INVERITAS - Innovative Technologien zur Relativnavigation (-bewegung) und Capture mobiler autonomer Systeme

DFKI Robotics Innovation Center

22. August 2012

Das diesem Bericht zugrunde liegende Vorhaben wurde von der Raumfahrt-Agentur des Deutschen Zentrums für Luft- und Raumfahrt e.V. mit Mitteln des Bundesministeriums für Wirtschaft und Technologie aufgrund eines Beschlusses des Deutschen Bundestages unter dem Förderkennzeichen 50RA0910 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.

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

Inhaltsverzeichnis

1 Einleitung

Das vorliegende Dokument stellt den Abschlussbericht des Projektes "INVERITAS - Innovative Technologien zur Relativnavigation (-bewegung) und Capture mobiler autonomer Systeme" dar. INVERITAS wurde von der Raumfahrt-Agentur des Deutschen Zentrums für Luft- und Raumfahrt e.V. mit Mitteln des Bundesministeriums für Wirtschaft und Technologie aufgrund eines Beschlusses des Deutschen Bundestages unter dem Förderkennzeichen 50RA0910 gefördert.

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

2 Kurzdarstellung nach NKBF 98 8.2 I

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

2.1 Aufgabenstellung

INVERITAS war ein Verbundprojekt zwischen den Projektpartnern EADS-ASTRIUM, Jena-Optronik und dem Robotics Innovation Center des Deutschen Forschungszentrums für Künstliche Intelligenz (RIC DFKI Bremen).

Das Gesamtziel des INVERITAS Verbundprojektes war die prototypische Realisierung eines breit einsetzbaren Rendez-Vous und Capture (RvC) Systems und die Entwicklung der zugehörigen Kerntechnologien im Sinne des Anhebens des jeweiligen Technology Readiness Level (TRL) bis zur Demonstration am Boden (TRL 4). Entwickelt werden sollten alle Technologien, die notwendig sind, um einen Capture-Satelliten zu bauen, der in der Lage ist, andere auch durch Ausfälle möglicherweise unkooperative Satelliten im Orbit einzufangen und dann Wartungsarbeiten wie Betanken, Bahnänderungen etc. durchzuführen.

Die Aufgaben des DFKI RIC in INVERITAS beschäftigten sich primär mit dem Aufbau eines Langstreckenbewegungssimulationsystems (LBSS) in Hard- und Software sowie der Entwicklung von alternativen Greifmöglichkeiten.

Das LBSS wurde dabei direkt mit einem neu entwickelten modularen Software-Simulator und einer 3D Visualisierung gekoppelt. Es war das Ziel zu ermöglichen, verschiedene Arten der Bahnsteuerung eines Capture-Satelliten und des Clients zu implementieren und zu evaluieren.

Die physischen Technologiedemonstratoren für den Capture-Satelliten und für den Klienten werden durch einen Sechs-Achsen-Roboter und durch ein seilgeführtes 3D-Bewegungsystem (=Kabelroboter) zum LBSS zusammengeführt. Bei der Übertragung der Simulationsergebnisse auf das LBSS mussten die insgesamt 12 Freiheitsgrade des realen Capture-Satelliten und des Clients auf die eingeschränkten insgesamt 9-10 Freiheitsgrade des Roboters und des seilgeführten Bewegungssystems umgesetzt werden. Das LBSS selbst wurde in einer Umgebung aufgebaut, in der eine komplette Kontrolle der Lichtverhältnisse möglich ist. Dadurch können bis auf die reduzierte Schwerkraft und den niedrigeren Atmosphärendruck alle physikalischen Parameter eines RvC-Vorgangs simuliert, evaluiert und demonstriert werden.

Das Simulationssystem sollte auch zur Entwicklung von alternativen Greifmöglichkeiten verwendet werden. In der Simulation wurden mit Hilfe von Evolutionsstrategien und genetischen Algorithmen verschiedene neuartige Kinematiken und Greifstrategien entwickelt.

2.2 Voraussetzungen, unter denen das Vorhaben durchgeführt wurde

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

Somit verfügt das RIC über das nötige Know-How, um wie für INVERITAS erforderlich ein Langstreckenbewegungssystem mit stabilen und präzisen Bewegungsreglern sowie flexiblen Simulationskomponenten zu entwickeln, sowie über die erforderlichen Kenntnisse zur Entwicklung neuer Greifmechanismen für Capture-Vorgänge über simulationsbasierte Evolutionsstrategien und genetische Algorithmen, was ebenfalls ein Thema des Projekts INVERITAS war.

2.3 Planung und Ablauf des Vorhabens

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

Neben dem Design von Sensorsystemen lag der Projektfokus des DFKI-RIC vor allem auf der Entwicklung des Langstrecken Bewegungssimulationssystems. Parallel wurde an der Entwicklung neuer Capture-Systeme gearbeitet, unter Einsatz simulationsgestützter Evolutionsstrategien und genetischer Algorithmen.

Der Konsortialführer EADS ASTRIUM hat im zweiten Quartal 2010 den Projektfokus angepasst. Dies erfolgte in Abstimmung mit dem Projektträger. Der Projektplan wurde für Astrium und DFKI entsprechend angepasst.

Im dritten Quartal 2010 kam es in den zwei Arbeitspaketen "AP31224-D: Bewegungssystem für Demonstrationen" und "AP41200: Implementierung und Kalibrierung des Kernsystems" zu Verzögerungen, die eine Verlängerung dieser Arbeitspakete sowie eine Verschiebung von drei angrenzenden abhängigen Arbeitspaketen erforderlich machten. Die kostenneutrale Optimierung des Arbeitsplans in Absprache mit dem Projektträger konnte den erfolgreichen Abschluss des Projekts sicherstellen.

Um Verzögerungen im Projektverlauf grade im Hinblick auf die Präzision der Anlage abzufangen, wurde im vierten Quartal 2011 eine kostenneutrale Verlängerung des Projekts bis zum 31.3.2012 beantragt und bewilligt. So konnte neben erfolgreichen Closed-Loop Tests auch eine verbesserte Kalibrierung der Anlage vorgenommen wer-

den. Damit konnte das Projekt mit finalen Tests der Sensorik, der Servicer-Steuerung und der Anlagenpräzision erfolgreich abgeschlossen werden.

Um den Projektstatus zwischen den Projektpartnern und mit dem DLR auszutauschen, fanden Progress Meetings statt, die im Folgenden mit den behandelten Themen aufgelistet werden:

- 30.09.2009, Bremen, Themen DFKI:
  - Spezifikation und Beschaffung Langstreckenbewegungssimulationsystem
  - Spezifikation Kamerasystem für Multimodalen Sensorkopf
  - Entwurf RvC Simulationsystem
- 14.01.2010, Bremen, Themen DFKI:
  - Kamerasystem
  - Bewegungsystem für Demonstration
  - Implementierung und Kalibrierung Kern
  - Konzepte für neue Capture Systeme
  - Multimodales Sensorsystem
  - Bewegungssimulationssystem
  - RvC Evaluierung und Simulation
  - Demonstration Systemsteuerung und Anlagen im DFKI RIC (Halle)
- 06.05.- 07.05.2010, Bremen, Themen Astrium, DFKI und Jena-Optronik:
  - Multimodales Sensorsystem
  - Lidar
  - Nahbereichs-Relativ-GNC/REGIS
  - Mehraktor Systemsteuerung
  - Multimodales Sensorsystem
  - Bewegungssimulationssystem
  - RvC Evaluierung und Simulation
  - Demonstration Systemsteuerung und Anlagen im DFKI RIC (Halle)
- 02.03.-03.03.2011, Bremen, Themen Astrium, DFKI und Jena-Optronik:
  - Systemdemonstration im DFKI RIC
  - Multimodales Sensorsystem
  - Lidar
  - Nahbereichs-Relativ-GNC/REGIS
  - Multimodales Sensorsystem
  - Strukturelemente / Mock-ups
  - RvC Evaluierung und Simulation

  - Bewegungssimulationssystem

- 19.04.-20.04.2012, Endpräsentation, Bremen, Themen Astrium, DFKI und Jena-Optronik:
  - Hardware des Bewegungssimulationssystems
  - Kern des Simulationssystems
  - Koordinatentransformationen
  - Automatische Vermeidung von Systemkollisionen
  - Systemgenauigkeit und Kalibrierung
  - Alternative Capturesysteme
  - Mock-ups im INVERITAS-Systemdemonstrator
  - Nutzung des INVERITAS-Systemdemonstrators
  - Systemdemonstration im DFKI RIC
  - Multimodales Sensorsystem
  - Kamerasysteme und Datenvorverarbeitung
  - LIDAR
  - Nahbereichs-Relativ-GNC/REGIS
  - GNC Ground Support
  - Mehraktorsteuerung: Basissystem
  - Mehraktorsteuerung: Bildverarbeitungsmodul

Abb. 1 zeigt die endgültige zeitliche Planung der Arbeitspakete.

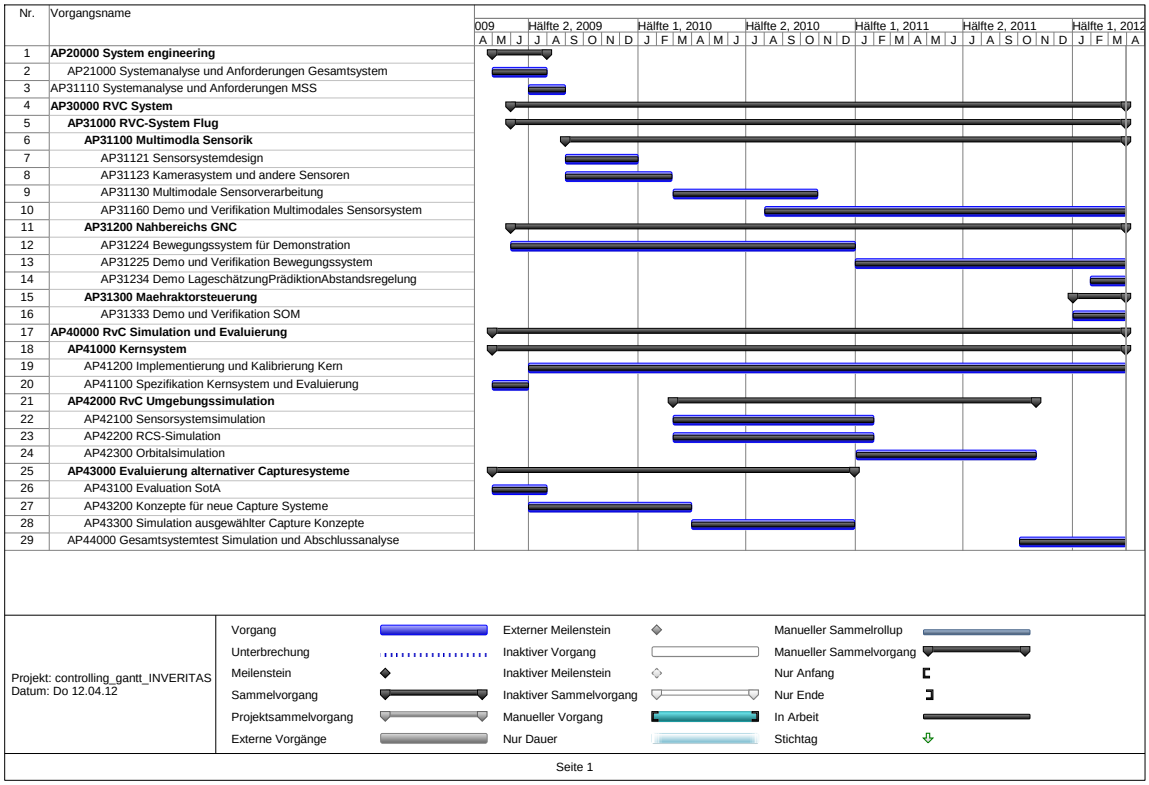


Abbildung 1: *Gantt Ablaufplan*

2.4 Wissenschaftlicher und technischer Stand zu Beginn des Vorhabens

2.4.1 Stand der Technik zu Beginn des Vorhabens

Es existieren zahlreiche Entwicklungen und Kenntnisse auf dem Gebiet der autonomen Robotersysteme in Deutschland. In dem hier relevanten Zusammenhang sollen nur die für den Bereich Raumfahrt zum Zeitpunkt der Antragstellung relevanten Dinge zum Stand der Technik diskutiert werden. Robotik in der Raumfahrt unterscheidet sich insofern von anderen Anwendungen als man es dabei zum Teil mit sehr spezifischen Randbedingungen zu tun hat, wie beispielsweise keine oder geringere Schwerkraft, extreme Anforderungen bei Masse und Energiebegrenzungen, thermale Bedingungen, kleine Bauräume und alles bei möglichst hoher Leistungsfähigkeit. Die Systeme können in aller Regel nicht gewartet werden und die Bedienung erfolgt über sehr große Strecken, welches Zeitverzögerungen hervorruft. Die Kommunikation unterliegt zum Teil zeitlich und bezüglich der Bandbreite starken Eingrenzungen. Im nationalen Bereich waren zum Zeitpunkt der Antragstellung, durch das DLR initiiert, die letzten ca. 10 Jahre von der Entwicklung von raumfahrttauglichen Manipulatorsystemen inklusive der notwendigen Steuerungs- und Sensorsysteme geprägt. Hintergrund der Entwicklungen waren die Anwendungsschwerpunkte Satellite Servicing und Astronautensupport. Hinzugekommen ist in den letzten Jahren vor der Antragstellung der Bereich Exploration von Himmelskörpern mit einem Fokus auf die Exploration der Oberflächen. Hierzu fanden zahlreiche Entwicklungen in DLR Vorhaben wie ROKVISS, TECSAS, VITAL, ESS-T, GETEX, ARAMMIS, LUNARES statt, manchmal unterstützt von begleitenden regionalen Vorhaben zur Technologie- und Komponentenentwicklung. In den letzten Jahren vor der Antragstellung hat darüber hinaus die Entwicklung von mobilen Robotersystemen an Bedeutung gewonnen, bedingt durch das wieder steigende Interesse an der Planetenexploration. Für den Bereich RvC Systeme war zum Zeitpunkt der Antragstellung die im Rahmen des DLR Vorhabens TECSAS Phase B vorgenommene Analyse immer noch gültig, in der insbesondere der Stand der Technik in Deutschland untersucht wurde. Zusammengefasst kann die Situation wie folgt dargestellt werden:

- Es existieren zahlreiche Vorarbeiten in vielen Bereichen
- Es existieren keine RvC Systeme / Demonstratoren als Gesamtsysteme
- Die wesentlichen Defizite bezüglich einzelner Technologiebereiche werden gesehen in
  - Sensorik
  - Perception / Sensordatenauswerung
  - GNC und Regelungsverfahren für (Kontakt-) Phasenübergänge
  - Steuerung komplexer Systeme
  - Kommunikation bei Telepräsenz

Stand der Technik im Bereich Manipulatorentwicklung war zum Zeitpunkt der Antragstellung, dass modulare leistungsfähige Gelenkmodule zur Verfügung stehen, deren

Leistungsfähigkeit bereits im Weltraum erfolgreich verifiziert wurde. Hier sind insbesondere die Arbeiten des Institutes für Robotik und Mechatronik in Oberpfaffenhofen zu nennen, denen weltweit eine führende Stellung zuerkannt wird, wobei das Institut durch die Firma Kayser-Threde bei der Raumfahrtqualifikation ihrer Komponenten intensiv unterstützt wurde. Auch Unternehmen wie Astrium verfügen hier über zahlreiche Entwicklungen und blicken in diesem Bereich auf langjährige Erfahrungen zurück.

