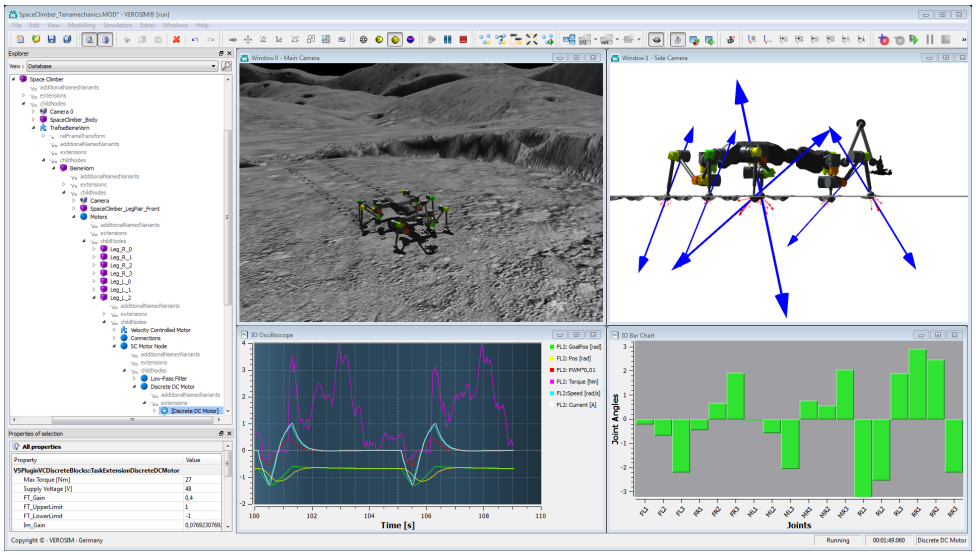


Abbildung 36: *Ganzheitliches Virtuelles Referenzexperiment mit komplettem SpaceClimber.*

AP4360: DFKI-CAD-Modellimport

Der Datentransfer zwischen Autodesk Inventor®und VEROSIM® wird mittels eines sog. Inventor®-AddIns realisiert. Dieses AddIn wird VEROSIM® als Bibliothek zu Autodesk®Inventor®„hinzuladen“ und Funktionen für den Datentransport bereitstellen.

Abbildung 37: *Ribbon des VEROSIM®-Inventor®-AddIns*

Abbildung 37 zeigt das Ribbon, das das Inventor®-VEROSIM®-AddIn in Autodesk®Inventor®zur Verfügung stellt. Derzeit unterstützt das AddIn folgende vier Grundfunktionen:

1. **Export nach VEROSIM®** Bei Aufruf dieser Funktion überträgt das AddIn die in Inventor modellierte Geometrie nach VEROSIM®. Dabei wird für jeden Oberflächenabschnitt (Inventor:Face) ein VSD3D::HullNode mit entsprechenden Facetten erzeugt. Diese werden in einen VSD3D::Node eingefügt, der das Inventor Part repräsentiert. Optional können auch die Materialien (in Inventor RenderStyles genannt) einschließlich Texturen übertragen werden.

   Falls der Export bereits aufgerufen wurde, hat der Benutzer die Wahl, ob die bereits vorhandenen Elemente der VEROSIM®-Arbeitszelle vor dem Export gelöscht werden sollen („clean export“) oder nicht („append“).

Beim Export wird die aktuelle Ansicht in Inventor auf die VEROSIM®-Kamera übertragen.

2. **Simulation (kontinuierlich oder Einzelschritt)** Überträgt dauerhaft bzw. einmalig die Positions- und Orientierungsinformationen der VEROSIM®-Nodes zu den dazu korrespondierenden Parts / Assemblies in Inventor®.

3. **Optionen** Öffnet einen Optionendialog zur Definition einiger Exporteinstellungen, wie z.B. Skalierungsfaktoren, Materialexportoptionen etc.

4. **Analyse der Assembly-Constraints** Mit Assembly-Constraints werden in Inventor Beziehungen zwischen zwei Komponenten (Parts oder Subassemblies) hergestellt, die die Bewegungsfreiheit der beteiligten Komponenten einschränken.

   Beim Export nach VEROSIM® analysiert das Addin diese Constraints und versucht, darin kinematische Ketten zu erkennen. Die Komponenten werden so angeordnet, dass Nachfolger in der kinematischen Kette ihrem Vorgänger als Kindknoten untergeordnet werden.

   Zusätzlich wird beim Export eine lesbare Beschreibung aller vorhandenen Assembly-Constraints erstellt.

Abbildung 38 zeigt den Screenshot eines Modells in VEROSIM®, das mittels des AddIns aus Autodesk®Inventor®nach VEROSIM® übertragen wurde.

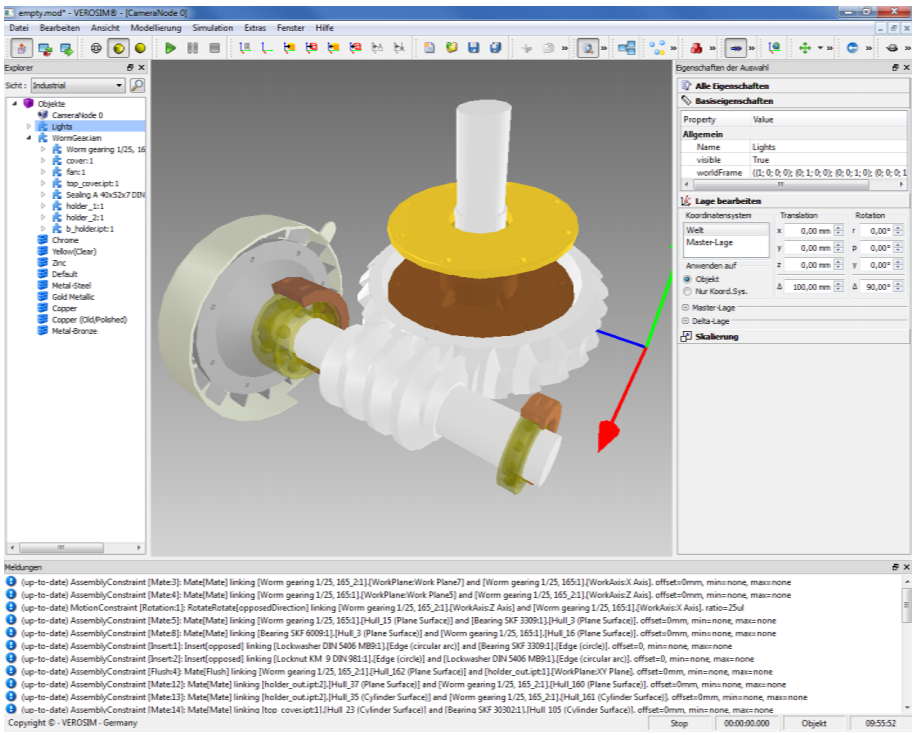


Abbildung 38: *Screenshot eines Autodesk®-Modells in VEROSIM® nach dem Exportvorgang*

AP4400 Modellierung und Aufbau der Referenzexperimente in virtuellem und realem Testbed

AP4420: Modellierung der Referenzexperimente im virtuellen Testbed

Referenzexperimente zum Motormodell

Der in AP3400 entworfene Motorteststand wurde schematisch in VEROSIM® nachmodelliert (siehe Abb.

39). Das zu testende Motormodell treibt die Winde an, die am oberen horizontalen Balken befestigt ist. Diese hebt dann die Last an, die entlang des vertikalen Balken läuft. Die Messgrößen wie die Geschwindigkeit und der Stromverbrauch des Motors lassen sich direkt in VEROSIM® visualisieren oder in eine Log-Datei schreiben. Mit dem vom DFKI entwickelten Automatisierungs-Plugin *VSPluginVCAutomatedExperiment* kann das Eingangssignal des Motors und die Masse der Last automatisch verändert werden. Dies ermöglicht die effiziente Durchführung dieses virtuellen Experiments in AP5420.

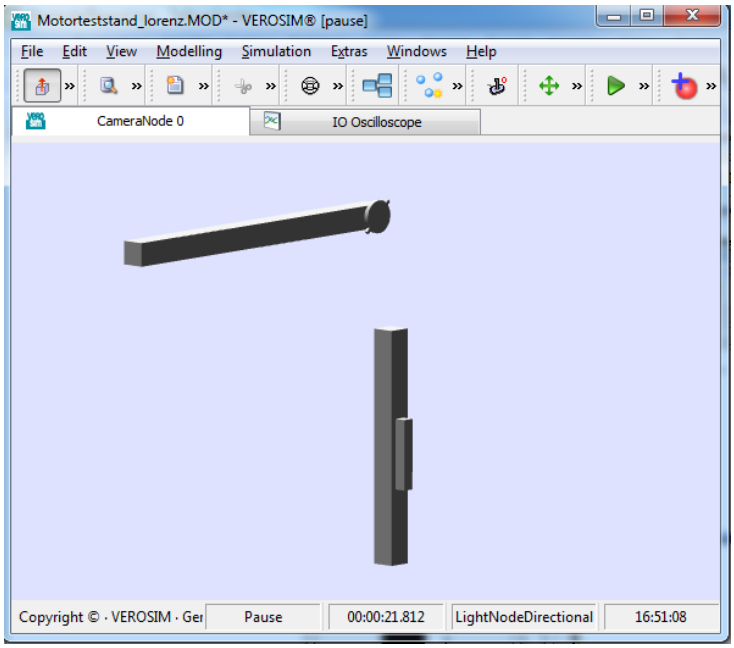


Abbildung 39: *Der in AP3400 definierte Motorteststand als schematisches Modell in VEROSIM®.*

Aufbau Bodenmechanikexperiment

Der Schunk-Arm aus dem Bodenmechanik-Experiment steht in der Modellbibliothek von VEROSIM® zur Verfügung und kann in beliebige Modellumgebungen importiert werden. Auf dieser Grundlage wurden diverse virtuelle Referenzexperimente mit dem Schunk-Arm im Zusammenspiel mit Varianten zur Bodenmechaniksimulation durchgeführt, vgl. Abb. 35 in AP4350: Bodenmechanik.