Für ganze Manipulatoren und die für einen sinnvollen Einsatz dazugehörigen Steuerungs- und Sensorikelemente galt der genannte Status der Flugqualifikation noch nicht. Im Bereich Steuerungssysteme gab es eine große Vielfalt unterschiedlichster Entwicklungen begründet durch die Tatsache, dass die meisten in diesem Bereich tätigen Unternehmen und Institute über die Jahre hinweg eigene leistungsfähige Systeme entwickelt haben. Mit praktisch allen Systemen war es im Endeffekt möglich, auch durchaus komplexe Robotersysteme zu steuern. Unterschiede bestanden in den Architekturprinzipien und den Entwicklungsschwerpunkten (z.B. Systemsteuerung, Subsystemsteuerung, Sensordatenverarbeitung, Autonomieprinzipien, Verfahrensentwicklung zu Einzelthemen [z.B. Navigation, Bewegung, Erkennung, Kommunikation usw.]) je nach Ausrichtung. Als Repräsentanten können in diesem Bereich eingegrenzt auf raumfahrtrelevante Systeme DLR-RM, Astrium, Uni-Dortnund, Uni-Bremen (DFKI), Fraunhofer Berlin genannt werden. Das einzige in Deutschland entwickelte und zur Zeit der Antragstellung genutzte und im Weltraum getestete System war MARCO, das auf eine Entwicklung von Astrium und DLR-RM zurückgeht und im Vorhaben ROKVISS auf der ISS genutzt wurde. Entwicklungsdefizite bestanden noch bei Systemen, mit denen sehr komplexe Systeme (z.B. eines oder mehrer mobile System(e) plus Roboterarm(e) plus komplexe Sensorsysteme plus komplexe Kommunikationsanforderungen) kontrolliert werden. In Deutschland waren zum Zeitpunkt der Antragstellung eine große Zahl Sensorsysteme und auch Softwaresysteme zur Datenauswertung verfügbar. Raumfahrtqualifizierte Kamerasysteme werden in Deutschland hauptsächlich von DLR in Berlin und der Firma VHS, häufig auch beiden gemeinsam, entwickelt. Für laserbasierte optische Sensoren hat die Firma Jena-Optronik eine internationale Führungsposition. Im Bereich Kraft-Momentensensorik sind die an die Manipulatorentwicklung angegliederten Arbeiten von DLR-RM maßgebend. Ganz wesentliches Ziel für die Sensorik und Sensordatenverarbeitung ist die Beschaffung räumlicher Daten sowie Analyse dieser Daten bezüglich der Erkennung von Objekten und deren Lage im Raum bzw. zu anderen Objekten. Für Softwaresysteme zur Sensordatenauswertung im Robotikbereich, die sich hauptsächlich auf die Bildverarbeitung beziehen (kamera- und laserbasiert) und den Stand der Technik darstellen, kann man DLR-RM, Astrium, Jena-Optronik und in Randbereichen VHS als Kompetenzen nennen. Dass es für den Bereich Sensorik/Bildverarbeitung einen ganzen Industriezweig außerhalb der Raumfahrt gibt, sei an dieser Stelle nur erwähnt. Der Stand der Technik zum Zeitpunkt der Antragstellung im Bereich GNC für Rendez-Vous Aufgaben wird im Kern durch die Erfolge von ATV beschrieben, dessen GNC Elemente von Astrium entwickelt und implementiert wurden. Hiermit kann Rendez-Vous und Docking bei größeren Systemen mit festen garantierten Randbedingungen erfolgreich gelöst werden. Ein Rendez-Vous und Capture von unkooperativen Zielen war mit solchen Systemen nicht möglich, da es allen Teilsystemen an Flexibilität und Autonomie fehlte, um eine Vielzahl

unterschiedlicher Situationen sensorisch, navigatorisch, regelungstechnisch und systemsteuerungsseitig abdecken zu können. In den einzelnen genannten Teilbereichen sind hierzu in der Vergangenheit in Deutschland Technologievorhaben (z.B. ESS-T) durchgeführt worden, die jedoch größtenteils mehr als 10 Jahre vor der Antragstellung zurückliegen und nicht mehr dem aktuellen Stand der Technik bzw. den derzeitigen Möglichkeiten entsprachen und auch die zum Zeitpunkt der Antragstellung in der Planung befindlichen Anwendungen nicht ausreichend berücksichtigten. Ein wesentlicher Unterschied zwischen den zum Zeitpunkt der Antragstellung bestehenden RVD Systemen und den angestrebten RvC Systemen ist in der Tatsache zu sehen, dass bei RvC mehrere mit einem Antrieb versehene Elemente (z.B. Bus des Servicers und Roboter auf dem Bus) koordiniert werden müssen und sich gegenseitig beeinflussen. Dieses, insbesondere bei unkooperativen Zielen, vielfältige Problem war bisher noch kaum betrachtet und bearbeitet worden. International gab es verschiedene Entwicklungen auf diesem Gebiet. Die größten Fortschritte markierten die Entwicklungen im Rahmen des amerikanischen Orbital Express Vorhabens in dem automatisiertes robotisches Rendez-Vous und Capture eines kooperativen Ziels mehrfach mit unterschiedlichen Randbedingungen in-orbit demonstriert wurde. Im Bereich der A&R Bodensteuerung spiegelten für den Raumfahrtbereich in Deutschland die ROKVISS Arbeiten und Ergebnisse den zum Zeitpunkt der Antragstellung aktuellen Stand am besten wieder. Hier wurden Teleoperation und Telepräsenzverfahren in einer einfachen Demonstrationsumgebung in-orbit getestet. Diese Elemente galt es weiterzuführen und zu erweitern, um sie auch bei komplexeren RvC-Systemen einsetzen zu können. Hierbei war insbesondere die notwendige Verbindung und Koordination zwischen durch Teleoperation oder Telepräsenz durchgeführten Sequenzen und autonom durchgeführten Sequenzen von immenser Bedeutung und bedurfte der weiteren Technologieentwicklung. Hier sind besonders Verfahren für schnelles Replanning zu nennen, wenn neue ungeplante Situationen entstehen und in naher Umgebung des Zieles schnell reagiert werden muss, um Schäden zu vermeiden und die Mission erfolgreich weiterführen zu können. Dies gilt sowohl für Teleoperation oder Telepräsenz (TO/TP) Sequenzen als auch für autonom durchgeführte Sequenzen.

2.4.2 Fachliteratur und verwendete Dokumentationsdienste

Für die Entwicklungen im Rahmen von INVERITAS am DFKI-RIC wurden neben den in Abschnitt 3 angegebenen Literaturreferenzen zusätzlich die im Folgenden referenzierten Quellen herangezogen: [15], [19], [2], [12], [16], [11], [3], [22].

Das vollständige Literaturverzeichnis ist am Ende dieses Dokuments ab Seite 137 zu finden.

2.5 Zusammenarbeit mit anderen Stellen

Generell war es für das DFKI im Projekt INVERITAS erforderlich, eng mit dem Projektpartner und Konsortialführer Astrium zusammenzuarbeiten. Insbesondere die Softwarekomponenten von Astrium für das Simulationssystem und die Satellitensteuerung

mussten in die Simulations-Kernkomponenten des DFKI auf der Ebene von Binärdateien integriert werden. Hierzu wurde ein entsprechendes Framework mit passenden Schnittstellen entwickelt.

Hardwareseitig waren exakte Schnittstellen zwischen dem Langestreckenbewegungssystem und den Mockups von Astrium sowie deren On-Board Hardware erforderlich.

Diese Zusammenarbeit hat sehr gut funktioniert und das INVERITAS-Simulationssystem konnte seine Modularität behalten, so dass es auch in Zukunft gut auf verschiedene neue Aufgabenstellungen angepasst werden kann.

Für die Entwicklung einiger Algorithmen die für die automatische Vermeidung von Kollisionen zwischen den bewegten Komponenten des Langstreckenbewegungssimulationssystems herangezogen wurde gab es einen Informationsaustausch mit dem DFKI-Forschungsbereich "Cyber-Physical Systems".

3 Eingehende Darstellung

3.1 Erzielte Ergebnisse

3.1.1 AP10000-D: Management

Im Projektzeitraum hat das DFKI die Management-Bereiche Coordination & Risk-Management, Controlling und Contracts für die Arbeitspakete mit DFKI-Anteilen durchgeführt. Um Verzögerungen im Projektverlauf grade im Hinblick auf die Präzision der Anlage abzufangen, wurde eine kostenneutrale Verlängerung des Projekts bis zum 31.3.2012 beantragt und bewilligt.

3.1.2 AP21000-D: Systemanalyse und Anforderungen Gesamtsystem

Die Arbeiten an diesem Arbeitspaket haben planmäßig begonnen. In einem Arbeitstreffen mit den Projektpartnern wurden die Randparameter bezüglich Bewegungsbereich, Stromversorgung, Zuladung und Geschwindigkeit des Langstreckenbewegungssystems abgesprochen.

Mit der Firma Spidercam wurden alle wichtigen Vorarbeiten hinsichtlich der Auftragserteilung durchgeführt. Dies beinhaltete einen einwöchigen Aufenthalt in Österreich vor Ort, wo Vorversuche an einem Demonstrationssystem durchgeführt wurden (siehe Abbildung 2). Der entsprechende Betrag zur Anschaffung des Langstrecken-Bewegungsystems wurde nach Abgabe des Angebots von SpiderCam entsperrt und damit die Bestellung auf den Weg gebracht.

Die Nutzlast des Spider-Cam-Systems, der Servicer, wurde bereits grob skizziert. Die Kernkomponenten, also das Dockingsystem und das multimodale Sensorsystem sowie deren Steuerungseinheiten dürfen die Gesamttraglast von 150 kg nicht überschreiten. Für das Multimodale Sensorsystem wurde eine Systemarchitektur entworfen, die eine Reihe von PC-Systemen vorsieht, die jeweils die Vorverarbeitung der an sie angeschlossenen Kameras übernehmen. Die derart aufbereiteten Kameradaten können dann in ähnlich aufgebauten Systemen weiter verarbeitet und mit anderen Sensordaten kombiniert werden.

Im Gespräch mit EADS ASTRIUM wurde auch die Position und der Bewegungsumfang des KUKA KR-60 zur Simulation des Klienten abgesprochen.

Ein während der Konzeption sehr kritisch bedachter Punkt war die Energieversorgung auf dem Flugsystem. Die ursprünglich geplanten 400W bei 48VDV waren für den Betrieb der Systeme auf der Plattform nicht ausreichend genug. Bei SpiderCam wurde, in Kooperation mit dem TÜV, die Möglichkeit erörtert bis zu 1KW bei 230VAC auf der Plattform zur Verfügung zu stellen. Diese Änderung konnte im dritten Quartal 2009 übernommen werden.

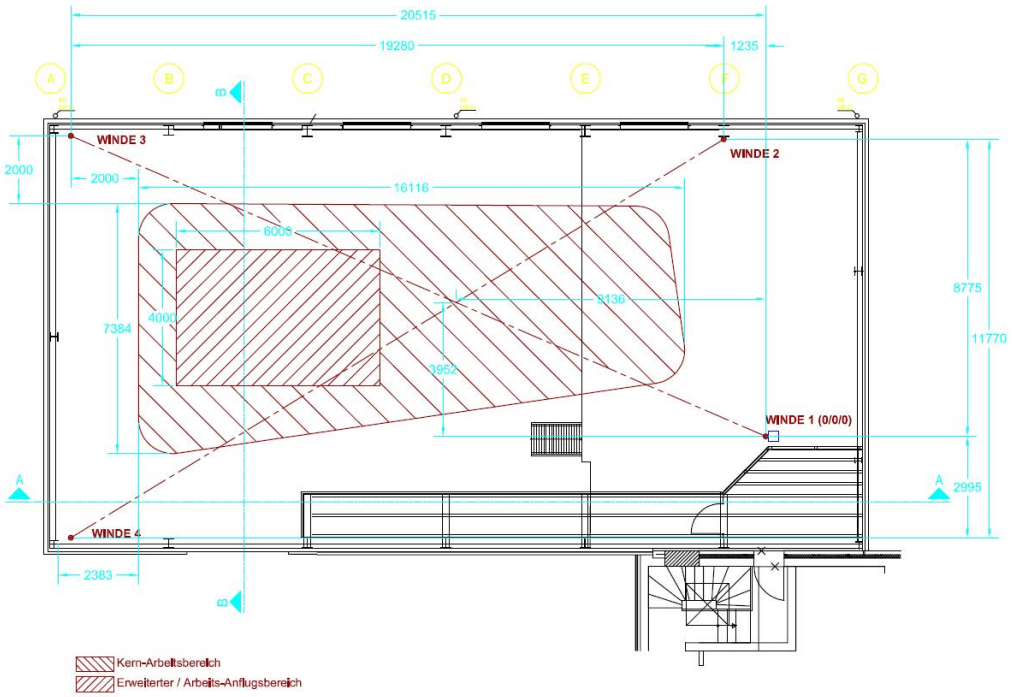


Abbildung 2: *Die Arbeitsbereiche des Langstrecken-Bewegungssystems nach den Vorversuchen.*

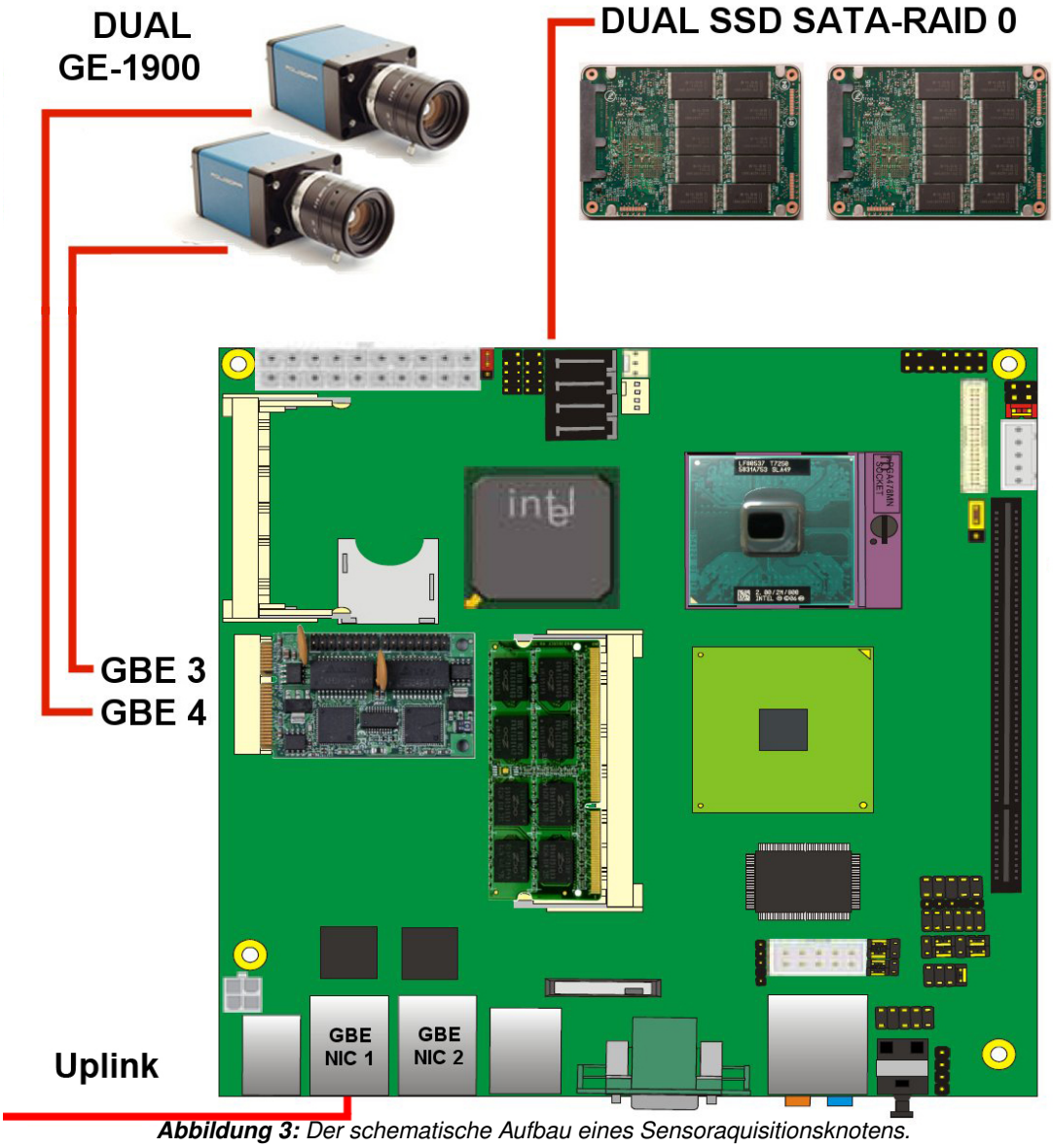


Abbildung 3: *Der schematische Aufbau eines Sensoraquisitionsknotens.*

3.1.3 AP31110-D: Systemanalyse und Anforderungen MSS

Mit dem Projektpartner EADS wurden die Anforderungen an das Multimodale Sensorsystem in mehreren Meetings besprochen. Dies betraf von Seiten des DFKIs insbesondere die Kamerasysteme. Ausgehend von den Anforderungen der DEOS Phase A Studie wurden verschiedene Arten und Anzahl an Stereokamerasystemen erörtert. Da im INVERITAS Projekt nur die letzten 15 Meter der Annäherung von Interesse sind, wurden die Anforderungen an ein Kamerasystem wie folgt definiert:

- Mindestens 2cm/Pixel bei maximaler Reichweite
- Mindestens 90% Überdeckung im Nahbereich
- Dynamische Auflösungsveränderung durch Binning zur Simulation von Space-Qualifizierbaren Kameras

Relevante Optimierungsgrößen waren Interface, Auflösung, Bildrate, Farbe/Monochrom, Anbindungsbandbreite der Kamera, Größe des Sensors, Lichtstärke, Sensorqualität und Öffnungswinkel des Objektives. Für das Stereosystem kommen Basislinie, Arbeitsabstand, Überlappungsfaktor und Minimalauflösung der Stereokorrespondenz hinzu.

Als Interfaces für das Kamerasystem kamen GigE, FireWire und USB 2.0 in die nähere Auswahl. Das CameraLink-System wurde aufgrund von zusätzlichem Hardware-Overhead nicht weiter betrachtet. Die anderen drei Interfaces unterscheiden sich in Datenraten sowie in Bezug auf Kabellängen. Das GigE Interface besitzt den Vorteil, netzwerktransparent zu sein, so dass darüber angebundene Kameras innerhalb eines LAN von verschiedenen Geräten benutzt werden können. Dies hat den Vorteil, dass auch Kamerabilder, auf denen hauptsächlich Bildverarbeitungsalgorithmen laufen sollen, zur Kontrolle oder für Darstellungsaufgaben menschlichen Nutzern dargestellt werden können ohne einen Software-Overhead zu verursachen. Aufgrund dieses Vorteils und der hohen Bandbreite des Interfaces (bis zu 125 MB/s) wurde GigE als Interface für die Systemkameras gewählt. Die eng zusammengehörigen Parameter Auflösung, Bildrate und Farbe/Monochrom müssen gemeinsam betrachtet werden. Ein oberer limitierender Faktor ist die Anbindungsbandbreite der Kamera, also bei GigE 125 MB/s. Es gilt also $res_X \cdot res_Y \cdot FPS \cdot colors \cdot 10^{-6} \leq 125$. Die Wahl ob Farbe oder Monochrom wird hauptsächlich durch die Hauptverwendung bestimmt. Operatorenkameras sollten wenn möglich Farbkameras sein, da so eine bessere Überwachung und Wiedererkennung möglich ist. Für Bildverarbeitung sind meistens monochrome Kameras aufgrund ihrer besseren Lichtstärke von Vorteil. Die Bildrate sollte so hoch wie möglich sein, mindestens aber 25 Bilder pro Sekunde (FPS). So empfindet ein menschlicher Betrachter die Bildsequenz als flüssig. Zur Simulation von niedrigeren Bandbreiten oder Verarbeitungsleistungen sollte die Bildrate aber reduziert werden können. Bei der Auflösung ist eine hohe Auflösung ebenfalls nur wünschenswert, wenn sie durch Einstellungen auf der Kamera reduziert werden kann, da Bildverarbeitung im Allgemeinen nur auf kleinen Auflösungen realisierbar ist. Solange die Reduktion per Software möglich ist, bietet eine Kamera mit hoher Auflösung nur Vorteile. Anhand der genannten Ãœberlegungen wurden Kameras mit eine Auflösung von 1920 x 1080 (Full-HD) mit einer Bildrate

von 30 FPS gewählt. Die Auflösung kann durch "Binning"(Zusammenfassen von Pixeln) oder "Region of Interest"(Wahl des relevanten Sub-Bildes) reduziert werden, die Bildrate kann frei gewählt werden. Für das Stereokamerasystem wurden monochrome Kameras, für die Szenenkamera eine Farbkamera gewählt. Die Größe des Sensors, seine Lichtstärke und Qualität sind herstellerspezifisch. Generell bedeutet ein großer Sensor eine hohe Lichtstärke aber geringere Qualität (Signal/Rauschverhältnis). Eine Untersuchung hat gezeigt, dass die Sensoren der Firma Kodak hier sehr gut abschneiden, und auch bei großen Sensorflächen exzellente Signal/Rauschverhältnisse aufweisen. So fiel die Wahl insgesamt auf Kameras der Firma Prosilica, die Modellreihe GE und der Typ 1900 bzw. 1900C. Sie besitzen den 1"Kodak KAI-2093 Sensor und einen C-Mount Objektivadapter. Ihre geringe Baugröße, die vielseitigen Einstellungsmöglichkeiten und der geringe Stromverbrauch von nur 5W zeichnen diese Kamera aus.

Das Stereokamerasystem wurde mit Objektiven mit einem $60°$ Öffnungswinkel ausgestattet. So ergibt sich bei einer Basislinie von 30 cm im Arbeitsbereich (1-17m) minimal eine Auflösung von einem Pixel pro cm, bei einem minimalen Überlappungsfaktor von 74%. Die dritte Kamera wurde mit einem $90°$-Objektiv ausgestattet, und dient als Szenenkamera.

3.1.4 AP31121-D: Sensorsystemdesign

Die Arbeiten in diesem Paket beschäftigen sich mit der Auslegung und der Beschaffung des Stereokamerasystems Anhand der Spezifikationen aus AP311110.

Zusammen mit EADS ASTRIUM wurde ein modulares Sensorbus-Konzept entwickelt, das auf einer sensornahen Vorverarbeitung basiert. Der Kerngedanke hinter diesem System ist es eine Austauschbarkeit der Sensoren und ihrer Verarbeitung zu erreichen. In INVERITAS ist insbesondere die Space-Qualification nicht von primären Interesse, es soll aber gezeigt werden können, dass dies prinzipiell möglich ist.

Das Sensorsystem ist spezifiziert. Es wurde eine verteilte Architektur mit mehreren Sensoraquisitionsknoten gewählt, die üer Gigabit Ethernet-Schnittstellen miteinander kommunizieren. Als Sensoraquisitionsknoten wurden Hochleistungs-Embedded-PCs gewählt, die genug Rechenleistung bieten, um eine Vorverarbeitung der Sensorsignale bereits verteilt durchzufüren. Ein Beispiel für eine solche Vorverarbeitung ist die Stereokorrespondenzberechnung eines Stereokamerasystems. Durch diese Architektur wird die benötigte Rechenleistung auf viele Knoten verteilt, was aufgrund der begrenzten Rechenleistung von Weltraumqualifizierter Rechenhardware dringend notwendig ist. Eine schematische Darstellung eines Knotens ist in Abbildung 4 dargestellt. Aufgrund von Lieferschwierigkeiten einiger Komponenten konnte die Spezifikation im vierten Quartal 2009 noch nicht abgeschlossen werden, da vor dem Abschluss der Spezifikation ein System aus zwei Sensorknoten exemplarisch getestet werden sollte.

Nach verspäteter Lieferung wurde der zweite Sensorknoten im ersten Quartal 2010 aufgebaut und in Betrieb genommen. Ein kooperierendes System aus zwei Sensorknoten erfolgreich getestet, und damit die Eignung des Systemdesigns nachgewiesen. Mit Abschluss dieser nachträglichen Tests war dieses Arbeitspaket beendet.

3.1.5 AP31123: Kamerasysteme und andere Sensoren

Die Aufgabe des DFKIs in diesem Arbeitspaket konzentriert sich auf die Beschaffung der Kameras und Bildverarbeitungssysteme. Anhand der in AP31110 aufgestellten Anforderungen wurde folgendes Kamerasystem ausgewählt:

Zwei Gigabit-Ethernet-Kameras (Prosilica GE1900):

- Sehr lichtempfindliches 1 Zoll CCD
- Hohe Auflösung: 1920x1080 (HD-TV) bei 30 FPS
- 12.5mm Brennweite, 54° Öffnungswinkel
- Dynamische Auflösungsanpassung durch Binning / ROI
- Gesamtgewicht mit Objektiv 306g

Bei einem Kamerabstand von $15cm$ ergibt sich bei $15m$ Entfernung eine Auflösung von $1.3cm$ pro Pixel und eine Überdeckung von 91% bei $1m$ Entfernung. Dies entspricht genau der Spezifikation des Systems.

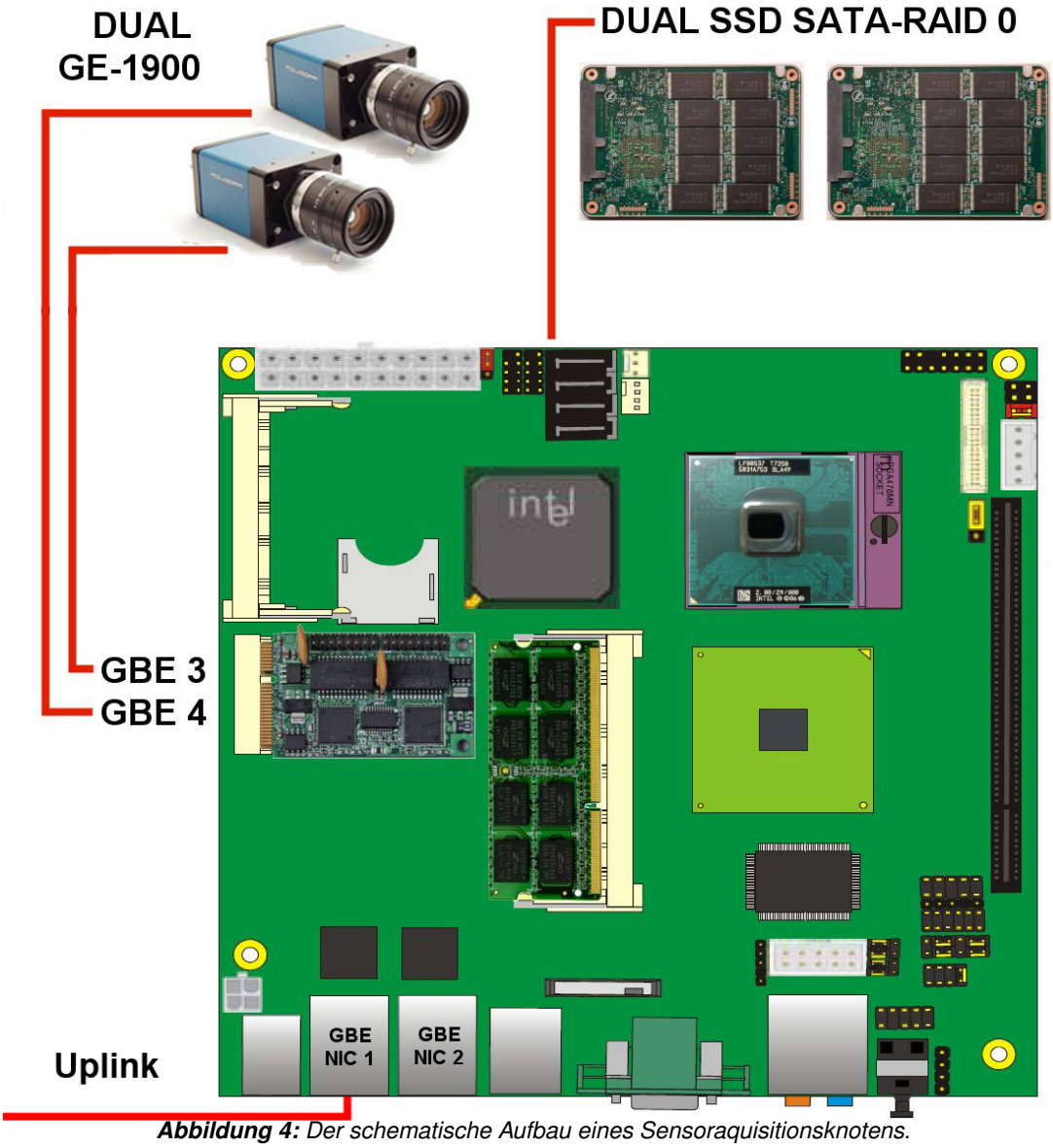


Abbildung 4: *Der schematische Aufbau eines Sensoraquisitionsknotens.*

Zur Bildverarbeitung wurde in Abstimmung mit EADS ein Quadcore-Embeded-System ausgewählt, an das beide Kameras direkt angeschlossen werden und das über einen weiteren Netzwerkanschluss die Daten auf den Sensorbus gibt. Der Vorteil dieses Aufbaus ist, dass auf dem PC verschiedene Auswertungsmethoden getestet werden können. Z.B. ist es auch möglich, ein FPGA basiertes Stereosystem zu simulieren und damit unter anderem die Weltraumtauglichkeit des Gesamtsystems zu demonstrieren.

Die Kamerasysteme sowie ihre Anbindung an das DHS wurden im vierten Quartal 2009 realisiert. Es wurden wie oben erläutert hochauflösende Kameras der Firma Prosilica ausgewählt. Diese zeichnen sich durch gerine Integrationsgröße, sehr hohe Bildqualität und breitbandige GiG-E-Schnittstelle aus. Insbesondere der letzte Punkt erleichterte die Anbindung an das DHS enorm. Es wurden drei Kameras vom Typ GE1900 akquiriert, welche eine Full-HD-Auflösung (1920x1080) bei Video-Bildraten (31 FPS) besitzen. Durch ihre großen 1"CCD Chips sind sie enorm lichtempfindlich und weisen ein sehr gutes Signal-Rauschverhältnis auf. Durch Reduktion der Auflösung (entweder durch Binning, also das Zusammenfassen von Bildpunkten, oder durch Auswahl einer Interessensregion (ROI)) kann die Bildrate bei Bedarf noch deutlich erhöht werden. Von den drei Kameras wurden zwei für eine Stereokonfiguration vorgesehen. Der Vorteil gegenüber einem fertig konfektionierten Stereokamerasystem ist die höhere Flexibilität bei der Wahl der Stereokonfiguration (Basislinie, Schielwinkel). Da auch ein kommerzielles Stereokamerasystem für den Anwendungsfall einer präzisen photometrischen Kalibrierung unterzogen werden müsste, ist an dieser Stelle kein Nachhteil der gewählten Lösung erkennbar.

Aufgrund von Änderungen am Projektfokus wurde das Kamerasystem für den Greifer nicht ausgelegt.