Aufbau IMU-Experiment

Das Experiment zur Untersuchung der simulierten IMU besteht aus einer IMU, die an der Spitze eines Manipulators montiert eine vorgegebene Trajektorie abfährt und deren Messwerte protokolliert werden. Als potenzielle Manipulatoren waren der Mitsubishi PA10 und der Schunk-Arm vorgesehen. Der simulierte Beschleunigungssensor (Bibliothek `VSPluginSensorSimIMU`), Mechanismen zur Fehlersimulation (`VSDIO::ElementNodes` in E/A-Netzwerken), zur Protokollierung (Bibliothek `VSPluginDataLoggerVSDIO`), der Schunk-Arm selbst sowie eine Schnittelle zum Einlesen von Trajektoriendaten aus Logfiles (Bibliothek `VSPluginVCMLController`) stehen in VEROSIM® zur Verfügung. Mit dieser Konfiguration konnten die Experimente durchgeführt werden.

Laserscanner / PMD Kamera Experiment

Die Modellierung erfolgte hier auf Grundlage simulierter Laserscanner (`VSPluginSensorSimLaserScanner`), einer simulierten PMD-Kamera (`VSPluginSensorSimPMDSensor`) sowie Mechanismen zur Berücksichtigung von Fehlermodellen.

AP4500: Aufbau einer Simulations-/Projektionsumgebung im DFKI

Im Rahmen dieses Arbeitspaketes wurde als Simulationsumgebung eine Mehrfach-Stereo-Projektion konzeptioniert. Diese wurde Ende 2010 / Anfang 2011 in den Räumen des DFKI als 5-fach-Projektion - mit der Möglichkeit der Erweiterung zur 7-fach-Projektion - realisiert. Die installierte Hard- und Softwareumgebung ist die Basis für die weiteren Entwicklungen im Rahmen des Projekts.

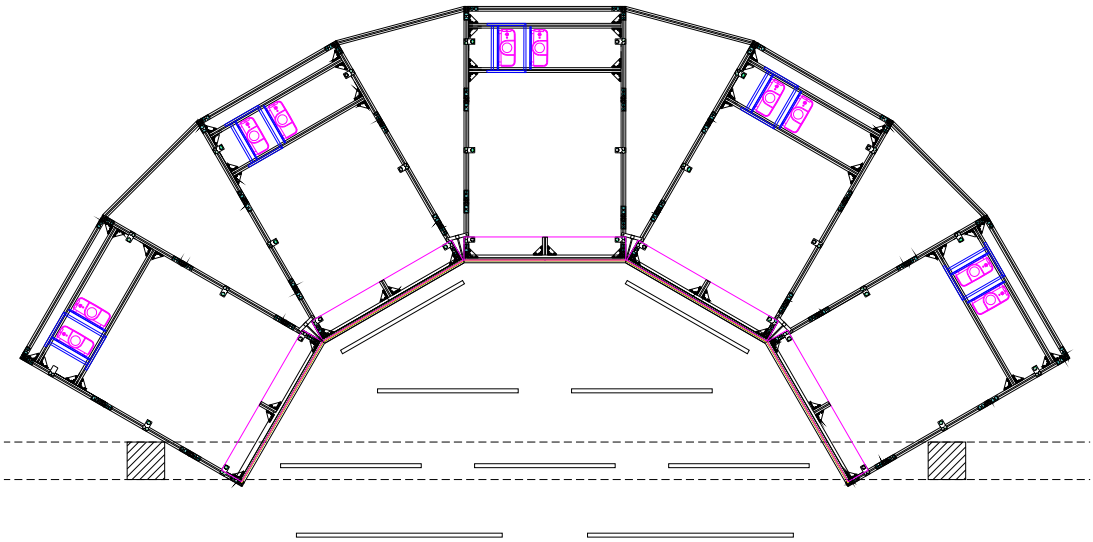


Abbildung 40: *CAD-Zeichnung der installierten 5 Projektionsboxen*

Mechanischer Aufbau des Projektionssystems und Infrastruktur

Die Simulationsumgebung wurde beim DFKI im Gebäude „Unicom“ in einem speziell umgebauten Raum mit einer (nutzbaren) Größe von etwa (B x T) 11,50 x 9,30 m aufgebaut. Sie besteht aus dem Projektionssystem, den Simulationsrechnern mit der Simulationssoftware, der Peripherie, etwa zur Benutzerinteraktion, und einem Bedienstand. Das Projektionssystem besteht aus 5 modularen Projektionsboxen (siehe Abb. 40). Eine Projektionsbox besteht aus Aufbaugestell, Projektoren, Projektorhalterungen, Umlenkspiegeln und der Projektionsscheibe (siehe Abb. 41). Die einzelnen Boxen stehen an Vorderseiten mit den Projektionsscheiben bündig zur nächsten Box, jeweils um 30° versetzt. Die modulare Bauweise bietet die Möglichkeit für eine leichte Erweiterbarkeit. Mit der Erweiterung auf 7 Seiten bilden die Projektionsscheiben einen Halbkreis.

Neben der Raumgröße war die Raumhöhe für die Dimensionierung und Anordnung der Projektionsboxen maßgeblich. Die Raumhöhe beträgt 3,06 m. In der Mitte des Raumes verläuft quer zur Anlage eine Unterzug (0,40 m), der auf 2 Säulen aufliegt. Der Unterzug ist außerdem in die Planung des Trackingsystems eingeflossen. Die Scheiben sind im Hochformat montiert, die Größe beträgt etwa (B x H) 1,52 x 2,03 m (100”). Der Arbeitsbereich für die Ingenieure im Halbrund vor den Projektionsleinwänden beträgt etwa 5,70 x 3,60 m. Dieser Bereich ist für die Arbeit eines einzelnen oder das kollaborative Arbeiten mehrerer Personen ausreichend groß. Durch den Freiraum vor der Mehrfachprojektion können auch für kleinere bis mittlere Besuchergruppen Vorführungen erfolgen (siehe Abb. 42).

Die Mehrfachprojektion mit ihren Projektionsboxen und Rechnerschränken nimmt den hinteren Teil des Raumes ein. Im vorderen Teil des Raumes bleibt, neben dem Freiraum für die Besuchergruppen, Platz für die Erweiterungsboxen und den Bedienstand. Die Mehrfachprojektion wird im Stereo-Modus betrieben. Pro Projektionsbox werden hierzu zwei 2 Videoprojektoren verwendet. Ein Projektor stellt das Bild für das linke, der andere für das rechte Auge dar. Polfilter vor den Projektoren und eine Polfilterbrille für den Betrachter ermöglichen den 3-D-Eindruck. Durch dieses passive System mit polarisierten Filtern und Brillen ist ein flimmerfreies und weitgehend ermüdungsfreies Arbeiten möglich. Die Brillen sind leicht und preisgünstig. Im Vergleich zu aktiven Stereo-Systemen ist für die Brillen kein Emitter mit einem zusätzlichen Schaltsignal notwendig. Ein solcher

Abbildung 41: *Blick in die Projektionsboxen*

Emitter müsste ständig im Sichtbereich des Betrachters liegen. Würde dieser verdeckt, käme es zu einem Verlust des 3-D-Eindrucks. Das verwendete passive System bietet somit insbesondere auch bei größeren Gruppen Vorteile. Die Projektoren der Anlage werden mit einer Auflösung von 1.400 x 1.050 Pixeln betrieben. Durch den großen Arbeitsbereich, die Scheibengröße, das weit abgedeckte Gesichtsfeld, die Stereodarstellung und die hohe Auflösung ist eine bestmögliche Immersion, also das Eintauchen in die virtuelle Szene, möglich. Vom Bedienstand aus können die gesamte Anlage und einzelne Modelle gesteuert werden. Auch das Update der Simulationssoftware und die Modellpflege können von hier erfolgen. Über einen mobilen Tablet-PC kann die Steuerung zusätzlich aus dem Arbeitsbereich heraus übernommen werden. Der Raum verfügt über eine Verdunkelungseinrichtung der Fenster. Eine Verkleidung der Projektionsboxen verhindert, dass Streulicht in den Arbeitsbereich des Ingenieurs dringt. Eine Klimaanlage kühlt die Abwärme von Projektoren und Rechnern und regelt die Temperatur im Arbeitsbereich. Über Bodentanks werden die Projektoren, Rechnerschränke und die Peripherie mit Strom versorgt. Auch die Netzwerk-Verbindung und die Bildübertragung zum Bedienstand erfolgt über die Bodentanks. Zur Fernwartung, z. B. durch die Projektpartner, wurde eine Möglichkeit zur Übertragung von Bilddaten, Maus- und Tastatureingaben über das Netzwerk mittels besonderer KVM-Switche geschaffen. Für die Benutzerinteraktion wurde ein Trackingsystem von Intersense installiert. Das System liefert mittels Ultraschall die Position von Eingabemedien im Raum. Mögliche Eingabegeräte sind ein Zeiger (Wand) oder die vorhandenen Datenhandschuhe des DFKI. Auch die Kopfposition des Benutzers ist so bestimmbar. Für eine einfache Steuerung der virtuellen Szene wurde eine 3D-Maus installiert. Durch den modularen Aufbau des Projektionssystems und die Erweiterbarkeit der Simulationssoftware können - beinahe beliebig - weitere Interaktionsmedien eingebunden werden. Für die akustischen Ausgaben wurde ein 5.1 Tonsystem installiert. Weitere Peripherie lässt sich durch die verwendeten Aluminium-System-Profile des Aufbaugestells leicht montieren. Die zum Simulationssystem gehörenden 16 Rechner und Server sind in 2 Serverschränken im hinteren Bereich der Anlage verbaut (siehe Abb. 43). In den Serverschränken befinden sich außerdem das Steuerungsmodul der Intersense, die Netzwerkswitche, die KVM-Switche, (schaltbare) Steckdosenleisten und der Access-Point für den Tablet-PC. Die Verwendung von Standardkomponenten ermöglicht auch hier die leichte Austauschbarkeit und Erweiterbarkeit.

Rechnerkonfiguration

Das Simulationssystem VEROSIM wird parallel auf mehreren Rechnern betrieben. Pro Leinwand und Auge berechnet ein Client-Rechner die virtuelle Szene. Die Simulation der Objekte in der Szene, die Interaktion mit den Eingabemedien, und die Berechnung der Motorsteuerung werden von Master-Rechnern übernommen.