Zur Kalibrierung der Kamerasysteme wurde das "Camera Calibration Toolkit"für Matlab benutzt. Dieses bietet eine einfache Kalibrierungsprozedur mit guter Kalibrierungsqualität. Danach sollte geprüft werden, ob die Qualität der Kalibrierung ausreicht, oder auf kommerzielle Systeme z.B. der Fa. AICON zurückgegriffen werden muss.

Im Bezug auf weitere Sensorik wurde geprüft, ob die Verwendung eines Laserscanners der Fa. Hokuyo sinnvoll ist. Während diese Sensoren eine sehr hohe Genauigkeit bieten sind sie durch ihre sequentielle Erfassungsmodalität schwierig auf ständig bewegten Systemen einzusetzen. Die Entscheidung über diesen Punkt wurde verschoben.
Zu Beginn des Jahres 2010 musste in diesem Arbeitspaket im Bezug auf weitere Sensorik (neben den Kamerasystemen) noch auf Rückmeldung von Astrium gewartet werden. Hauptkriterium, ob weitere Sensoren notwendig sein würden, war die Eignung des LIDAR-Systems von Jena Optronik für die Experimentalzwecke in der Halle. Es bestand hier die Befürchtung, dass dieses System auf die relativ kurzen Distanzen nicht sinnvoll einsetzbar ist. Dies konnte erst genauer feststellbar sein, sobald das LIDAR-System von Astrium ausführlich getestet worden war. Aufgrund von verschiedenen Defekten an dem System war dies anfangs noch nicht möglich gewesen. Da die Zeit für den Einbau weiterer Sensoren mittlerweile zu knapp geworden war, wurde entschieden, keine weiteren Sensoren einzubauen.

Zusätzlich zu dem bereits vorhandenen Stereokamerasystem wurde auch das Über-

sichtskamerasystem ausgelegt. Es handelt sich um eine Prosilica GX 3300C-Kamera mit einer Auflösung von acht Megapixeln bei einer maximalen Bildrate von 17 Bildern pro Sekunde. Sie ist mit einem Fisheye-Objektiv ausgerüstet, welches einen großen Blickwinkel ermöglicht und so die gleichzeitige Beobachtung des Servicers und des Clients erlaubt. Da die Kamera die gleiche Schnittstelle wie die Kameras des Stereokamerasystems verwendet, ist die softwareseitige Anbindung kein Mehraufwand.

3.1.6 AP31130-D: Multimodale Sensordatenvorverarbeitung

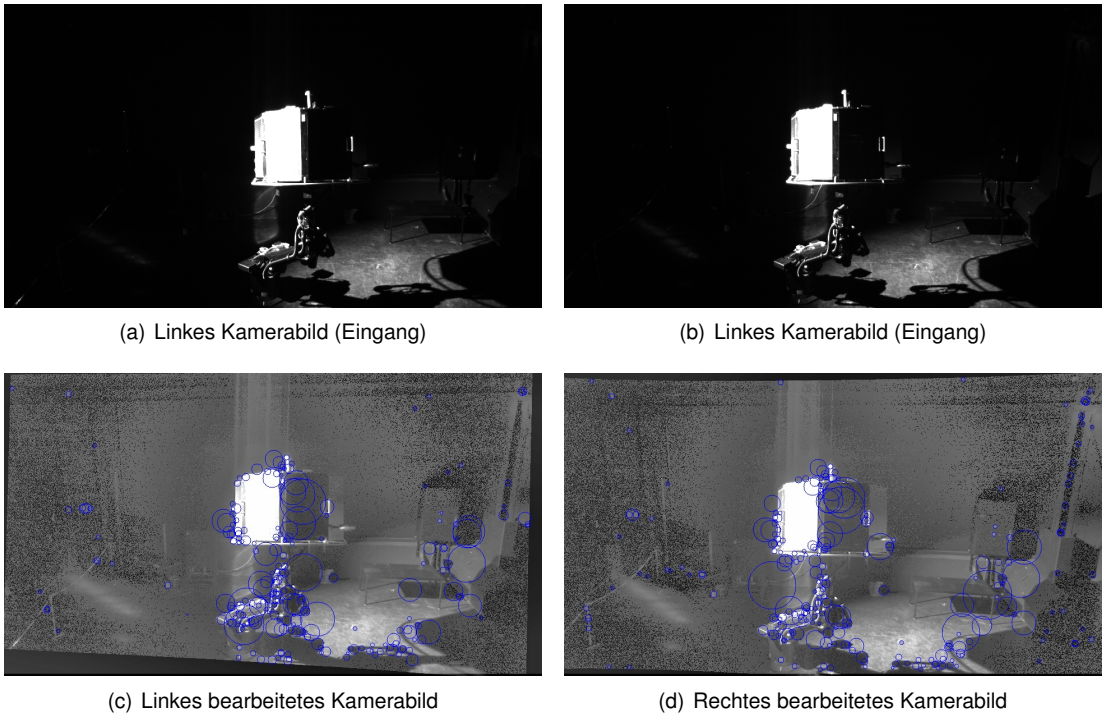
Der Fokus in diesem Arbeitspaket liegt auf dem Stereokamerasystem. Für die Bilddaten der beiden Kameras wurde während des ersten Quartals 2010 ein Modul entwickelt, welches mit hoher Effizienz eine Entzerrung und Rektifizierung auf Basis der Kalibrierungsdaten durchführt. Auf derartig vorverarbeiteten Kamerabildern lassen sich Stereoalgorithmen deutlich vereinfacht ausführen. Dies wurde dann demonstriert, indem zuerst mit einem SURF-Detektor markante Punkte auf beiden Kamerabildern identifiziert wurden (Zeitaufwand 150 ms für beide Kameras zusammen), und diese dann unter Berücksichtigung der Epipolargeometrie eines rektifizierten Stereopaares miteinander in Korrespondenz gebracht wurden. Da der letztgenannte Schritt durch die Vorverarbeitung deutlich vereinfacht wurde, konnte so eine Laufzeit von unter 200 ms (gemessen auf dem Sensorvorverarbeitungsknoten) für die Extraktion einer 3D-Punktwolke aus den Kameradaten erzielt werden. Diese Daten können dann zusammen mit anderen Sensordaten zusammengeführt werden. Die damit vorliegenden Verarbeitungsalgorithmen wurden im folgenden AP31160-D getestet. Diese Tests waren nun möglich, seitdem das Client-Mockup verfügbar ist, das als wichtiges Ziel der Sensordatenverarbeitung getestet werden sollte.

3.1.7 AP31160-D: Demo und Verifikation Multimodales Sensorsystem

Nach Anlieferung des Satelliten-Mockups des Klienten wurden mit dem Bewegungssystem ein Teilstück einer Satellitenannäherung simuliert und dabei Kamerabilder mit dem Stereokamerasystem aufgenommen (siehe Abb. 5(a) und Abb. 5(b)). Mit den gewonnenen Bilddaten wurden die entwickelten Algorithmen getestet (siehe Abb. 5(c) und Abb. 5(d)). Dabei wurde ersichtlich, dass die Kamerakalibrierung eine sehr wichtige Rolle einnimmt. Eine ungenügende Kalibrierung führt dazu, dass nach der Rektifizierung die selben gefundenen Merkmale in beiden Bildern nicht die gleiche y-Koordinate besitzen. Damit zeigten die nachfolgenden Algorithmen noch nicht den gewünschten Effekt. Außerdem wurde ersichtlich, dass neben dem Satelliten-Mockup auch Boden, KUKA Roboter und Hallenwand zu sehen sind. Hier waren weitere Experimente notwendig, die zum einen ein besseres Einstellen der Parameter ermöglichen und zum anderen zum Test weiterer Verbesserungsmaßnahmen (zusätzliche Abdeckungen, LEDs, Filter, ...) dienen sollten.

Kameras

Die von uns bereitgestellten Bilder während einer Trajektorienfahrt wurden von unserem Partner Astrium ausgewertet. Da deren Algorithmen nicht zu voller Zufriedenheit funktionierten, wurde eine erneute Kamerakalibrierung durchgeführt. Danach wurden neue Bilder aufgenommen, und anhand der rektifizierten Kamerabilder die Güte der Kalibrierung untersucht. Nachdem die Kalibrierung als gut empfunden wurde, wurden die Bilder zu Astrium geschickt. Deren Algorithmen funktionieren jetzt besser, womit Astrium die bessere Kalibrierung bestätigte.



(a) Linkes Kamerabild (Eingang)

(b) Linkes Kamerabild (Eingang)

(c) Linkes bearbeitetes Kamerabild

(d) Rechtes bearbeitetes Kamerabild

Abbildung 5: *Eingangsbilder des Stereokamerasystems und bearbeitete Ausgangsbilder (Grauwertanpassung, Rektifizierung, SURF)*

Als Vorbereitung auf die Verifikation des MMS wurde das Programm zum Akquirieren und Vorverarbeiten von Kamerabildern mit einer Schnittstelle erweitert. Über diese können zum einen direkt die Eingabeparameter der Kameras geändert und zum anderen die Vorverarbeitungsschritte selektiv ausgewählt werden. Dabei wurde sich an die vorgegebene Schnittstellenspezifikation (MMS ICD) gehalten. Zusätzlich wurde ein kleines Programm mit GUI entwickelt, um alle Funktionen zu testen. Im Anschluss konnte das MMS dann in Zusammenarbeit mit Astrium getestet werden.

Es wurde erfolgreich die Schnittstelle zwischen der Low-Level Bildverarbeitung vom DFKI und der High-Level Bildverarbeitung von Astrium getestet. Obwohl der Datentransfer funktioniert, wurde dieses AP zu dem Zeitpunkt noch nicht geschlossen, da weitere Tests mit Astrium anstanden. Dabei sollten vor allem die verwendeten Algorithmen auf ihre Tauglichkeit untersucht werden. Die Testergebnisse sollten Aufschluss darüber geben, ob noch Änderungen im multimodalen Sensorsystem (MMS) durchzuführen waren.

Laserschutz

Für die Demo und Verifikation des MMS wurde nun auch der im Servicer integrierte LIDAR genutzt. Dieser verwendet einen Laser, der infrarotes Licht (1550 nm) mit

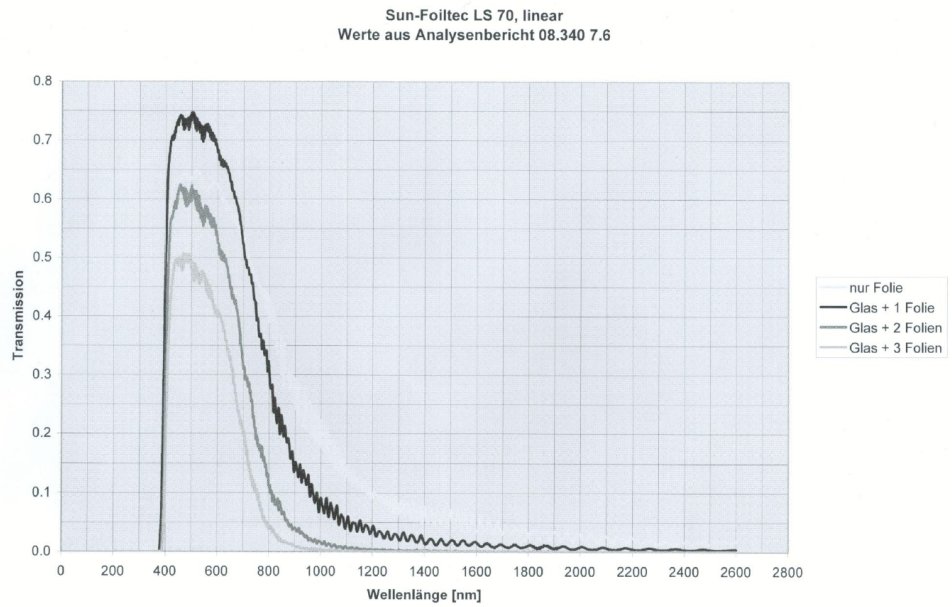


Abbildung 6: *Spektralanalyse der LS70 Laserschutzfolie*

ca. 38 mW aussendet und somit in Laserklasse 3b fällt[1]. Damit ist er gefährlich für das Auge. Aus diesem Grund wurden die Fenster des RIMRES und des INVERITAS Leitstands mit Laserschutzfolie beklebt. Diese lässt im angegeben Wellenlängenbereich ca. $\frac{1}{100}$ der Leistung durch (siehe Abb. 6)[2], wodurch eine Gefährdung für das Auge auszuschließen ist. Neben den Schutz der beiden Leitstände durch das Bekleben mit Laserschutzfolie ist es notwendig, den Gefahrenbereich vor unbefugtem Zutritt zu schützen um gesundheitliche Schäden zu vermeiden. Zu diesem Zwecke werden die beiden Außentüren der Weltraumexplorationshalle zukünftig nicht mehr von Außen zu öffnen sein, so dass ein Betreten während des aktiven Lasers nicht mehr möglich ist. Für zukünftige Projekte ebenfalls in Planung ist der Einbau einer zusätzlichen Wand, welche die Explorationshalle räumlich von dem ISS-Raum trennen soll und somit auch unbefugten Zutritt aus dem weißen Raum unterbindet. Die restlichen drei Eingänge (Haupttür der Halle, Leitstand 1. Ebene und Leitstand 2. Ebene) sollen zukünftig mit Laserwarnleuchten ausgestattet werden, welche auf eine Gefahrensituation hinweisen und unbefugten Zutritt untersagen. Die Funktionalität der Laserwarnleuchten würde vom Leitstand aus überwacht werden. Vor Betrieb des Laser muss dann immer sicher gestellt werden, dass alle Türen geschlossen sind, die Laserwarnleuchten in Betrieb und sich keine Person im Gefahrenbereich befindet. Zusätzlich sind auch Laserschutzbrillen vorhanden für Fälle, in denen man sich nicht hinter der Laserschutzfolie befindet.