Abbildung 42: *Arbeitsbereich der fertig montierten und verkleideten Stereo-Mehrfachprojektion*

Die Verteilung der errechneten Größen und resultierenden Ansichten ist inhärenter Teil des Simulationssystems. Ein Server hält die Anwendungs- und Modelldaten und stellt sie den Client- und Masterrechnern zur Verfügung. Auf diesem Server läuft auch das Steuerungsprogramm der Projektionsanlage, mit dem sich die Anlage mitsamt den Projektoren starten und ausschalten lässt oder mittels derer die verschiedenen Modelle geladen und ausgeführt werden können. 2 Reserverechner stehen für den kurzfristigen Austausch bereit. Für das Backup wurde ein separater Server installiert. Dieser stellt ein Versionsverwaltungssystem bereit, in dem die verschiedenen Modellstände gespeichert werden können. Damit wurde beim DFKI eine ähnliche Infrastruktur geschaffen, wie bei den Projektpartnern RIF und MMI. Dies begünstigte den Austausch und die Weiterentwicklung der verschiedenen Modelle und Applikationen. Durch die Skalierbarkeit von VEROSIM können Modelle sowohl in der Stereo-Mehrfachprojektion betrieben werden als auch auf einem einzelnen Rechner. Dazu ist in der Regel keine oder nur eine minimale Modellpflege notwendig. Somit können Mitarbeitende im Projekt ihre Änderungen am Simulationsmodell, in den Programmbibliotheken oder den externen Programmen erst am lokalen Arbeitsplatzrechner verifizieren, bevor ein umfangreicher Test in der Mehrfachprojektion stattfindet.

Inbetriebnahme

Das Gesamtsystem wurde Anfang 2011 in Betrieb genommen. Nach dem mechanischen Aufbau wurde das Projektionssystem eingerichtet. Die Projektoren wurden mittels der Verstellmöglichkeiten des Gestells grob justiert. Anschließend wurde eine Kalibrierung mittels der Steuerungssoftware vorgenommen. Diese Kallibierung muss regelmäßig wiederholt werden. Parallel dazu erfolgten die Einrichtung der Rechner und die Anpassung des Simulationssystems. Die Schnittstelle zum DFKI-Intranet wurde definiert und umgesetzt. Die Peripherie wurde eingebunden; die Feineinstellung des Trackingsystems selbst wurde vom DFKI vorgenommen. Das Datenverzeichnis auf dem Server und die einzelnen Simulationsrechner wurden auf den Backup-Server gesichert. Bereits während der Installation und insbesondere in den Wochen danach fand die Testphase statt. Dabei wurden diverse Hardwarefehler erkannt und behoben. Während der Testphase aber auch im gesamten späteren Verlauf des Projekts wurden neuere Versionen der Simulations- und Steuerungssoftware eingespielt. Im Zuge der Testphase wurden die Mitarbeiter in Bedienung des Systems geschult. Im Anschluss wurde eine Dokumentation und Bedienungsanleitung erstellt und übergeben.

Abbildung 43: *Serverschrank, Vor- und Rückansicht*

AP4600: Einarbeitung in das und Schulung auf dem Simulationssystem

Im November 2009 wurde am MMI in Aachen eine zweitägige Einführung in das Simulationssystem VEROSIM® durchgeführt. Teilgenommen haben zwei Mitarbeiter vom RIC Bremen (Michael Rohn und Yong-Ho Yoo). Inhalt der Einführung war 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, alle notwendigen Daten für eine C++-Entwicklungsumgebung sowie einige Beispiel-Plugin-Ins erhalten, so dass nun auch im DFKI an der Umsetzung in VEROSIM® gearbeitet werden kann.

Parallel zum im Oktober 2010 begonnenen Aufbau und während der Einrichtung der Projektionsumgebung bzw. während der Nacharbeiten wurde den Mitarbeitern des DFKI der Umgang der CAVE näher gebracht und Einzelheiten erklärt. Mitarbeiter des DFKI wurden sehr eng bei dem Aufbau und der Einrichtung der Umgebung eingebunden, so dass eine gesonderter Schulungstermin nicht mehr nötig war. Zur Einarbeitung wurden mehrere Modelle in die Umgebung gebracht und diese bearbeitet. Hiermit wurde dem Personal am DFKI ein Einblick in die Arbeit mit und an der Projektionsumgebung gegeben.

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 bereits intensiv genutzt und hat die Effizienz der Kommunikation deutlich gesteigert. Über den Sharepoint Server bestanden neben der reinen Forums-Funktion weitere Möglichkeiten der Zusammenarbeit für die Zukunft, wie z.B. das gemeinsame Bearbeiten von Dokumenten.

Modellbibliothek

Um das Arbeiten mit VEROSIM® für Einsteiger im Rahmen von VirtualCrater und anschließend zu erleichtern, wurde eine Modellbibliothek erstellt, in der die Implementierungen an VEROSIM® der Pakete AP4100 - AP4330 mit Beispielmodellen und kurzer Dokumentation enthalten sind (siehe Abb. 44). Außerdem wurde ein internes Dokument aufgesetzt, in dem die Implementierungen entsprechend dem Anforderungsdokument aufgelistet und beschrieben sind und auf die dazugehörigen Modelle in der Modellbibliothek verwiesen ist. Diese Modellbibliothek kann jederzeit vom Benutzer um eigene Modelle erweitert werden.

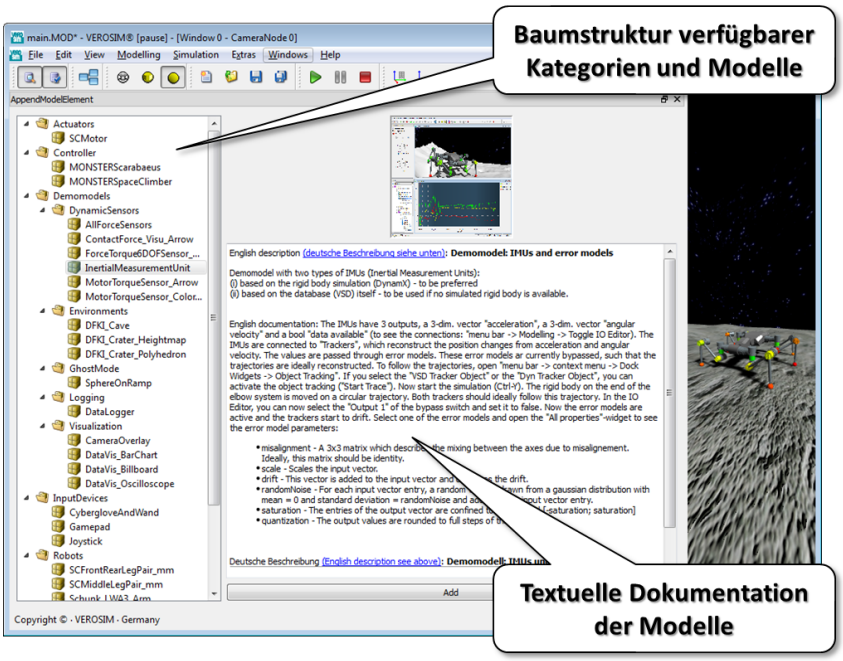


Abbildung 44: *Modellbibliothek für VitualCrater*

AP5100: Systemtest und Anpassungen der Benutzerschnittstellen des Simulationssystems

Im Zuge der Erstellung des Prototypen für den Milestone I, der am 23.04.2011 am DFKI vorgestellt wurde, ist das gesamte System einem Praxistest unterzogen worden. Daraufhin erfolgte aufgrund von Problemen bei der Stabilität der Verbindung eine Überarbeitung der Anbindung des Datenhandschuhs.

Der Test der Entwicklungsumgebung erfolgt u.a. durch die intensive Arbeit der Mitarbeiter am DFKI an eigenen Simulationskomponenten für VEROSIM®. Werden hier Fehler oder Verbesserungsmöglichkeiten entdeckt, werden sie auf einer gemeinsamen Kommunikationsplattform („Sharepoint"-Server des MMI) dokumentiert und diskutiert. Eine Vielzahl kleinerer Probleme und Verbesserungen konnte auf diesem Wege identifiziert und durch Mitarbeiter von RIF oder dem MMI beseitigt bzw. umgesetzt werden.

AP5200: Systemtest und Anpassungen der Schnittstellen zwischen Steuerungshardware und Simulationssystem

Das Steuerungs-Interface HOBBIT hat eine TCP-Schnittstelle erhalten, welche High-Level-Befehle der Form „*desired_turn_amplitude 0.15*" entgegennimmt und an MONSTER zur Generierung der Laufmuster weiterleitet. In VEROSIM® wurde eine Erweiterung der bisherigen TCP-Schnittstelle dahingehend vorgenommen, dass zusätzlich zur vorhandenen periodischen Kommunikation mit MONSTER über Datenpakete auch einzelne Befehle gesendet werden können. Zum Test dieser Anbindung wurde im Rahmen der Erstellung des Milestone-I-Prototypen ein Pfadplaner implementiert, der den SpaceClimber zu einer beliebigen, veränderlichen Position im Modell hinlaufen lässt. Dabei wurden zunächst keinerlei Sensoren verwendet, sondern lediglich direkt auf der Geometrie des Modells gearbeitet. Abb. 45 zeigt, wie in VEROSIM® verschiedene Befehle formatiert und mit einem TCP-Port assoziiert werden können. Die Argumente bzw. Werte der Befehle können dabei von anderen Komponenten entgegengenommen werden, wobei diese Verbindungen komfortabel im IO-Editor hergestellt werden können.

Abb. 46 zeigt beispielhaft die Kommunikationsinfrasztruktur zwischen VEROSIM® und den High-Level-

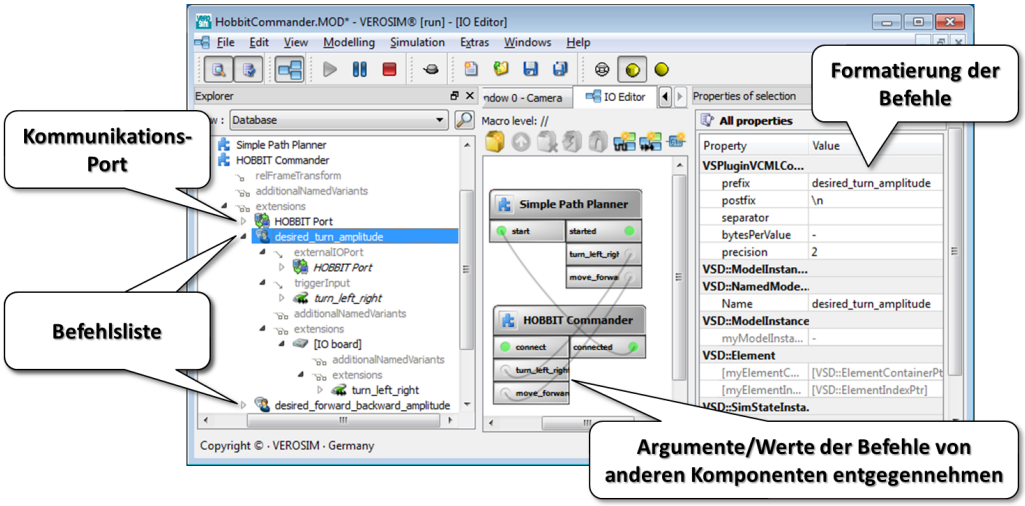


Abbildung 45: *Konfigurieren und Senden von Befehlen an HOBBIT*

Controllern HOBBIT bzw. dem Pattern-Generator MONSTER des DFKI. Der Pfadplaner schickt ereignisbasiert Befehle z.B. zu Laufgeschwindigkeit oder Anstellwinkel der Beine an HOBBIT. Dieser verarbeitet diese Anforderungen in einer synchron getakteten bidirektionalen Kommunikation mit MONSTER. MONSTER ist ebenfalls synchron über die in AP4200: Schnittstellen zwischen realer Steuerung und Simulation beschriebene Schnittstelle an VEROSIM®angekoppelt, worüber Sensorwerte und Aktuator-Sollwerte ausgetauscht werden.

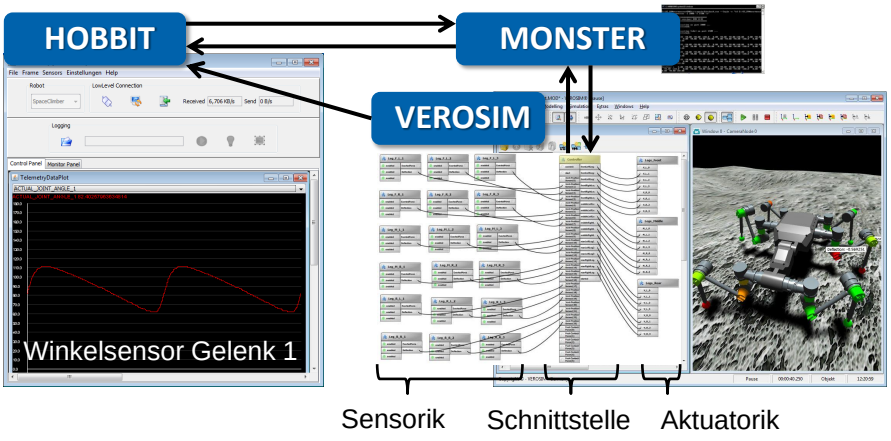


Abbildung 46: *Beispielhafte Kommunikationsinfrastruktur zwischen VEROSIM® und den High-Level-Controllern HOBBIT bzw. dem Pattern-Generator MONSTER des DFKI*

Im Rahmen der Präsentation auf der 2. Nationalen Konferenz zur Raumfahrt-Robotik (siehe Kapitel 1.1) musste die Schnittstelle zu externen Komponenten erweitert und überarbeitet werden:

- Erweiterung um UDP-Schnittstelle
- Möglichkeit zum Logging und Playback der Kommunikation
- Erweiterung der allgemeinen XML-Beschreibung des Kommunikationsprotokolls

Dies eröffnet auch neue flexiblere Möglichkeiten zur Nutzung des Systems im Anschluss an das Projekt Virtual Crater.

AP5300: Anpassung von Simulationsmodellen und -system an reale Ergebnisse der Referenzexperimente

Farbabhängiges Fehlermodell für den simulierten Laserscanner

Experimente am DFKI mit dem Laserscanner vom Typ Hokuyo URG-04LX haben gezeigt, dass die Unsicherheit der Entfernungsmessung stark von der Farbe des gescannten Materials abhängt (siehe AP5410). Mittlerweile kann im simulierten Laserscanner in VEROSIM® die Farbinformation zusammen mit der Tiefeninformation aus der Szene ausgelesen und im IO-Framework zur Verfügung gestellt werden. Des Weiteren ist ein farbabhängiges Fehlermodell implementiert worden, welches an die Referenzexperimente angepasst werden kann.

VEROSIM®-Scheduling-Komponente und Performanz

Zu diesem Projektzeitpunkt wurde festgestellt, dass für die Simulation eines allumfassenden Modells noch keine Echtzeit erreicht werden kann. Daher wurde zur besseren Analyse und Optimierung des Laufzeitverhaltens die zentrale Scheduling-Komponente des Simulationssystems überarbeitet. Ziel war es, auf diese Weise auf der einen Seite ein weitgehend echtzeitfähiges interaktives Simulationsmodell als auch auf der anderen Seite ein rechenintensives, nicht-echtzeitfähiges Modell zur Detailsimulation auf derselben Softwarebasis zu realisieren.

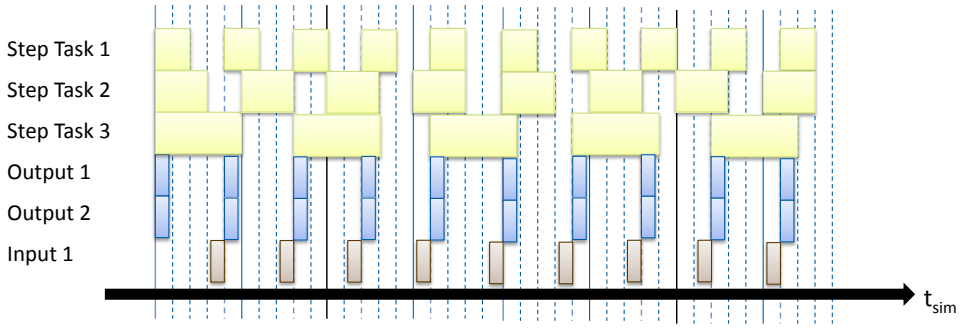


Abbildung 47: *Varianten von Tasks in der neuen VEROSIM®-Scheduling-Komponente VSS*

Abb. 47 zeigt eine beispielhafte Zusammenstellung verschiedener Typen von Aufgaben (Tasks), wie sie von der neuen Scheduling-Komponente *VSS* unterstützt werden. Typen von Tasks:

- *Step*: Einstellbare Aufrufzeit
- *Input*: Aufruf direkt nach dem Rendern (mit Renderzeit)
- *Output*: Aufruf direkt vor dem Rendern (mit Renderzeit)
- *RunState*: Nur Simulation Start/Stop wird weitergeleitet

Der wichtigste Typ *Step* wird in Abb. 48 noch einmal detailliert betrachtet. Die Ausführung wird in drei Phasen unterteilt: *Load*, *Execute* und *Save*. Im *Load*-Schritt werden die zur Berechnung notwendigen Daten aus der Datenbasis ausgelesen. Im *Execute*-Schritt finden die Berechnungen statt, die von der Datenbasis entkoppelt sind und somit parallelisiert werden können. Im *Save*-Schritt werden die Ergebnisse in die Datenbasis zurück geschrieben. Der n-te *Load*-Aufruf eines *Step-Tasks* mit der Aufrufzeit t_c erfolgt bei Simulationszeit $t_{sim} = n \cdot t_c$. Der *Execute*-Schritt wird zum gleichen Simulationszeitpunkt aufgerufen, das spielt aufgrund der Entkopplung aber keine Rolle. Es lässt sich eine Dauer Δt der *Step-Tasks* einstellen, so dass der dazu gehörige n-te *Save*-Aufruf zum Simulationszeitpunkt $t_{sim} = n \cdot t_c + \Delta t$ erfolgt. Dadurch lassen sich Nebenläufigkeiten und Kaskadierungen von *Tasks* frei konfigurieren.

Da die Menge an *Tasks* in einem komplexen Modell recht schnell ansteigt, wurde auch ein Analyse-Werkzeug entwickelt, mit dem sich der Simulationsablauf in gewähltem Ausschnitt betrachten lässt (siehe Abb. 49). Das Werkzeug dient auch zur Performance-Analyse, da sich sofort Diskrepanzen zwischen Rechenzeit und Simulationszeit von *Tasks* identifizieren lassen.