Fortführung der Tests

Der Interface-Test zusammen mit Astrium war erfolgreich. Der MMS Rechner von Astrium kann diverse Bilder und extrahierte Bildinformationen vom MMS Rechner des DFKI

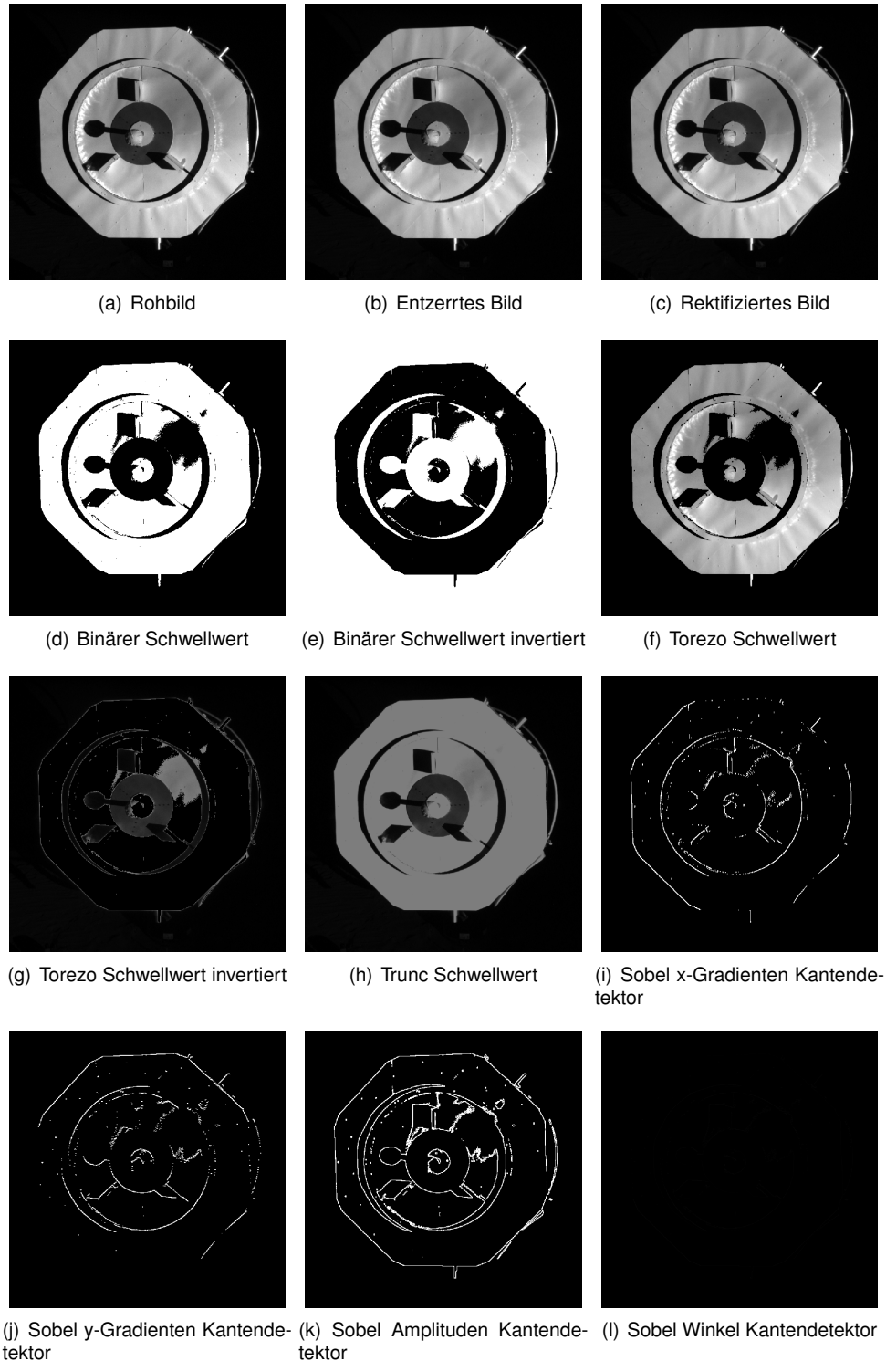
[1] entsprechend Spezifikation Servicer Laser, Laservision GmbH & Co. KG
[2] entsprechend des Datenblatts der Herstellerfirma Sun-Foiltech

empfangen und weiterverarbeiten. Während der Laufzeit kann dem DFKI Vorverarbeitungsrechner befohlen werden, welche Vorverarbeitungsschritte durchgeführt, welche vorverarbeiteten Bilder übertragen, mit welchen Parametern die Vorverarbeitungsalgorithmen durchgeführt und wie die Kameras selbst eingestellt werden sollen.

Die einfachste Vorverarbeitung besteht in der Bildakquise und dem Übermitteln der Rohbilder (Abb. 7(a)). Zusätzlich kann die Entzerrung (Abb. 7(b)) und im Falle des Stereokamerasystems die Rektifizering durchgeführt werden (Abb. 7(c)). Außerdem ist es möglich, Schwellwerte zu setzen (Abb. 7(d) bis Abb. 7(h)) oder verschiedene Kantendetektoren auszuwählen (Abb. 7(i) bis Abb. 8(a)). Die Größen für Kantendetektoren und andere Parameter sind online einstellbar, damit in Experimenten schnell die besten Einstellungen gefunden werden können. Dies ist auch für verschiedene Filter zum Glätten oder zur Rauschunterdrückung möglich (Abb. 8(b) bis Abb. 8(d)).

Die Vorverarbeitung kann auch die Detektion von Blobs, Harris Punkten und SURF Merkmalen übernehmen (Abb. 8(e) bis Abb. 8(h)). Die getrackten Merkmale werden als Listen an weiterverarbeitende Nutzer übertragen. Außerdem können direkt 3D Punkte anhand der Disparität berechnet und weitergeleitet werden (Abb. 8(i)). Die genaue Auflistung aller Möglichkeiten und Parameter lässt sich dem entsprechenden Interface Control Dokument entnehmen.

Neben dem Programm für die Vorverarbeitung und Weiterleitung der Bilder bzw. Bildinformationen wurde ein Client-Programm geschrieben welches die Daten empfangen und anzeigen lassen kann. Außerdem können neben den Vorverarbeitungsparamtern auch sämtliche Kameraparameter eingestellt werden. Dies erlaubt zum einen die Voreinstellung der Kameraparameter für unterschiedliche Lichtverhältnisse und die Überprüfung der Kamerakalibrierung anhand der SURF Merkmale auf den rektifizierten Bildern. Außerdem erlaubt das Client-Programm das Aufnehmen von Bildern und Videos.



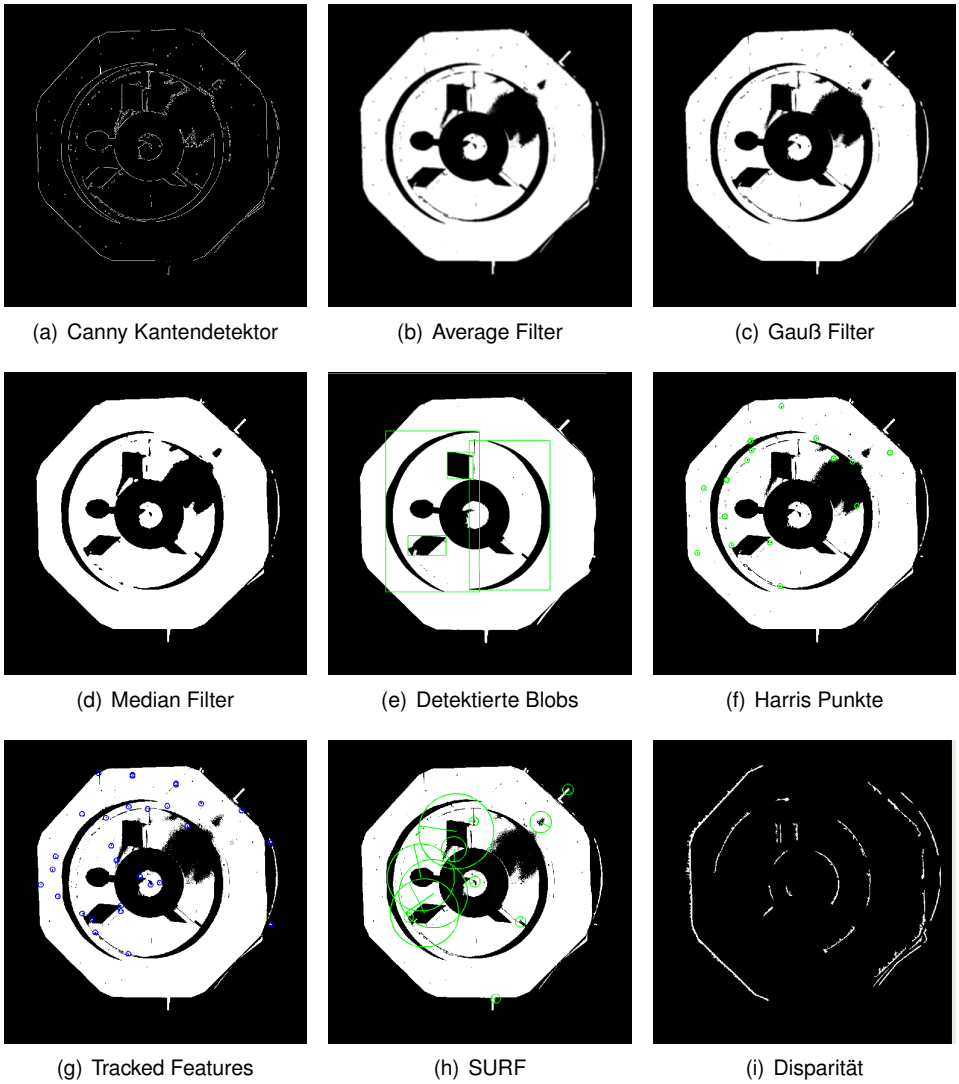
(a) Rohbild (b) Entzerrtes Bild (c) Rektifiziertes Bild

(d) Binärer Schwellwert (e) Binärer Schwellwert invertiert (f) Torezo Schwellwert

(g) Torezo Schwellwert invertiert (h) Trunc Schwellwert (i) Sobel x-Gradienten Kantendetektor

(j) Sobel y-Gradienten Kantendetektor (k) Sobel Amplituden Kantendetektor (l) Sobel Winkel Kantendetektor

Abbildung 7: *Möglichkeiten der Vorverarbeitung (1)*



(a) Canny Kantendetektor (b) Average Filter (c) Gauß Filter

(d) Median Filter (e) Detektierte Blobs (f) Harris Punkte

(g) Tracked Features (h) SURF (i) Disparität

Abbildung 8: *Möglichkeiten der Vorverarbeitung (2)*

3.1.8 AP31224-D: Bewegungssystem für Demonstrationen

Die Arbeiten in diesem Arbeitspaket beschäftigen sich mit der Auslegung und Beschaffung des in AP2100-D spezifizierten Bewegungssimulationssystems. Mit der Firma SpiderCam wurden im dritten Quartal 2009 sämtliche Spezifikationen abgesprochen und die Umsetzung beschlossen. Damit konnte die Beschaffung des Systems angestossen werden.

Die Problematik von durch die Seile induzierten Schwingungen bei ruckartigen Bewegungen wurde mit EADS ASTRIUM und SpiderCam ausführlichst diskutiert. Die Anforderungen an das System in dieser Richtung wurden in einem internen Papier festgehalten und als Abnahmebedingung festgelegt.

Mit EADS ASTRIUM wurde darüber hinaus vereinbart, dass nach der Inbetriebnahme des Systems Anfang 2010 interne Testläufe mit Parametern aus dem GnC durchgeführt werden um die genaue Leistungsfähigkeit des Systems hinsichtlich den Anforderungen in INVERITAS zu überprüfen.

Das SpiderCam und der Kuka KR-60 wurden im dritten Quartal 2009 bestellt.

Die Kernkomponenten des Bewegungsystems sind:

- Das SpiderCam System (= "CableRobot"/ "Kabelroboter") zur Simulation der Bewegungen des Chasers
- Ein Kuka KR-60 zur Simulation der Bewegungen des Klienten

Die Beschaffung des SpiderCam Systems erfolgte anhand der in AP 21000-D erstellten Anforderungen. In Zusammenarbeit mit SpiderCam wurde das System im vierten Quartal 2009 an die Gegebenheiten in der Space-Exploration Halle am DFKI RIC angepasst. Die Software-Schnittstelle zum SpiderCam System wurde spezifiziert und mit dem Lieferanten abgesprochen. Im Dezember 2009 fand die Werksabnahme bei SpiderCam in Österreich statt. Die Abnahme war erfolgreich und die vor Ort messbaren Größen Geschwindigkeit (0.001m/s bis 2m/s) und halbe Zuladung (75kg) wurden erfüllt.

Der Kuka KR-60 wurde beschafft und im dritten Quartal 2009 in Betrieb genommen. Die Aufstellung in der Space-Explorations-Halle wurde dabei so ausgelegt, dass mehrere verschiedene Positionen des Systems möglich sind. Somit können verschiedene Szenarien, vor allem der lange Anflug und das Fly-Around, simuliert werden.

Komponenten

Für die automatische Steuerung des KUKA KR-60 durch einen PC wurde für die KUKA Steuerung eine Ethernetkarte mit zugehöriger Robot Sensor Interface Software (RSI) gekauft. Über diese Schnittstelle können dem Manipulator alle 12 ms neue Sollpositionen oder Achswinkel vorgegeben werden.

Die SpiderCam wurde planmäßig in die Explorationshalle eingebaut. Sie kann manuell über ein Handbediengerät sowie über eine PC Schnittstelle gesteuert werden. Der Arbeitsbereich wird durch zwei Lichtvorhänge begrenzt, die gleichzeitig den Not-Aus

des Systems beim Betreten des Arbeitsbereichs auslösen. Die Anlage wurde wurde vom TÜV abgenommen. Mit der Anlage können Punkte mit einer Wiederholgenauigkeit kleiner 5 mm angefahren werden. Die Schwingungen in Richtung eines 1 m langen Auslegers sind kleiner $8°$, quer dazu kleiner $12°$ und Schwingungen um die Hochachse sind kleiner $32°$.

Es wurde ein Programm zur automatischen SpiderCam Steuerung über eine PC Schnittstelle geschrieben. Dieses sendet relative Positionsänderungen zur SpiderCam über einen CAN Bus und wertet folgende Statusinformationen aus:

- Anlage aktiv
- Bereit für externe Daten
- Absolutposition (aus Kabellängen berechnet)
- Positionsgrenze erreicht
- Geschwindigkeitsgrenze erreicht
- Beschleunigungsgrenze erreicht
- Zielkoordinate erreicht

Der SpiderCam müssen relative Positionsänderungen mitgeteilt werden. Unter Betrachtung eines 4 ms Zyklus und der maximalen Geschwindigkeit von 2 m/s ergibt sich eine maximale Positionsänderung pro Zyklus von 8 mm. Sollte die Relativkoordinate diesen Wert überschreiten, wird die Anlage auf die Maximalgeschwindigkeit maximal beschleunigt ($1,5\,m/s^2$) und versucht die Koordinate zu erreichen.

Abb. 9 zeigt die entwickelte graphische Benutzeroberfläche (GUI). Sie erlaubt dem Bediener die Einsicht der Statusinformationen und das Steuern der SpiderCam. Es können manuell Geschwindigkeiten vorgegeben und vordefinierte Trajektorien geladen werden.

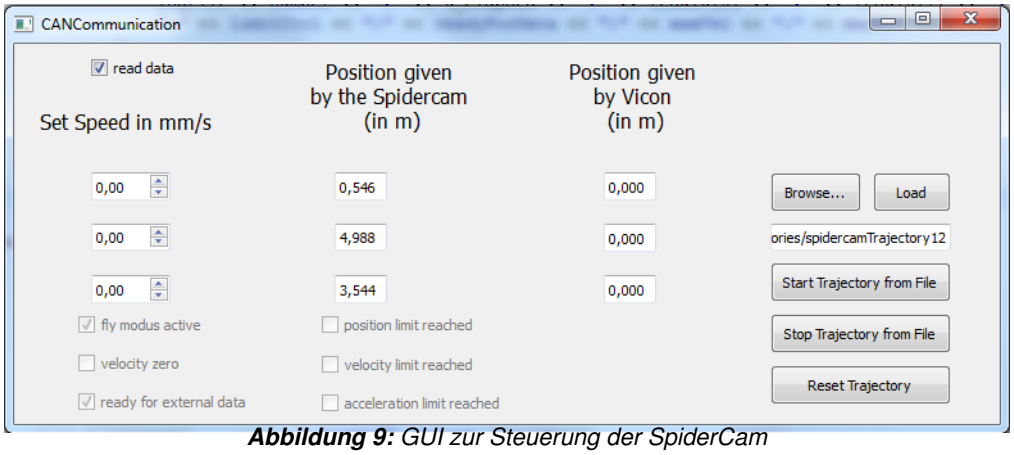


Abbildung 9: *GUI zur Steuerung der SpiderCam*

In der Explorationshalle wurde ein Motion Tracking System (MTS) der Firma Vicon installiert. Dieses besteht aus sechs Kameras. Es gibt millimetergenaue Positionsangaben von Kugelreflektoren aus, solange diese von mindestens zwei Kameras gesehen

werden. Die Genauigkeit wurde mit dem hochauflösenden Leica Laserscanner überprüft. Dafür wurden Positionen von zehn Markern (verteilt im unteren Bereich der Explorationshalle) mit dem MTS und dem Laserscanner gemessen. Der Vergleich der Distanzen zwischen den Markern ergab, dass die von der MTS gemessenen Strecken um ca. 0,25% bis 0,5% kleiner als die mit dem Laserscanner gemessenen Distanzen sind. Das Ergebnis wurde bestätigt, durch zusätzliche Messungen mit einem Maßband. Die Genauigkeit des MTS kann minimiert werden, in dem Körper aus mehreren Markern definiert werden. Dabei steigt die Genauigkeit je weiter die Marker des Körpers von einander entfernt sind. Aus diesem Grund wurde ein Aluminium-Kreuz an der SpiderCam befestigt, welches mehrere Marker trägt, die fast immer von allen Kameras gesehen werden können. Die Genauigkeit der MTS ist ebenfalls von der Bildfrequenz abhängig. Bei ungefähr 100 Hz oszilliert die Positionsangabe eines feststehenden Markers um +/- 1 mm während bei einer Frequenz von 250 Hz sich eine Oszillation von +/- 10 mm einstellt.