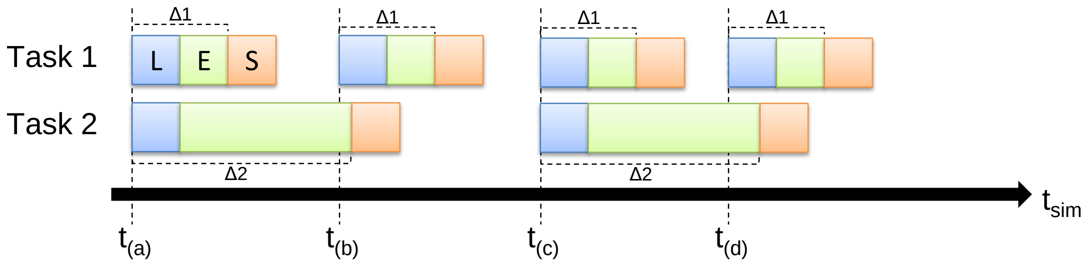


Abbildung 48: *Beispiel für die Aufrufreihenfolge der Phasen eines Step-Tasks*

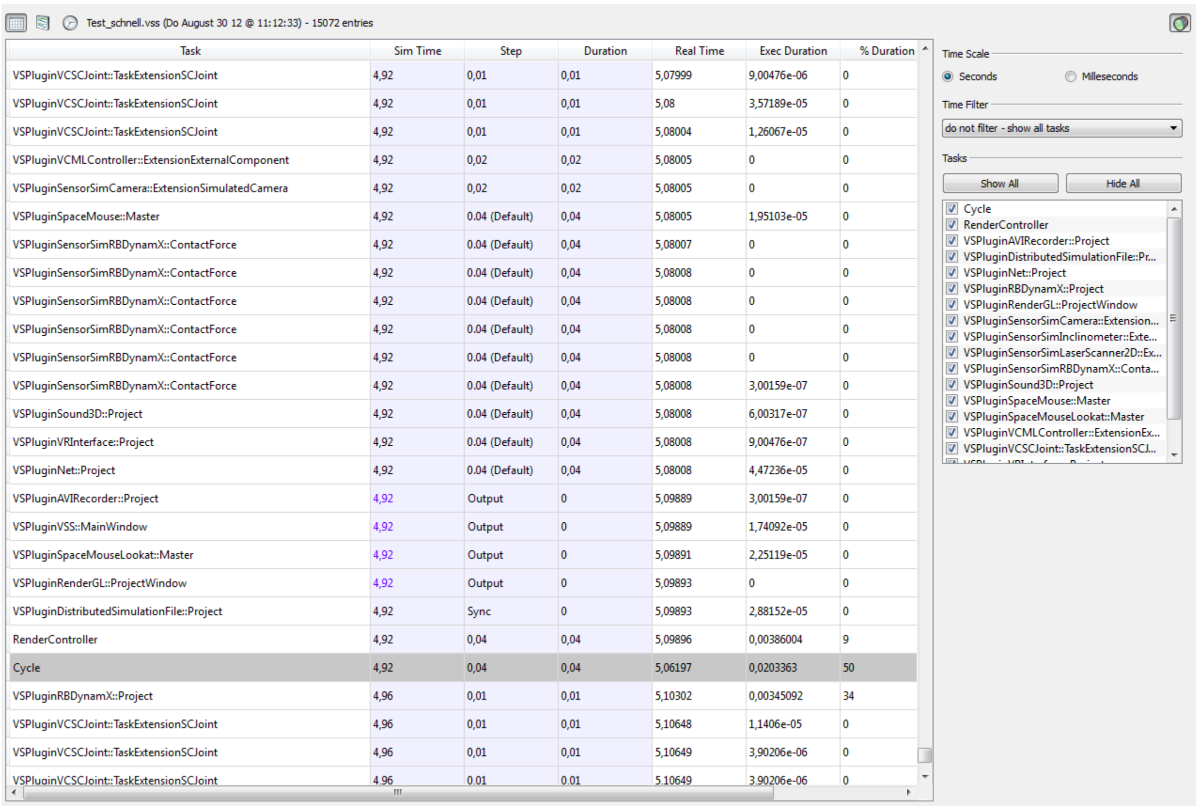


Abbildung 49: *Analyse-Werkzeug zur Betrachtung von Tasks, Simulationsablauf und Performance*

Trennung von Null- und Initialstellung an Gelenken

Bislang hat VEROSIM® an Gelenken in Mehrkörpermodellen nicht zwischen Null- und Initialstellung unterschieden. Der Initial-Zustand eines Gelenks zum Lade-Zeitpunkt des Modells war damit automatisch gleich seiner Nulllage. Beim SpaceClimber beispielsweise entspricht die Nullstellung einer Streckstellung aller 6 Beine. Dies ist als Initialstellung für eine realitätsnahe Simulation ungeeignet. Um dieses Problem zu lösen, verfügen Gelenke jetzt über zwei zusätzliche Verweise *rigidBody0JointFrame* und *rigidBody1JointFrame* (siehe Abb. 50 links) auf *Frames*, also Lagen im Raum. Darüber kann jetzt die Nullstellung des Gelenks definiert werden. Die beiden *Frames* müssen logisch unter den beiden beteiligten Starrkörpern angeordnet sein. Sie definieren dabei die Lage des Gelenkkoordinatensystems vom jeweiligen Körper aus gesehen. Die Nullstellung ist durch die Identität der Lage beider Koordinatensysteme gekennzeichnet. Wird nun das Gelenk bewegt, bewegen sich die beiden beteiligten Starrkörper gegeneinander und somit auch die beiden Koordinatensysteme.

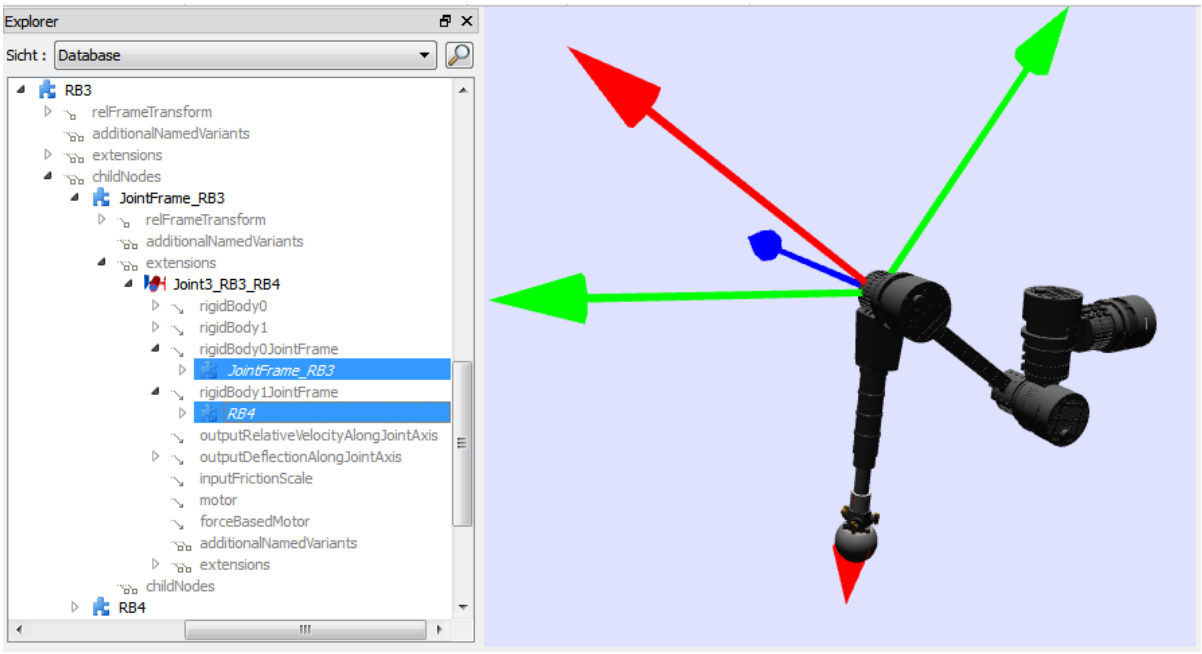


Abbildung 50: *Getrennte Modellierung von Null- und Initialstellung eines Gelenks im VEROSIM®- Explorer*

Abb. 50 zeigt ein Bein des SpaceClimbers. Selektiert sind die beiden Koordinatensysteme *JointFrame_RB3* und *RB4*, welche die Lage des Gelenks zwischen dem 3. und 4. Teilkörper des SpaceClimber-Beins (Joint3_RB3_RB4) jeweils von den beiden Teilkörpern aus definieren. Das Gelenk ist als Rotation um die z-Achse (blau) modelliert. Die x-Achse (rot) zeigt jeweils entlang der Körper-Achse des 3. und 4. Teilkörpers.In gestreckter Stellung (Nulllage) lägen beide Koordinatensysteme übereinander. Die Auslenkung des 4. Teilkörpers nach unten ist durch die Rotation der beiden Koordinatensysteme gegeneinander um die gemeinsame z-Achse definiert.

Somit ist die Nulllage zu jedem Zeitpunkt fest definiert und es kann auch eine davon abweichende Initialstellung gespeichert werden. Dann sind beim geladenen Modell die beiden Koordinatensysteme eben nicht identisch und deren Abweichung definiert die initiale Auslenkung von der Nulllage.

AP5400 Durchführung der Referenzexperimente

Wie im Antrag zur Projektverlängerung dargelegt, wurde die verlängerte Laufzeit unter anderem zur Durchführung diverser Referenzexperimente genutzt. Entsprechend einer Rechereche der am Markt verfügbaren Systeme, die als repräsentativ für diese Gruppe von Sensoren betrachtet werden können, wurden folgende Systeme betrachtet:

- SwissRanger SR4000 3D Camera, 10m Reichweite, 44° Öffnungswinkel
- KVH Fluxgate Kompass Sensor C-100
- IG-500A Sub-miniature Attitude and Heading Reference System (AHRS)

Der Grund für diese Auswahl ist, dass das entwickelte Sensorsimulations-Framework für ein möglichst breites Spektrum realer Sensoren für das relevante Anwendungsgebiet eingesetzt werden können soll. Daher wurden diese realen, aktuellen und marktüblichen Sensoren aus dem Bereich der mobilen Robotik dahingehen untersucht, in wie weit sie sich bereits jetzt in das Simulationsframework einbinden lassen bzw. welche Anpassungen noch notwendig sind, um solche Sensoren realitätsnah nachbilden zu können. Ziel ist jetzt der Vergleich der bereits in Virtual Crater eingesetzten Algorithmen zur Simulation dieser Sensoren mit Messergebnissen realer Sensoren. Durch die Auswertung dieser Sensoren konnte daher das Projektergebnis weitergehend verifiziert und optimiert werden.

Kraft-Momenten-Sensor

In dem Laufroboter *SpaceClimber* des DFKI sind in den Gelenken jeweils zwischen den beiden beteiligten Teilkörpern 6-Komponenten-Kraft-Momenten-Sensoren verbaut (siehe Abb. 51), welche die lokal auftretenden Kräfte und Drehmomente in allen 6 Freiheitsgraden messen.

Abbildung 51: *6-Komponenten-Kraft-Momenten-Sensor vom Typ ATI Mini 45*

In der Starrkörperdynamiksimulation wird ein solcher Sensor abgebildet, indem eine starre Verbindung zwischen zwei Körpern eingeführt wird. Die 6 Freiheitsgrade zwischen den Körpern werden durch 6 Zeilen in der Jacobi-Matrix der Zwangsbedingungen entfernt. Beim Lösen des LCP werden die zur Einhaltung der Zwangsbedingung notwendigen Kräfte und Momente berechnet und können von einem Sensormodell über das E/A-Framework ausgegeben werden. Realer und virtueller Kraft-Momenten-Sensor wurden einer Reihe von Experimenten unterzogen. Dabei wurden die auf alle Achsen aufgebrachten Kräfte und Momente durch Belastung mit Gewichten an unterschiedlichen Hebellängen variiert.

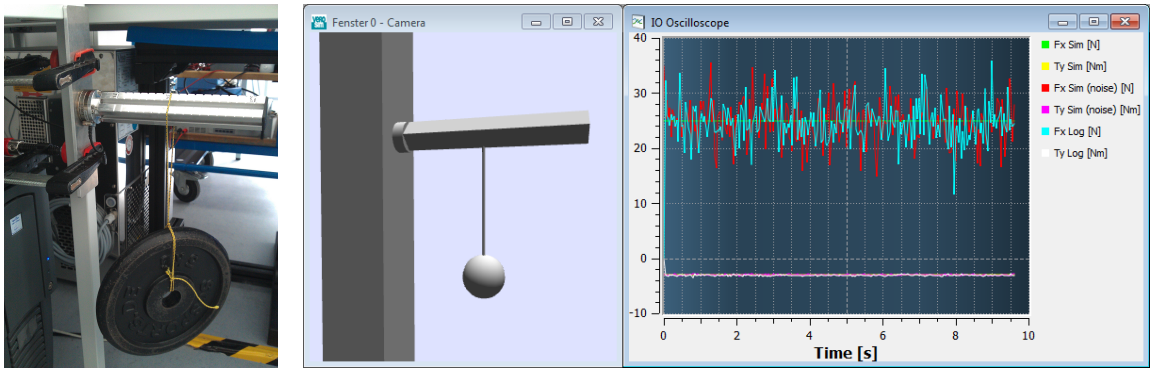


Abbildung 52: *Reales (links, durchgeführt vom DFKI) und virtuelles Referenzexperiment zum Kraft-Momenten-Sensor*

Abb. 52 zeigt den Aufbau des Experiments zur Messung einer Kraft in x-Richtung und eines Drehmoments in y-Richtung. Zum Vergleich werden in der Simulation die realen aufgenommenen Messwerte über eine Playback-

Mechanismus abgespielt und können zur Simulationslaufzeit grafisch dargestellt werden. In grün und gelb sind ideale Kraft bzw. ideales Moment des virtuellen Sensors aufgetragen, in rot und violett die durch ein Fehlermodell verrauschten und in türkis und weiß die vom realen Sensor aufgenommenen Werte. Für diesen Sensor wurde ein einfaches Fehlermodell verwendet, das normalverteilte Zufallszahlen auf den Messwert addiert. Die Breite der Verteilung aus der die Zufallszahlen gezogen werden ist dabei eine Kombination aus dem relativen und absoluten Fehler der Messung, die aus den Messreihen berechnet wurden. Während der Durchführung der Referenzexperimente ist anhand der Daten des Simulationsmodells ein Fehler in der Verortung des Referenz-Koordinatensystems im realen Sensor aufgefallen.

Inertialnavigationssystem (IMU) IG 500-A

Die IMU und das entsprechende Simulationsmodell wurden bereits in AP4330: Sensorik vorgestellt. Zur Verifikation des Simulationsmodells und Kalibration auf einen bestimmten Sensor ist es notwendig, realen und virtuellen Sensor eine Reihe von Trajektorien durchlaufen zu lassen. Dabei muss die reale Trajektorie möglichst genau bekannt sein, um Abweichungen zwischen Realität und Simulation zu minimieren. Abb. 53 zeigt das Miniatur-*Attitude and Heading Reference System (AHRS)* der Firma SBG Systems, welches über die Eigenschaften einer IMU (Gyroskope und Accelerometer in allen drei Achsen) hinaus noch Magnetometer in allen drei Achsen besitzt.

Abbildung 53: *Miniatur-Attitude and Heading Reference System (AHRS) der Firma SBG Systems vom Typ IG-500A*

Für die Referenztrajektorien konnte ein Kuka Leichtbauroboter (LBR) des *Planetary Landing*-Mockups aus dem Projekt FastMap verwendet werden.Abb. 54 zeigt eine Konfiguration der Bedienoberfläche des hybriden Testbeds zur Durchführung der Referenzexperimente. „Hybrid“ deshalb, da die Simulationsumgebung gleichzeitig zur Steuerung der Leichtbauroboter (Details zur Robotersteuerung siehe [RSS12a]) und zur Simulation des virtuellen Experiments genutzt wird. Datenaufnahme und -darstellung erfolgen ebenfalls mit der Simulationssoftware. Das Grafikfenster (links oben) zeigt die Stellung des Roboters im virtuellen Experiment, das entspricht gleichzeitig der Soll-Stellung des realen Roboters. Die gemessenen Gelenkwinkelstellungen des realen Roboters werden als blaue halbtransparente Überlagerung dargestellt. Abweichungen sind nicht zu vermeiden, aber relativ gering. Die Messkurven unten zeigen links die Winkelgeschwindigkeiten und rechts die Achsbeschleunigungen entlang der Koordinatenachsen des Sensors (simuliert: x, y, z = grün, gelb, rot; real gemessen: x, y, z = violett, türkis, weiß). Hier wurden zunächst nur die idealen virtuellen Messwerte dargestellt, diese noch mit Fehlermodellen zu belegen ist natürlich möglich, wie in AP4330: Sensorik erläutert und am Beispiel des Kraft-Momenten-Sensors gezeigt. Die Mittelwerte der realen Messungen für die Beschleunigungen stimmen gut mit den idealen Messwerten überein. Die etwas zu geringen und stark verrauschten Messwerte der Winkelgeschwindigkeiten sind vermutlich auf Vibrationen des Leichtbauroboters zurück zu führen und bedürfen weiterer Untersuchungen.

Kompass KVH Industries C100

Für den Kompass wurden analog zu dem Referenzexperiment für die IMU ebenfalls Testtrajektorien mit dem FastMap-Mockup abgefahren und die realen Sensorwerte mit den simulierten Sensorwerten verglichen.

Bei diesem Experiment ist aufgefallen, dass die Magnetfelder der LBR-Antriebe den Kompass relativ stark stören. Es ergeben sich Fehler in der gemessenen Nord-Süd-Achse von bis zu 50° am TCP des LBR. Bei einem Versatz von 50cm Abstand vom TCP reduzieren sich die Störungen auf ca. 15°. Falls in einem mobilen Roboter ein Kompass eingesetzt werden soll, muss entweder ein ausreichender Abstand zu Antrieben und anderen

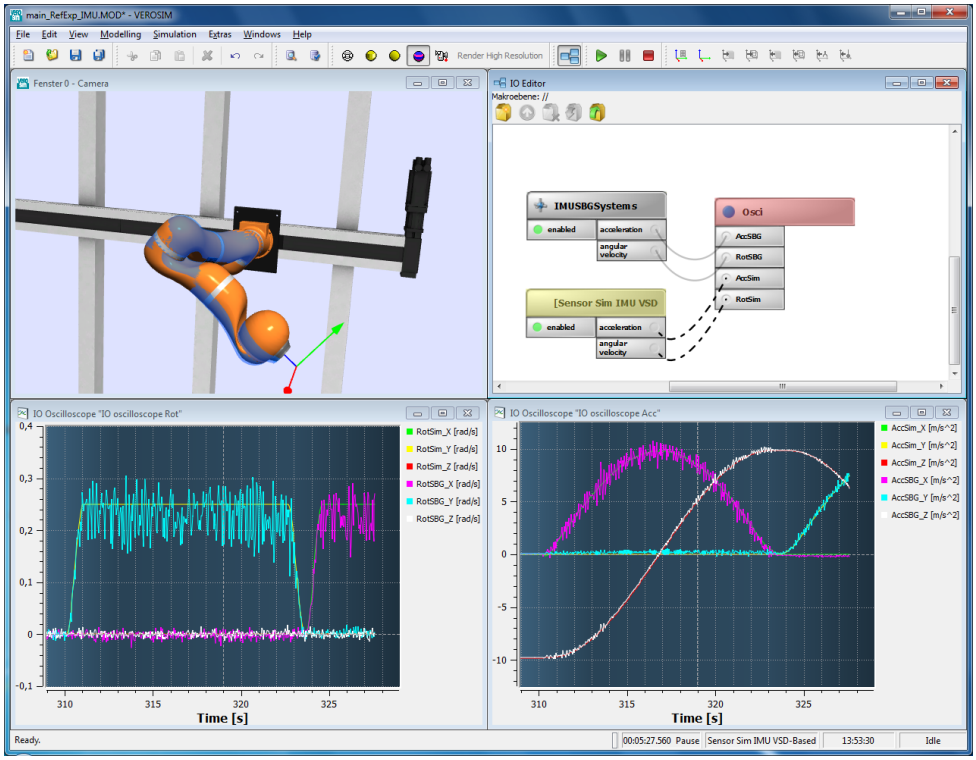


Abbildung 54: *Hybrides Testbed-Setup zur Ansteuerung des Leichtbauroboters, Datenaufnahme und Auswertung der IMU-Referenzexperimente*

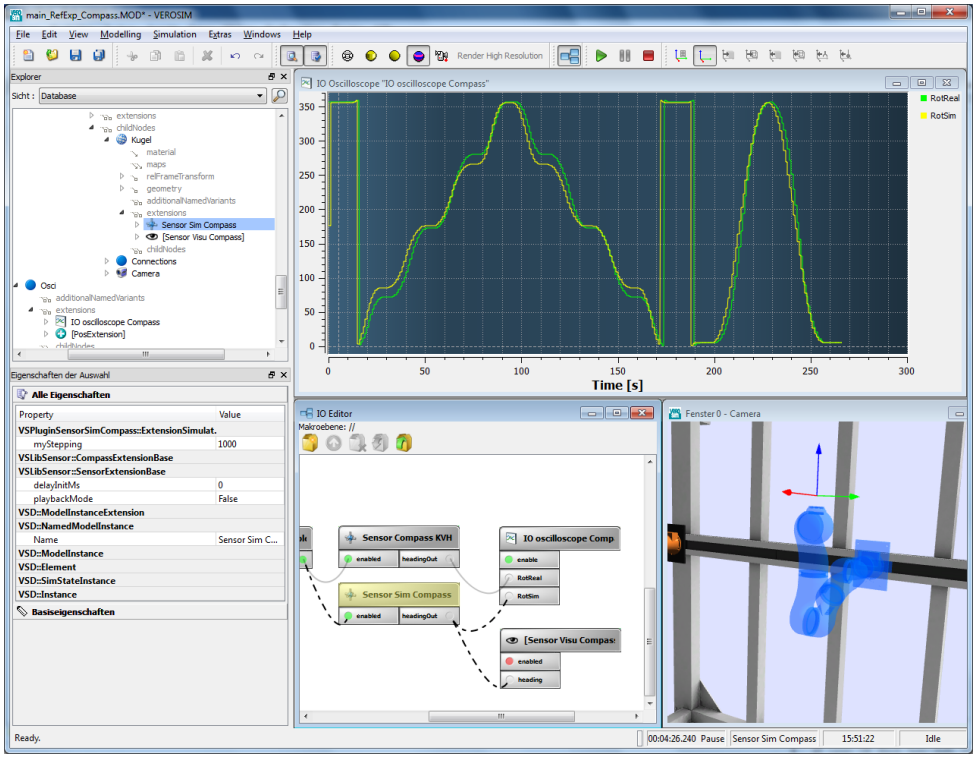


Abbildung 55: *Hybrides Testbed-Setup für Referenzexperimente mit dem Kompass*

Komponenten mit starken Magnetfelder gewährleitet sein oder solche Komponenten müssen entsprechend abgeschirmt werden.