Um die Genauigkeit der SpiderCam zu überprüfen wurden mit dem System vordefinierte Trajektorien abgefahren und anschließend die Positionsdaten von der SpiderCam mit denen vom MTS verglichen (siehe Abb. 10). Hier müssen noch weitere Präzisionsexperimente durchgeführt werden. Jedoch haben bereits erste Tests gezeigt, dass die Koordinatensysteme MTS und SpiderCam nicht einfach ineinander überführt werden können. Die absoluten Koordinaten aus der SpiderCam weisen zum Teil nicht-lineare Abweichungen auf.

Wegen der unzulänglichen absoluten Positionsangaben der SpiderCam wird deshalb das MTS dazu verwendet, um eine übergeordnete Positionsregelung zu entwerfen. Die Regelung müsste alle 4 m neue Positionsänderung zur Verfügung stellen. Aufgrund der ungenaueren Messwerte der MTS bei 250 Hz sollte überlegt werden, die Regelfrequenz auf 83,3 Hz zu reduzieren und die benötigten Sollvorgaben für die SpiderCam zu interpolieren. Diese Frequenz würde sich anbieten, da diese der KUKA Regelfrequenz entspricht.

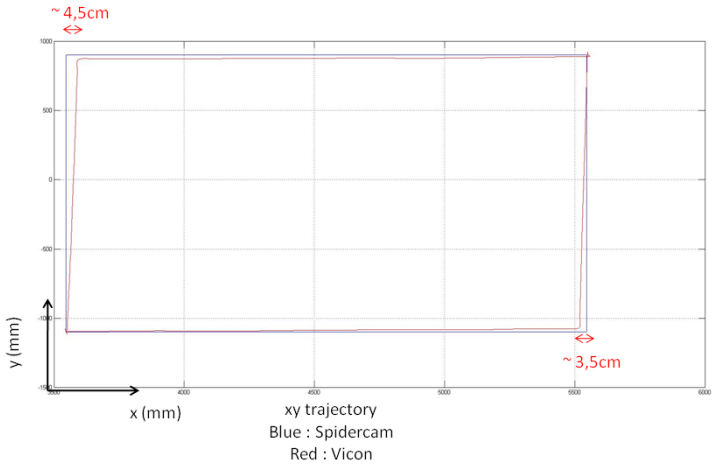


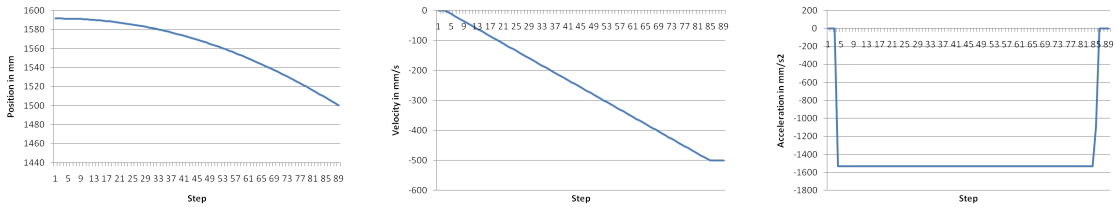
Abbildung 10: *Test : MTS und SpiderCam Messdaten einer abgefahrenen XY-Trajektorie*

Cable Robot Regelung

Dem Cable Robot müssen alle 4 ms neue Positionsänderungen zur Verfügung gestellt werden. Im Gegenzug erhält man die Absolutposition, die auf der Messung der Kabellängen basiert.

Steuerung des Cable Robots

Zunächst wurde experimentell festgestellt, wie der Cable Robot auf Positionsänderungen reagiert. Dazu wurde ihm eine konstante Geschwindigkeit von 500 mm/s vorgegeben, welche multipliziert mit der Periodendauer von 4 ms einer Positionsänderung pro Takt von 2 mm entspricht. Wie Abb. 11(b) zu entnehmen ist, wird die Geschwindigkeit aufgrund der begrenzten Beschleunigung nicht sofort erreicht. Diese beträgt laut den Experimenten 1529,75 mm/s^2. Generell wurde festgestellt, dass jede Positionsänderung exakt ausgeführt wird, sofern diese mit Rücksicht auf die Geschwindigkeits- und Beschleunigungsgrenzen gewählt wird. Wenn Positionsänderungen außerhalb dieser Grenzen an den Cable Robot geschickt werden, versucht dieser mit der maximalen Beschleunigung und der maximalen Geschwindigkeit von 2000 mm/s der Änderung so nahe wie möglich zu kommen. Wie vor allem in Abb. 11(c) deutlich wird, kann die maximale Beschleunigung in einem Zeitschritt erreicht werden. Jedoch besteht generell eine feste Verzögerung zwischen Befehl und messbarer Positionsänderung von 12 ms.



(a) Gemessene Position (b) Gemessene Geschwindigkeit (c) Gemessene Beschleunigung

Abbildung 11: *Reaktion des Cable Robots auf Sprungfunktion der Positionsänderung (2 mm pro Zeitschritt)*

Einfache Positionsregelung des Cable Robots

Für die Nutzung des Cable Robots sind absolute Positionsvorgaben oft nützlicher als Positionsänderungen. Aus diesem Grund musste ein Regler entwickelt werden, der die Sollposition mit der gemessenen Istposition vergleicht und daraus die neue Positionsänderung generiert. Abbildung 12 zeigt den entstehenden einfachen Regelkreis mit:

s_s - Sollwert in Absolutkoordinaten
s_d - Differenzposition als Regelwert in interne Cable Robot Regelung
s - Istwert in Absolutkoordinaten
s_m - gemessener Istwert in Absolutkoordinaten

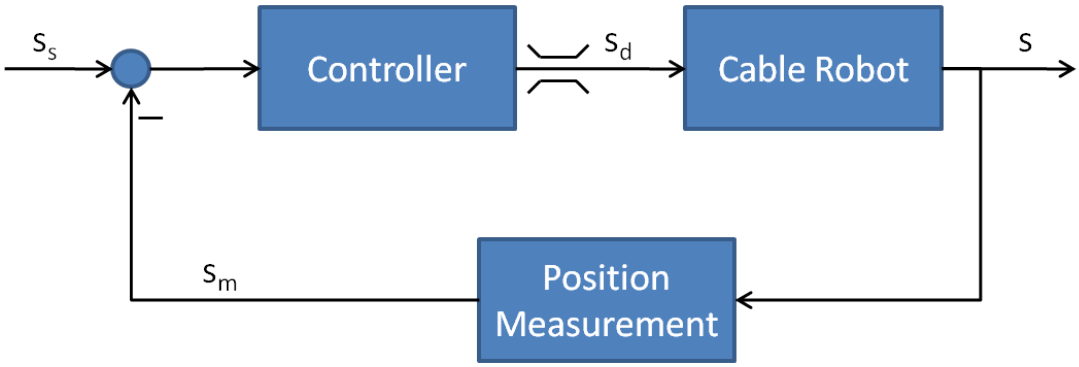
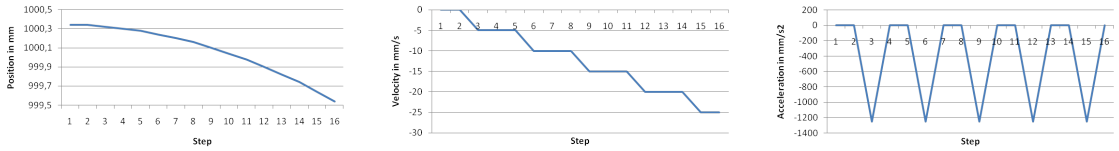


Abbildung 12: *Einfacher Regelkreis*

Es wurde ein PID-Regler implementiert, aber nur der P-Anteil genutzt und auf eins gesetzt. Höhere Werte würden Überschwinger erzeugen, während niedrigere Werte die Regelung verlangsamen würden. Der I- und D-Anteil wurde zunächst nicht berücksichtigt, da die interne Regelung des Cable Robots bereits einen kompletten PID-Regler nutzt. Nachteil bei dieser Lösung ist, dass ohne Begrenzung große Regelwerte entstehen, wodurch der Cable Robot mit voller Beschleunigung ruckhaft startet und abbremst. Um dies zu verhindern wurde eine Beschleunigungs- und Geschwindigkeitsbegrenzung in den Regler eingebaut. Diese begrenzt den Regelwert anhand einstellbarer Beschleunigungs- und Geschwindigkeitsgrenzen sowie der aktuell gemessen Positionsänderung. Experimente zeigten, dass Trajektorien mit Absolutkoordinaten gut nachgefahren werden konnten, solange die generierten Sollpositionen keine großen Beschleuingungsänderungen bedurften. Jedoch kann mit der implementierten Begrenzung nicht die volle Dynamik des Cable Robots ausgenutzt werden. Dadurch, dass die Begrenzung u.a. von der gemessenen Position abhängt, regelt der Regler auf einen Wert, der bereits 12 ms veraltet ist. Dadurch entsteht eine Treppenfunktion in der Regelwertvorgabe (siehe Abb. 13(b)). Die tatsächliche Beschleunigung beträgt damit nur ein Drittel, da nur alle 12 ms ein neuer Messwert zu einem neuen Regelwert führt (siehe Abb. 13(c)).



(a) Gemessene Position (b) Gemessene Geschwindigkeit (c) Gemessene Beschleunigung

Abbildung 13: *Bewegung des Cable Robots bei beschleunigungsbegrenzter Positionsregelung*

Regelung mit Sollwertgenerator

Damit eine Treppenfunktion vermieden werden kann, ist ein Sollwertgenerator notwendig, der Sollwerte innerhalb der Geschwindigkeits- und Beschleunigungsgrenzen erzeugt, sodass der Positionsregler lediglich Abweichungen ausregeln muss [14]. Der

Sollwertgenerator wurde so ausgelegt, dass er eine zeitoptimale Regelung ermöglicht, d.h. dass die Endposition möglichst schnell mit einer vorgebbaren maximalen Geschwindigkeit und Beschleunigung erreicht wird. Die berechneten Sollwerte liegen somit innerhalb der Grenzen, wodurch die Begrenzung am Reglerausgang entfallen kann. Abb. 14 zeigt die zeitoptimale Lageregelung mit integriertem Sollwertgenerator, der Referenzpositionen s_r erzeugt, die als Sollwerte in den eigentlichen Regelkreis eingespeist werden. Für den Sollwertgenerator musste ein Zusammenhang zwischen Position und Geschwindigkeit hergestellt werden. Diese Zustandstrajektorie beschreibt (1).

$$v_s = \sqrt{2 \cdot a_{max} \cdot |s_s - s_r|} \cdot sign(s_s - s_r) \tag{1}$$

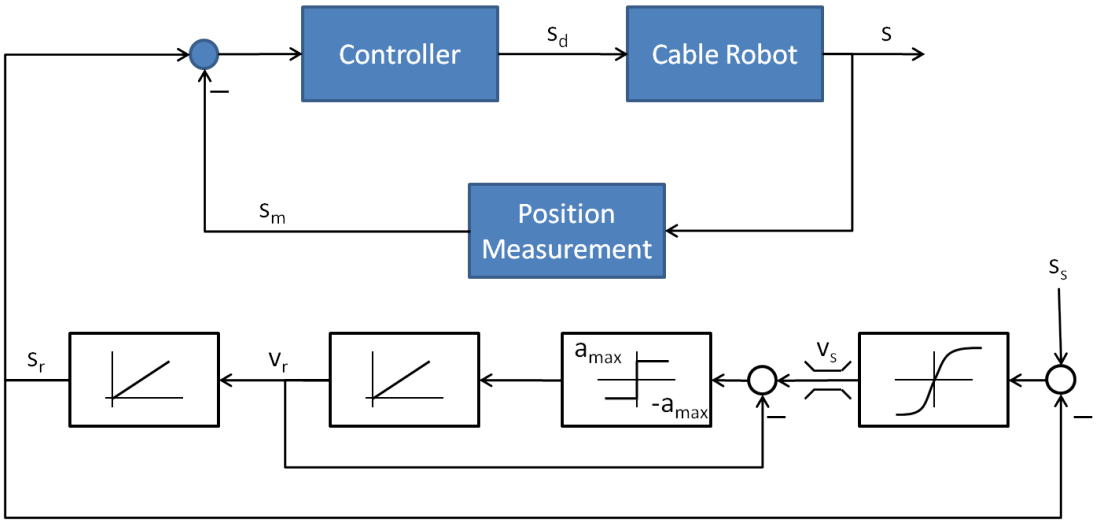
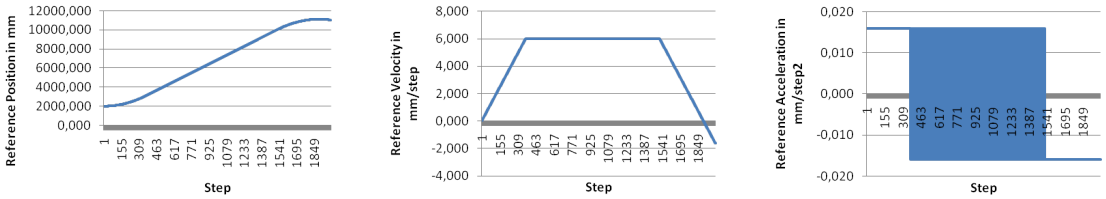


Abbildung 14: *Regelkreis mit Sollwertgenerator*

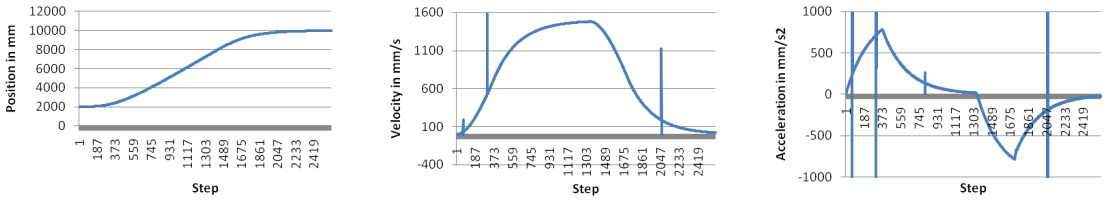
Wie Abb. 15(a) zeigt, erzeugt der Sollwertgenerator eine weiche Bewegung, um von 2000 mm nach 10000 mm zu fahren. Die Referenzbeschleunigung erreicht sofort ihren maximalen Wert von gewünschten 1000 mm/s^2, wodurch die Geschwindigkeit linear ansteigt, bis der gewünschte Maximalwert von 1500 mm/s erreicht wird. Das verwendete Zweipunktglied führt jedoch dazu, dass an Stellen gewünschter konstanten Geschwindigkeit die Beschleunigung nicht Null sondern abwechselnd den positiven und negativen Maximalwert erreicht (siehe Abb. 15(c)). Durch die Integration ist aber die Oszillation der Geschwindigkeit wesentlich kleiner und durch erneute Integration ist bei der erzeugten Referenzposition die Oszillation quasi nicht mehr vorhanden. Es entstehen aber Überschwinger in den Refernzwerten. Abb. 16(a) zeigt aber, dass durch den eingesetzten P-Regler diese sich nicht auf die Bewegung des Cable Robots übertragen. Es dauert jedoch länger, bis die gewünschte Sollposition erreicht wird. Dies kommt daher, dass die gewünschten Geschwindigkeiten und Beschleunigungen nicht erreicht werden, wie Abb. 16(b) und Abb. 16(c) zeigen. Der Grund hierfür ist die Regelung auf die gemessenen Positionswerte, die 12 ms Verzögerung besitzen. In diesem Beispiel wird ebenfalls deutlich, dass in der Kommunikation Fehler auftreten und damit Positionsänderungen sowie Positionsmessungen nicht ankommen. Die Position wird

nicht stark dadurch beeinflusst. Aber die abgeleitete Geschwindigkeit und die doppelt abgeleitete Beschleunigung erfahren große Peaks. Dadurch ruckt der Cable Robot, was kaum zu sehen, aber zu hören ist. Das Problem sollte mit der zu diesem Zeitpunkt geplanten Verwendung des dSpace-Systems beseitigt werden.