PMD-Tiefenkamera SR4000

Mit der PMD-Tiefenkamera vom Typ SR4000 sind Aufnahmen des Oberflächenmodells des FastMap-Mockups gemacht weorden, da dieses relativ hoch aufgelöst ist und nach einem CAD-Modell gefertigt wurde, das für die virtuellen Experimente ebenfalls zur Verfügung steht (siehe Abb. 56). Es wurden Aufnahmen im Abstandsmessbereich 500-850mm gemacht. Die durchschnittliche Abweichung beträgt 2,5mm (siehe Abb. 57)

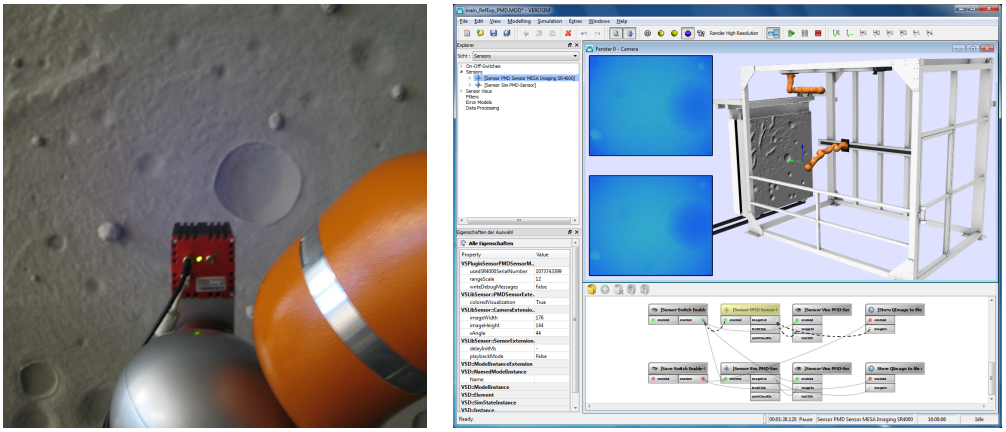


Abbildung 56: *Hybrides Testbed-Setup für Referenzexperimente mit der PMD-Tiefenkamera*

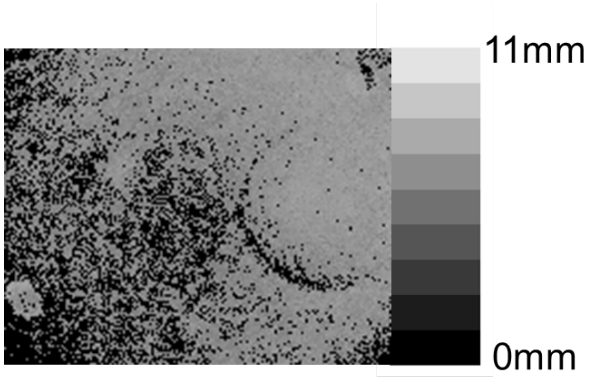


Abbildung 57: *Differenzbild zwischen realer und virtueller Aufnahme der PMD-Tiefenkamera*

Auswertung und Optimierung des Laufzeitverhaltens einer ganzheitlichen Spaceclimber-Simulation

Um festzustellen, in welchem Umfang ein ganzheitliches Simulationsmodell die Echtzeitanforderung erfüllt, wurden entsprechende Messungen durchgeführt. Das Testmodell besteht aus Laufroboter („Spaceclimber"), 24 Motordynamikmodellen, Umgebung und Visualisierungsmetaphern wie dem Oszilloskop und Metaphern zur Colorierung von Motoren entsprechend dem aktuellen Drehmoment, vgl. Abbildung 58. Die Simulation wurde auf einem verteilten System entsprechend der am DFKI aufgebauten Mehrfachprojektion durchgeführt.

Es wurde festgestellt, dass ein solches Modell zunächst noch nicht in Realzeit simuliert werden kann, die Simulationszeit verläuft ungefähr um den Faktor 4 verzögert zur Echtzeit.

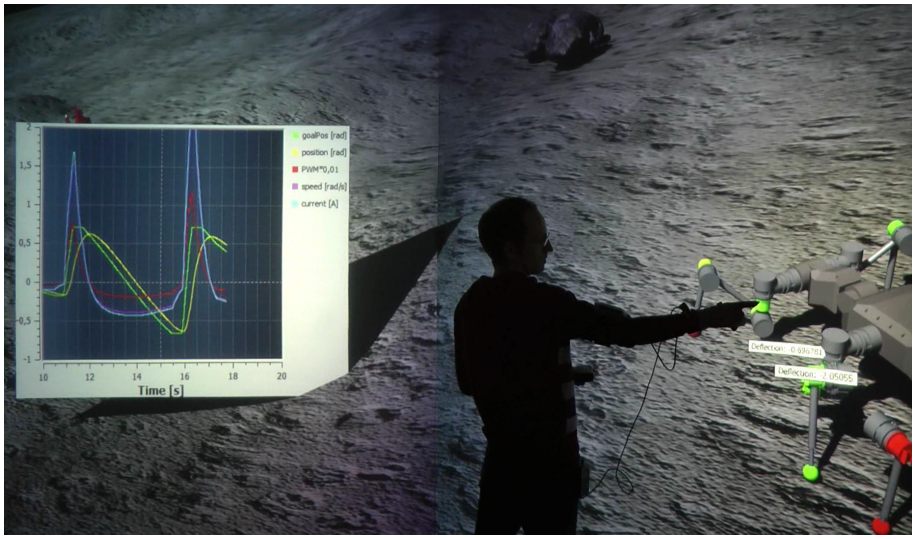


Abbildung 58: *Das ganzheitliche Simulationsmodell zur Bewertung des Laufzeitverhaltens*

Als erster Schritt zur Verbesserung des Laufzeitverhaltens wurde die Realisierung der Oszilloskop-Visualisierung in einer Mehrfachprojektion verbessert. Anstatt den generischen in VEROSIM® verfügbaren Verteilungsmechanismus zu nutzen, der notwendigen Daten für die Oszilloskopvisualisierung - unnötigerweise - auf alle Slaves des Simulationsclusters zu übertragen, werden sie nun mittels eines speziellen Mechanismus nur auf diejenigen Slave-Rechner übertragen, die den entsprechenden Bildbereich bedienen und somit die Daten tatsächlich benötigen. Mit dieser Optimierung ließ sich das Laufzeitverhalten bereits auf einen Faktor von ca. 2 x Echtzeit verbessern.

Weitere kleinere Optimierungen und die Überarbeitung der Scheduling-Komponente wurden im Zeitraum der Projektverlängerung durchgeführt. Dadurch war es möglich zum Projektende die Referenzmission in Echtzeit in der Stereoprojektion zu demonstrieren.

AP6200: Modellierung and Implementierung einer virtuellen Referenz-Mondmission

Eine fiktive Mondkraterumgebung, die als Ausschnitt ein Modell der Kraterwand aus der DFKI Weltraumexplorationshalle enthält, wurde vom DFKI in VEROSIM® bereitgestellt. 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. 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.

Basierend auf einer Weiterentwicklung des Simulationssystems wurde zunächst das Modell des „SpaceClimbers“ nochmal grundlegend hinsichtlich des Aufbaus aus Untermodellen und Verwaltung von Referenzen zwischen den Untermodellen umstrukturiert. Das erleichtert die Wartung und Weiterentwicklung des Modells und vermeidet doppelte Datenhaltung bei parallelen Entwicklungen, wie z.B. einem SpaceClimber mit vereinfachtem und komplexem Motormodell. Es wurde die Möglichkeit geschaffen, Gruppen von Visualisierungsmetaphern (Einfärben der Gelenke entsprechend des Lastmoments) über Eingabegeräte im Cave-System ein- und auszuschalten. Des Weiteren kann mittlerweile vollständig auf die Darstellung der Datenvisualisierung (Oszilloskop, Balkendiagramm) auf dem Simulationsmaster im Cave-System verzichtet werden, was zu einer Verbesserung des Laufzeitverhaltens führt. Des Weiteren wurde ein Selektionsmechanismus entwickelt, der die Auswahl der zu visualisierenden Sensorwerte in der Rundumprojektion ermöglicht (siehe Abb. 59).

Es wurde ein aktualisiertes detaillierteres Geometriemodell des „SpaceClimbers“ in das VEROSIM®-Format überführt und mit den entsprechenden physikalischen Eigenschaften und Sensoren ausgestattet. Das Modell

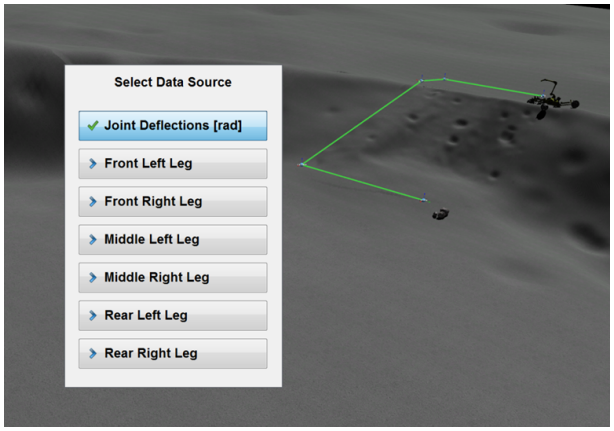


Abbildung 59

beinhaltet jetzt auch den Kopf mit Kamera und Laserscanner. Die geglätteten Werte des Neigungssensors wurden erfolgreich genutzt, um den Anstellwinkel der Beine zu steuern und so eine günstigere Haltung des Laufroboters bei der Bewältigung von Steigungen zu erzielen.