(a) Referenzposition (b) Referenzgeschwindigkeit (c) Referenzbeschleunigung

Abbildung 15: *Generierte Trajektorie durch Sollwertgenerator*



(a) Gemessene Position (b) Gemessene Geschwindigkeit (c) Gemessene Beschleunigung

Abbildung 16: *Abgefahrene Trajektorie durch Cable Robot*

Regelung mit Sollgeschwindigkeitsgenerator

Die ersten Experimente zeigten, dass solange die generierten Sollwerte innerhalb der dynamischen Grenzen des Cable Robots liegen, die gewünschten Positionen auch erreicht werden. Aus diesem Grund könnte generell auf einen Positionsregler verzichtet werden, der den Nachteil besitzt, dass auf 12 ms veraltete Positionswerte geregelt wird. Dafür muss ein Sollgeschwindigkeitsgenerator entwickelt werden, der aufgrund der aktuellen Position und Geschwindigkeit, die nötigen Geschwindigkeiten berechnet, um einen gewünschten Zielpunkt mit Geschwindigkeit Null zu erreichen. Der vorgestellte Sollwertgenerator (siehe Abb. 14) eignet sich nicht, da dieser zum einen eine leicht oszillierende Geschwindigkeit erzeugt und zum anderen keine Startgeschwindigkeit v_0 zulässt. Das Geschwindigkeitsprofil kann, wie in Abb. 17 dargestellt, aussehen.

Es müssen die Zeitintervalle berechnet werden wie lange voll beschleunigt und wie lange die maximale Geschwindigkeit gehalten werden darf. Die Zeit t_1, wie lange positiv oder negativ beschleunigt werden muss, um die gewünschte maximale positive oder negative Geschwindigkeit zu erreichen, berechnet sich nach (3). Generell gilt wenn $|v_0| < |v_{max}|$ (sonst Vorzeichen von a vertauscht):

$$\begin{aligned} v = v_{max}\ und\ a = a_{max},\ wenn\ s_s - s_0 > 0 \\ v = -v_{max}\ und\ a = -a_{max},\ wenn\ s_s - s_0 < 0 \end{aligned} \tag{2}$$

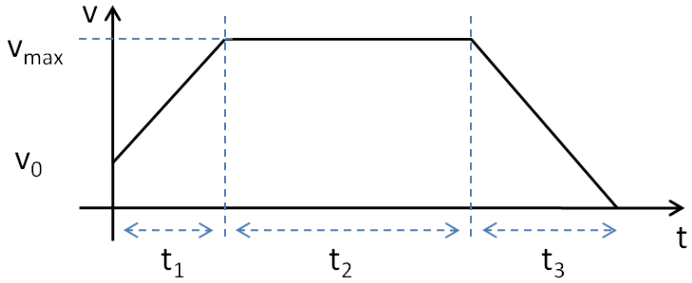


Abbildung 17: *Regelkreis mit Sollwertgenerator*

$$t_1 = |\frac{v - v_0}{a}| \tag{3}$$

Die Zeit t_3, die angibt, wie lange abgebremst werden muss, um am Zielort keine Restgeschwindigkeit zu besitzen, berechnet sich nach (4):

$$t_3 = \frac{v}{a} \tag{4}$$

Die Intervalllänge t_2, in der mit konstanter Geschwindigkeit verfahren wird, berechnet sich aus dem Gesamtweg von Start- bis Sollposition, und den Zeiten, die für das Beschleunigen notwendig sind (t_1 und t_3). Es ergibt sich (5):

$$t_2 = \frac{s_s - s_0 - \frac{1}{2}at_1^2 - v_0t_1 + \frac{1}{2}at_3^2 - vt_3}{v} \tag{5}$$

Falls t_2 kleiner 0 wird, kann die gewünschte Maximalgeschwindigkeit nicht erreicht werden, da der Positionsunterschied zu gering ist. Dann ergibt sich für t_1 (6) und für t_3 (7):

$$t_1 = -\frac{v_0}{a} + \sqrt{\frac{v_0^2}{2a^2} + \frac{s_s - s_0}{a}} \tag{6}$$

$$t_3 = \frac{v_0}{a} + t_1 \tag{7}$$

In der ersten Phase ($t < t_1$) wird die Referenzgeschwindigkeit nach (8) berechnet:

$$v_r = at + v_0 \tag{8}$$

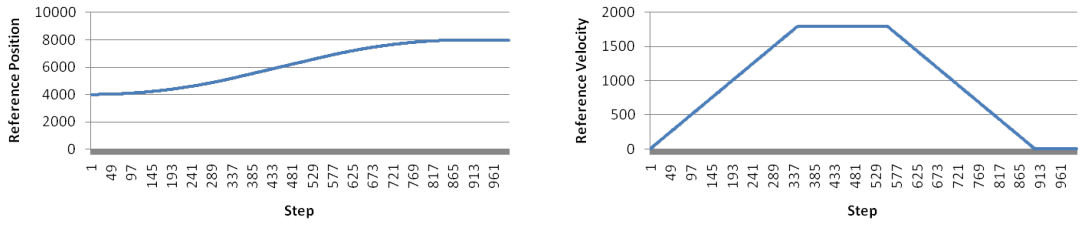
In der zweiten Phase ($t < t_1 + t_2$) wird die Referenzgeschwindigkeit nach (9) berechnet:

$$v_r = v \tag{9}$$

Und in der dritten Phase ($t < t_1 + t_2 + t_3$) wird die Referenzgeschwindigkeit nach (10) berechnet, wobei v_{max} in diesem Fall auch die erreichte Geschwindigkeit sein kann:

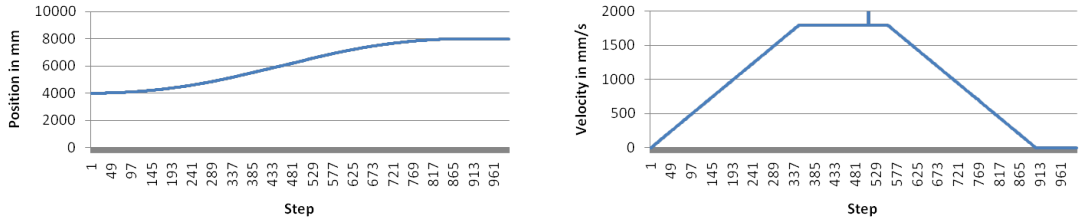
$$v_r = v - a(t - t_1 - t_2) \tag{10}$$

Danach wird die Geschwindigkeit konstant auf Null gehalten. Abbildung 18(b) zeigt ein Beispiel eines erzeugten Geschwindigkeitsprofils. Außerdem wird über Integration die Referenzpositionen berechnet (siehe Abb. 18(a)). Die Graphen in Abb. 19 zeigen, dass sowohl Position als auch Geschwindigkeit vom Cable Robot umgesetzt worden sind. Die Graphen sind nur um 3 Zeitschritte (= 12 ms) aufgrund der oben angesprochenen Verzögerung verschoben. Das abgefahrene Geschwindigkeitsprofil (siehe Abb. 19(b)) zeigt eine Diskontinuität, die wieder durch Datenverlust in der Kommunikation entstanden ist. Dadurch entstehen Schleppfehler, die ausgeregelt werden müssen. Der Fehler kann durch einen Vergleich zwischen der gemessenen Position und der errechneten Position vor 12 ms ermittelt werden und als zusätzliche Positionsänderung aufaddiert werden (siehe Abb. 20). Da aufgrund von Kommunikationsfehlern zu diesem Zeitpunkt die Regelung noch negativ beeinflusst wurde, mussten abschließende Experimente erfolgen, sobald das dSpace-System zur Kommunikation eingesetzt wurde.



(a) Referenzposition (b) Referenzgeschwindigkeit

Abbildung 18: *Generierte Trajektorie durch Sollgeschwindigkeitsgenerator*



(a) Gemessene Position (b) Gemessene Geschwindigkeit

Abbildung 19: *Abgefahrene Trajektorie durch Cable Robot*

Motion Tracking System Kalibrierung

Das Motion Tracking System (MTS) dient zur Genauigkeitsüberprüfung des Cable Robots. Um eine Aussage über die Genauigkeit des Cable Robots machen zu können,

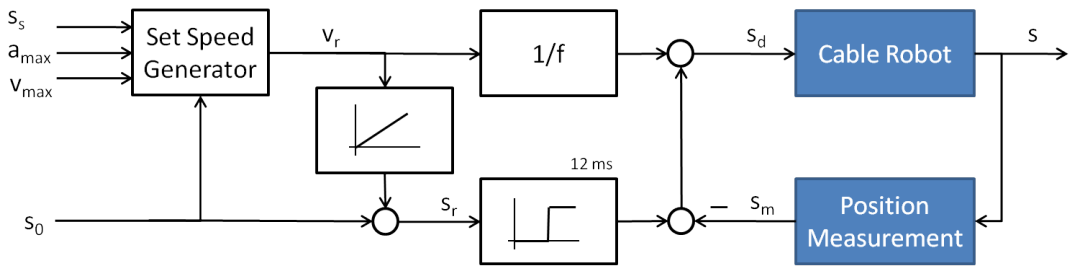


Abbildung 20: *Regelkreis mit Sollgeschwindigkeitsgenerator*

muss die Genauigkeit des MTS bekannt und hoch genug sein. Wie oben beschrieben, wurde festgestellt, dass das MTS um ca. 0,25% - 0,5% kleinere Distanzen, als Maßband und Laserscanner, ermittelt. D.h. auf 4 m Entfernung entstanden bereits Fehler von über 1 cm. Der Grund für die Abweichung ist die Systemkalibrierung. Der zu verwendete Kalibrierstab war ungenau gefertigt und wurde deshalb umgetauscht. Die Ergebnisse nach der Kalibrierung mit dem neuen Stab waren unwesentlich besser. Mit Rücksprache der Vicon Ingenieure ist nun der Grund für mangelnde Genauigkeit die Auflösung der Kameras. In ca. 10 m Entfernung bedecken die kleinen Marker ca. 1 Pixel, der auf dieser Entfernung ca. 1 cm entspricht. Somit sind Fehler um 0,5 cm kaum zu vermeiden. Es müsste ein neue MTS mit höherer Auflösung gekauft oder größere Marker eingesetzt werden. Es wurden deshalb Kugeln mit einem Durchmesser von 5 cm mit reflektierendem Selbstklebeband beklebt, um neue größere Marker herzustellen. In die Marker wurden Gewinde geschnitten, damit mit ihnen ein neues Subjekt des Cable Robots und ein neues Koordinatenkreuz erstellt werden konnte. Das neue größere Koordinatenkreuz mit den neuen Markern sollte helfen, das Koordinatensystem des MTS möglichst gut an die Explorationshalle anzupassen. Dann musste ein Verdecken der großen Marker ausgeschlossen werden. Dieses führte dazu, dass der Mittelpunkt des von Vicon erkannten Objekts sich bewegt, da auch ein Teil des Markers für die Markerdetektion ausreicht.

Verbesserung des Motion Tracking Systems

Für das Motion Tracking System werden nun Marker mit doppeltem Durchmesser (5 cm statt 2,5 cm) verwendet, damit mehr Pixel auf den Kamerabildern bedeckt werden und der Mittelpunkt der Kugel genauer bestimmt werden kann. Das Koordinatensystem wird nun mit einem Frame von ca. 2 m Kantenlänge bestimmt, damit Ungenauigkeiten in der Fertigung einen kleineren Einfluss auf die Ausrichtung des Koordinatensystems besitzen. Die XY-Ebene entspricht nun wesentlich genauer dem Hallenboden. Zusätzlich wird das MTS jetzt mit einem steiferen und genaueren Kalibrierstab kalibriert. Eine erneute Genauigkeitsüberprüfung mit Hilfe des Leica Laserscanners wurde durchgeführt, um den Nutzen der Änderungen zu verifizieren. Dazu wurden Marker in der Halle verteilt und deren Position sowohl mit dem Laserscanner als auch mit dem MTS vermessen und anschließend die Distanzen zwischen den Markern mit einander verglichen. Das Ergebnis ist, dass das MTS im Durchschnitt eine um 0,2% kleinere Distanz misst als der Laserscanner. Im Vergleich dazu lag der Fehler vor den Verbesserungen

bei 0,25 - 0,5%. Im internen Arbeitsbereich liegt der mittlere Fehler unter 5 mm. Im erweiterten Arbeitsbereich steigt dieser jedoch bis auf 20 mm an. Die Messmethode besitzt eine Genauigkeit von unter 2 mm.

Kuka Steuerung

Der Kuka KR-60 wird mit einem Remote Sensor Interface (RSI) gesteuert. RSI besteht aus einer Ethernetkarte und einer KRL Objektsbibliothek (KRL = Kuka Robot Language). RSI erlaubt eine TCP/IP Kommunikation zwischen dem Roboter und einem Computer.