Entsprechend den neuen Anforderungen aus der Definition der Referenzmission (AP6100) unterstützt VEROSIM® jetzt auch Spot-Lights für den Sherpa-Scheinwerfer und zweifach Schatten von verschiedenen Lichtquellen. Die einfache Pfadplanungskomponente, die abhängig von einer Zielvorgabe Bewegungsbefehle an den HOBBIT-Controller sendet, wurde dahingehend erweitert, dass nun auch Listen von Wegpunkten nacheinander angesteuert werden können. Diese für den Laufroboter geplante Route kann auch visualisiert werden. Zur vereinfachten Navigation durch das Modell der Referenzmission in der Rundumprojektion wurde eine flexibel konfigurierbare Bedienoberfläche geschaffen, mit der ausgewählte Aspekte der Simulation gesteuert und einzelne Checkpoints angesprungen werden können. Das bedeutet, dass sowohl die Ansicht dorthin gleitet, als auch der Laufroboter an diese Position gesetzt wird. Das funktioniert auch bei laufender Simulation, die Simulationsalgorithmen (insbesondere die Mehrkörperdynamik) reagieren auf Änderungen ihrer Zustandsgrößen. Die Neuentwicklungen sind in Abb. 60 zusammengefasst.

AP6300: Demonstration Referenz-Mondmission

In der virtuellen Mondkraterumgebung, die als Ausschnitt das Modell der Kraterwand aus der DFKI-Weltraumexplorationshalle enthält, beginnt der SpaceClimber seine Mission beim Modell des Sherpa am oberen Kraterrand. Von dort aus läuft er eine vordefinierte Route von Wegpunkten ab: Ein günstiger Einstiegspunkt am Rand des Kraters, der untere Rand des Kraters, ein geeigneter Felsbrocken zur Entnahme von Bodenproben und die Rückkehr zum Ausgangspunkt. Dabei kann er seine Umwelt über Kamera und Laserscanner erfassen. Das Beleuchtungsmodell kann zwischen verschiedenen Uhrzeiten wechseln, bis hin zu völliger Dunkelheit im Krater, die dann nur durch den Scheinwerfer des Sherpa durchbrochen wird. Das Verhalten des Bodenmechanikmodells wird durch die Visualisierung der Fußabdrücke nachvollziehbar. Das zunächst eingestellte schnellere Laufmuster ist nicht geeignet, um die Steigung des Kraterrands auf dem Rückweg zu bewältigen, der Laufroboter rutscht ab. Wählt man ein langsameres Laufmuster, bei dem immer mindestens vier Füße Bodenkontakt haben, kann die Steigung bewältigt werden.

Das Projektergebnis wurde dem Auftraggeber am 30.10.2012 am DFKI in Bremen präsentiert. Dabei wurden sowohl die erreichten Ziele in einer Präsentation dargestellt, als auch der praktische Nutzen anhand einer Demonstration des Arbeitsablaufes sowohl am PC-Arbeitsplatz (Laptop) als auch in der Stereoprojektionsumgebung veranschaulicht.

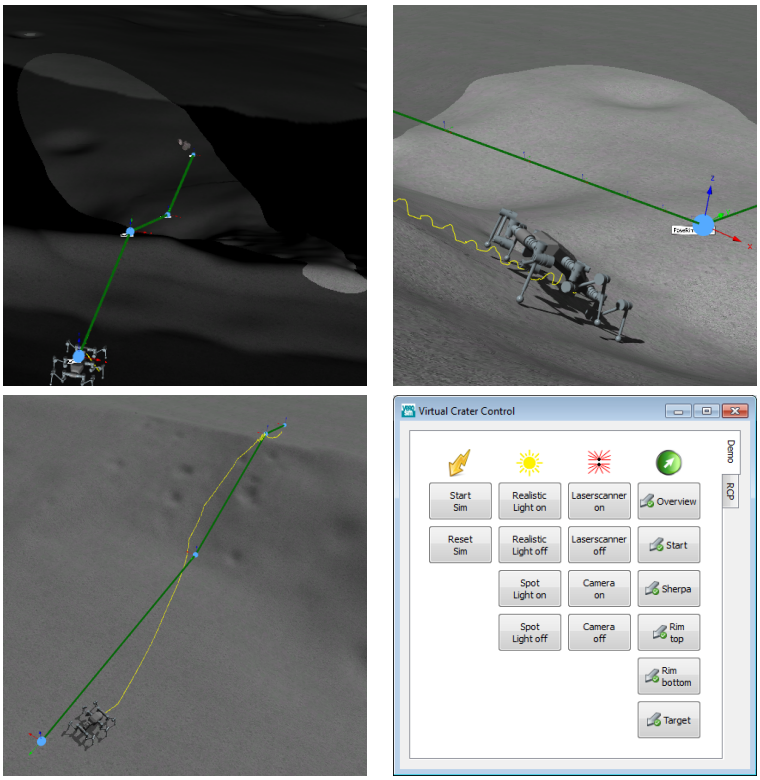


Abbildung 60: *Oben links: Beleuchtung durch einen Scheinwerferkegel.*
Oben rechts: Automatisches Anstellen der Beine zur Verbesserung des Laufmusters auf Schrägen.
Unten links: Sequentielles Ansteuern von Wegpunkten.
Unten rechts: Konfigurierbare Bedienoberfläche für das Modell der Referenzmission.

Literatur

[Bek96] Bekker, M.: *Theory of Land Locomotion: Mechanics of Vehicle Mobility*. University of Michigan Press, 1996

[BLB05] Bauer, R. ; Leung, W. ; Barfoot, T.: Development of a Dynamic Simulation Tool for the ExoMars Rover. In: *i-SAIRAS Proceedings, Munich, Germany*, 2005

[Joz02] Jozwowski, Timothy R.: *Real Time Photon Mapping*, Michigan Technological University, Diplomarbeit, 2002

[JRKR11] Jung, Thomas J. ; Rast, Malte ; Kaigom, Eric G. ; Rossmann, Jürgen: Fast VR Application Development Based on Versatile Rigid Multi-Body Dynamics Simulation. In: *Proceedings of the ASME 2011 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (IDETC/CIE)*. Washington, DC, August 2011

[LKC+09] Langosz, M. ; Kuehn, D. ; Cordes, F. ; Yoo, Y.-H. ; Kirchner, F.: Concept Evaluation of Modeling Terrain Mechanics by a Neural Network. In: *ISTVS 2009*, 2009

[MRP+04] Michaud, S. ; Richter, R. ; Patel, N. ; Thuer, T. ; Huelsing, T. ; Joudrier, L. ; Siegwart, R. ; Ellery, A.: Rover Chassis Evaluation Tool (RCET). In: *ASTRA Proceedings*. Noordwijk, The Netherlands, 2004

[PSE04] Patel, N. ; Scot, G. ; Ellery, A.: Application of Bekker Theory to Wheeled, Tracked and Legged Vehicles. In: *SPACE 2004*. San Diego, California, USA, September 2004

[RJR10a] Rossmann, J. ; Jung, T. ; Rast, M.: Developing Virtual Testbeds for Mobile Robotic Applications in the Woods and on the Moon. In: *Proceedings of the International Conference on Intelligent Robots and Systems (IROS)* IEEE/RSJ, 2010, S. 4952–4957

[RJR10b] Rossmann, J. ; Jung, T. ; Rast, M.: Entwicklung Virtueller Testbeds mit Dynamik- und Bodenmechaniksimulation für Aufgaben in Forschung und Entwicklung. In: *Augmented und Virtual Reality in der Produktentstehung*. J. Gausemeier and M. Grafe, 2010

[RJR10c] Rossmann, J. ; Jung, Thomas J. ; Rast, Malte: Developing Virtual Testbeds for Tasks in Research and Engineering. In: *Proceedings of the ASME WinVR*, 2010

[RRJGK11] Rossmann, J. ; Rast, M. ; Jung, T. ; Guiffo Kaigom, E.: Modellierung und Synchronisation von Simulationsaufgaben für Virtuelle Testbeds. In: *Wissenschaftsforum 2011 Intelligente Technische Systeme - 10. Paderborner Workshop Augmented & Virtual Reality in der Produktentstehung* Bd. 295. Heinz Nixdorf Institut, Universität Paderborn : Gausemeier, J. and Grafe,M. and Meyer auf der Heide, F., 2011, S. 145–156

[RSJR09] Rossmann, J. ; Schluse, M. ; Jung, T. ; Rast, M.: Interaktive integrierte Starrkörperdynamik- und Schüttgutsimulation. In: Grafe, J. Gausemeier; M. (Hrsg.): *Augmented & Virtual Reality in der Produktentstehung* Bd. 252. Heinz Nixdorf Institut, Universität Paderborn, 2009, S. 31–48

[RSS12a] Rossmann, J. ; Schlette, C. ; Springer, M.: Kinematic Robot Control for a Planetary Landing Mockup. In: Bruzzone, A. (Hrsg.): *Proceedings of the 20th IASTED International Conference on Applied Simulation and Modelling (ASM)*. Napoli, Italy, 2012, S. 134–141

[RSS+12b] Rossmann, Juergen ; Schluse, Michael ; Sondermann, Bjoern ; Emde, Markus ; Rast, Malte: Advanced Mobile Robot Engineering with Virtual Testbeds. In: *Robotics; Proceedings of ROBOTIK 2012; 7th German Conference on* (2012), may

[SMH+] Smith, M. J. ; Moore, T. ; Hill, C. J. ; Noakes, C. J. ; Hide, C.: *Simulation of GNSS/IMU Measurements*. `http://www.isprs.org/commission1/theory_tech_realities/pdf/p26_s5.pdf`. – Institute of Engineering Surveying and Space Geodesy (IESSG) The University of Nottingham

[Tsa86] Tsai, Roger Y.: An Efficient and Accurate Camera Calibration Technique for 3D Machine Vision. In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 1986, S. 364–374

[YH02] Yoshida, K. ; Hamano, H.: Motion dynamics and control of a planetary rover with slip-based traction model. In: *Proc. SPIE Vol. 4715, p. 275-286*, 2002

[YIMS08] Yoshida, A. ; Ihrke, M. ; Mantiuk, R. ; Seidel, H.: Brightness of the glare illusion. In: *Proceedings of the 5th Symposium on Applied Perception in Graphics and Visualization* ACM, 2008

[YJR+10] Yoo, Y.-H. ; Jung, T. J. ; Römmermann, M. ; Rast, M. ; Kirchner, F. ; Rossmann, J.: Developing a Virtual Environment for Extraterrestrial Legged Robots with Focus on Lunar Crater Exploration. In: *The 10th International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS 2010)*. Sapporo, Japan, 2010, S. 206–213

[Yoo07] Yoo, Y.-H.: An Energy-Based Approach towards Modeling of Mixed Reality Mechatronic Systems. Version: 2007. `http://dx.doi.org/10.1007/978-1-84628-974-3_16`. In: *Lecture Notes in Control and Information Sciences* Bd. 360. 2007. – DOI 10.1007/978–1–84628–974–3_16, S. 177–184