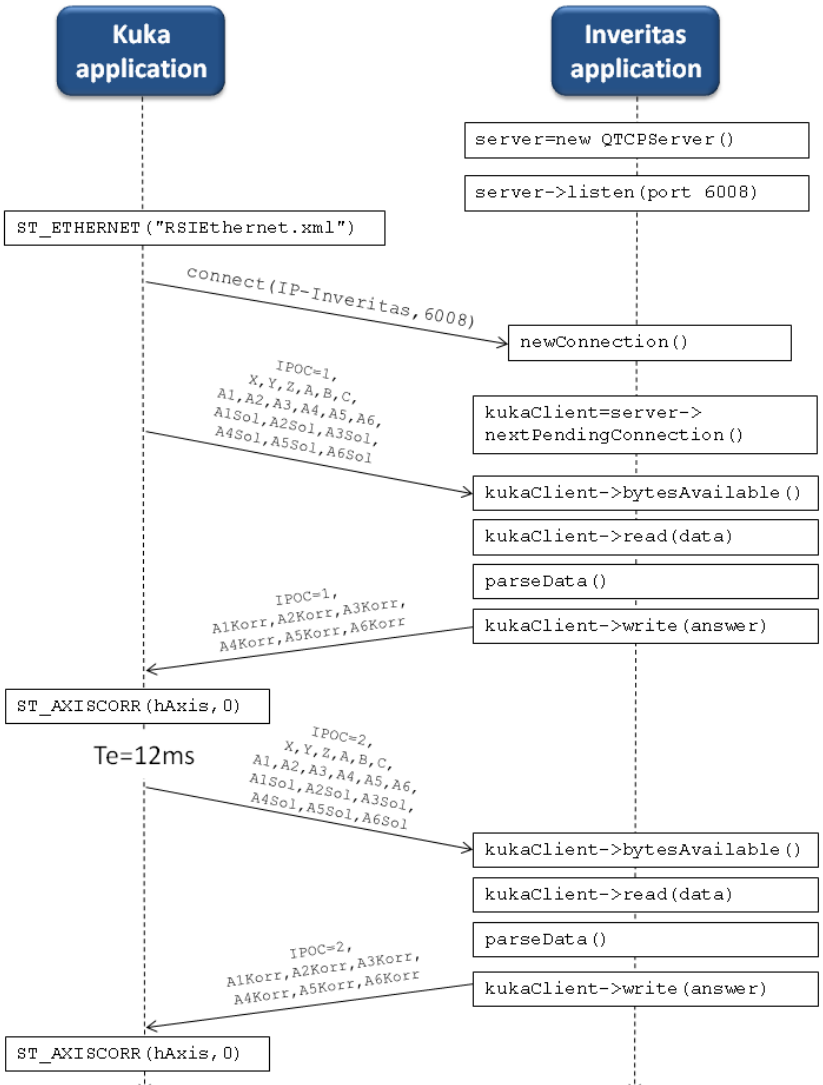


Abbildung 21: *Kommunikationsprotokol zwischen Kuka und einem externen Computer*

In einer XML Nachricht sendet der Roboter alle 12 ms seine Position, Orientierung sowie Achskonfiguration. Im Gegenzug erwartet die Robotssteuerung eine XML Nachricht mit Sollposition oder Sollwinkeln. Sollte eine Antwort verloren gehen oder zu spät

eintreffen, verwendet der KUKA erneut den zuletzt übermittelten Wert. Nach 10 verlorenen oder verspäteten Antworten gibt das System einen Fehler aus und stoppt die Bewegung. Der Steuerung können entweder relative Positions- bzw. Winkeländerungen oder absolute Positionen oder Winkel geschickt werden. Die Absolutwerte beziehen sich auf die Initialkonfiguration des Roboters.

Abbildung 21 zeigt das Kommunikationsprotokoll zwischen dem Roboter und einem externen Computer. Wenn eine Antwort verloren ist oder zu spät kommt nimmt Kuka die letzte Korrektur.

Die Wiederholgenauigkeit des KR-60 beträgt ± 0.20 mm. Die Antwort des Roboters im geöffneten Regelkreis wurde getestet und sieht wie folgt aus:

- Um t=0 wurde eine Winkeländerung an Kuka geschickt
- nach 24 ms (2 Zyklen) hat Kuka die Korrektur erfasst
- nach 36 ms (1 weiterer Zyklus) fängt die Bewegung an

Für die Achse 6 wurde die Sprungantwort getestet (siehe Abbildung 22) Die schwarze Linie represäntiert die gewünschte Positionskorrektur, die an die Steuerung geschickt wurde. Die rote Linie represäntiert den Sollwinkel des internen Systems von Kuka. Es ist zu sehen, dass die Steuerung 2 Zyklen benötigt (24 ms) um auch die interne Sollwertvergabe anzupassen. In blau ist der aktuelle Winkel der Achse 6 eingezeichnet. Die Achse erreicht 95 % des Zielwinkels (in grün) nach ca. 120 ms. Diese Zeit ist unabhängig von dem Wert, wenn der Wert erreichbar ist (z.B. $0.1°$, $1°$ oder $2°$). Der erreichte Winkel hat eine Genauigkeit von $0.01°$ (1%).

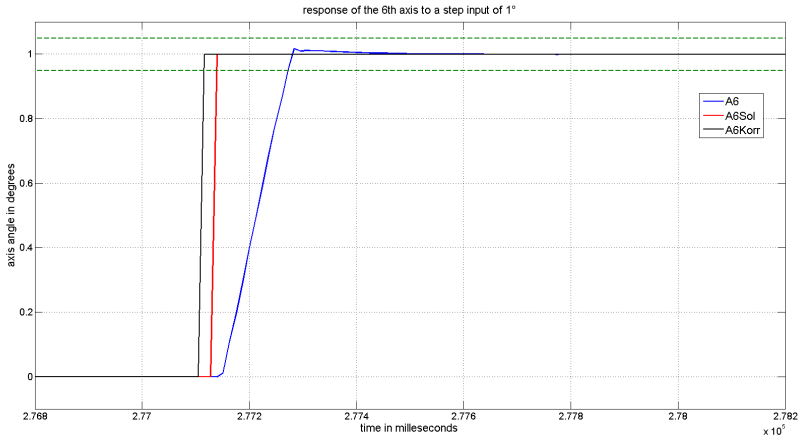


Abbildung 22: *Sprungantwort der Achse 6*

Abbildung 23 zeigt die Drehung einer $1°/s$ Rampenfunktion. In schwarz ist die geschickte Korrektur, in rot der interne Sollwinkel und in blau die Drehung der sechsten Achse abgebildet. Die sechste Achse braucht ungefähr 120 ms (10 Zyklen) um eine Geschwindigkeit von $1°/s$ zu erreichen. Danach existiert eine Verzögerung von ca. 85 ms (7 Zyklen) zwischen Korrekturbefehl und tatsächlicher Drehung. Die erreichte Geschwindigkeit hat eine Genauigkeit von $0.01°/s$ (1%).

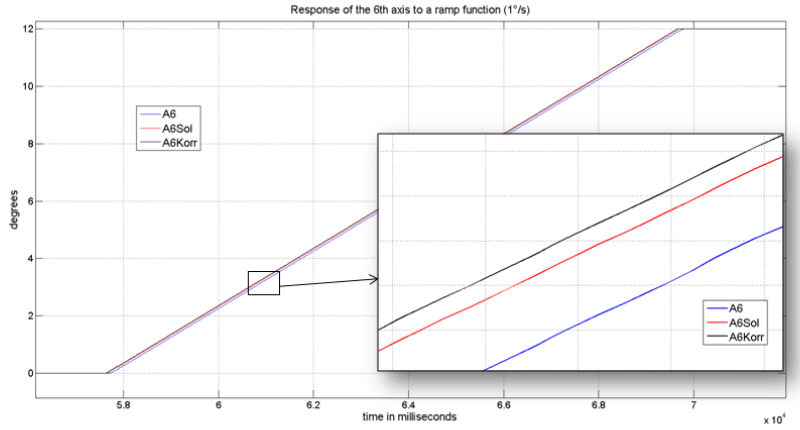


Abbildung 23: *Sprungantwort der Achse 6 nach eine 1 Grad/Sekund Rampenfunktion*

Inertialmesseinheit

Das Bewegungssystem wurde im dritten Quartal 2010 um eine IMU (Inertialmesseinheit) erweitert. Dieses Gerät der Firma XSens liefert hochfrequente (100Hz) Daten über Beschleunigung in den drei Hauptachsen sowie Winkelbeschleunigung in den drei Drehachsen. Zusammen mit der bereits vorhandenen Positionierungssoftware des Bewegungssystems (auf Basis der Seillängen und des Trackingsystems) sollte so durch Fusion der Daten (z.B. mit Hilfe eines Kalman-Filters) die Positionierungsgenauigkeit auch bei hohen lokalen Beschleunigungen oder in Extrempositionen verbessert werden.

Umbauten am Cable Robot

Durch das Arbeiten mit dem Cable-Robot sind im Laufe der Zeit bis zum dritten Quartal 2010 folgende Fehlfunktionen aufgetreten:

- Steuerpult wird von der Steuerungssoftware nicht erkannt. Ausziehen und erneutes Einstecken behebt das Problem.
- Das Öffnen der Bremse bei Winde 1 wird von dem dafür vorgesehenen Schalter nicht detektiert, da dieser nicht korrekt eingestellt ist. Das Problem wurde durch das Einsetzen eines Zwischenstücks vorerst behoben.
- Das untere Kabel von Winde 3 war gerissen.

Der Riss des Kabels ist auf einen Fertigungsfehler der Kabel mit integrierter Stromleitung zurückzuführen. Die dynamischen Belastungen der Kabel führten dazu, dass

die Leitungen der Winde 3 keinen konstanten Kontakt hatten. Durch das Öffnen und erneute Schließen des Kontakts aufgrund der Bewegung des CableRobots ist es sehr wahrscheinlich, dass hohe Temperaturen entstanden sind, wodurch das Kevlar verbrannte und das Kabel riss. Aus diesem Grund wurden zusammen mit einem Techniker der SPIDERCAM GmbH alle stromführenden Kabel, also die von Winde 3 und 4, durch besser gefertigte Kabel ersetzt. Außerdem wurde eine Sicherung pro Kabel eingesetzt, die ein Reißen des Seils aus diesem Grund in Zukunft vorzeitig verhindert.

Außerdem wurden die Schalter von Winde 1 wieder korrekt eingestellt. Das Problem mit dem Steuerpult konnte vorerst nicht behoben werden. Die SPIDERCAM Techniker versuchten eine Lösung zu finden. Desweiteren wurden kleinere Verbesserungen an der GUI der Steuerungssoftware vorgenommen. Den Technikern wurde gezeigt, wie weit die Position, die der CableRobot anhand der Seillängen ausrechnet, von der mit dem MTS gemessen Position abweicht. Da ihnen die Möglichkeit einer Referenzmessung fehlt, sie ihren Algorithmus zur Positionsbestimmung aber gerne verbessern würden, ist nun angedacht enger zusammenzuarbeiten. Iterativ würden sie ihren Algorithmus verbessern und wir ihre Positionsbestimmung mit dem MTS verifizieren. So könnte die Präzision der Anlage nach und nach erhöht werden.

dSpace Integration

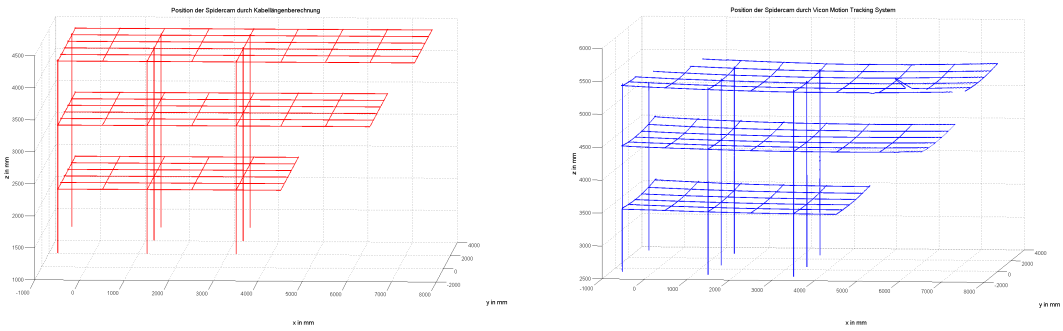
Für eine robuste Kommunikation in Echtzeit mit dem CableRobot, dem KUKA und dem Motion Tracking System (MTS) wurde im dritten Quartal 2010 ein dSPACE System mit 4 Kernen, Ethernet- und CAN-Schnittstellen gekauft. Mit diesem System ist es nun möglich, Matlab/Simulink Modelle in harter Echtzeit auszuführen. Dafür müssen diese mit dem Realtime Workshop kompiliert und anschließend auf die dSpace-Kerne kopiert werden. Diese Variante erlaubt eine schnelle und unkomplizierte Einbindung von anderen Simulationsmodellen z.B. von Astrium.

Für die bereits entwickelte Software waren Anpassungen erforderlich. Es musste ein Matlab Simulink Modell erstellt werden, dass zum einen alle Schnittstellen verwaltet und zum anderen die oben beschriebenen Funktionen enthält. Die CAN-Schnittstelle zum CableRobot wurde nun durch Simulink Blöcke realisiert. Für die restlichen Schnittstellen (Simulation, Visualisierung, KUKA, MTS, ...) wird Ethernet benötigt. Diese Verbindungsart ist mit dem gekauften dSpace System nur über einen dedizierten Linux Kern möglich. Alle Funktionen wurden dort neu implementiert. In dem gegenwärtigen Setup wird beim Start des dSpace Systems ein Server gestartet, der alle Ethernetverbindungen verwaltet. Es können sich Clients während der Laufzeit mit diesem Server verbinden und die entsprechenden Informationen austauschen.

Positionskorrektur des CableRobots

Die Verbindung zum CableRobot funktioniert seit der dSpace Umstellung ohne Probleme. Dadurch konnten weitere Präzisionsexperimente durchgeführt werden. Es wurden definierte Bahnen mit der SpiderCam abgeflogen und die interne Positionsmessung

der SpiderCam mit der genaueren Positionsmessung des Motion Tracking Systems (MTS) verglichen. Bei jeder Bahn verlief die Bewegung in jeweils nur eine Achsrichtung des Koordinatensystems, während die anderen zwei Achsrichtungen konstant gehalten wurden (siehe Abb. 24).



(a) Die Positionsmessung der SpiderCam durch Kabellängenberechnung zeigt gerade Bahnen

(b) Die Positionsmessung der SpiderCam durch das Motion Tracking System zeigt, dass in Wirklichkeit parabelförmige Bahnen abgefahren wurden

Abbildung 24: *Vergleich der internen Positionsbestimmung der SpiderCam über Kabellängen und der Positionsbestimmung mittels Motion Tracking System*

Die Ergebnisse des Experiments zeigten, dass die interne Positionsmessung über Kabellängen nicht rauscht. Die Messung des MTS rauscht zwar leicht, ist aber absolut genauer. Der Vergleich zeigt, dass die SpiderCam Höhe zur Mitte der Halle parabelförmig abnimmt, obwohl eine konstante Höhe eingehalten werden sollte. Dieser Fehler liegt in der Ungenauigkeit des internen Bewegungsmodells der SpiderCam begründet. Doch kann das MTS nur bedingt in normalen Anlagenbetrieb eingesetzt werden, da zum einen nicht der gesamte Arbeitsbereich der SpiderCam erfasst werden kann und zum anderen das infrarote Licht des MTS die Kameras des Servicers beeinflussen würde. Da in das Bewegungsmodell der SpiderCam von unsere Seite aus nicht eingegriffen werden kann, entstand die Idee, die SpiderCam im Sichtbereich des MTS zu bewegen, die Positionsabweichungen aufzunehmen und anhand dieser Kalibrierergebnisse eine Polynom-Näherungsfunktionen für jede xyz-Komponente zu berechnen, die den Abstand zwischen MTS-Messung und Seillängenmessung angibt.

Im Folgenden wird erläutert, wie schrittweise von einer eindimensionalen Näherungsfunktion ein dreidimensionale errechnet wird. Die Abhängigkeit zwischen x und z zum Beispiel wird wie folgt berechnet.

$$\Delta z = z_{Kabel} - z_{Vicon} = \begin{bmatrix} x^5 & x^4 & x^3 & x^2 & x & 1 \end{bmatrix} \cdot \begin{bmatrix} p_5 \\ p_4 \\ p_3 \\ p_2 \\ p_1 \\ p_0 \end{bmatrix} \tag{11}$$

$$\Delta z = \begin{bmatrix} x_1^5 & x_1^4 & x_1^3 & x_1^2 & x_1 & 1 \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ x_n^5 & x_n^4 & x_n^3 & x_n^2 & x_n & 1 \end{bmatrix} \cdot \begin{bmatrix} p_5 \\ p_4 \\ p_3 \\ p_2 \\ p_1 \\ p_0 \end{bmatrix} = V_{1d} \cdot P_{1d} \tag{12}$$

$$V_{1d} = Q \cdot R \tag{13}$$

$$P_{1d} = R^{-1} \cdot Q' \cdot \Delta z \tag{14}$$

Hier soll die Näherungsfunktion durch ein Polynom 5-ten Grades beschrieben werden, weswegen sechs Koeffizienten bestimmt werden müssen. Während die Gleichung 11 die vertikale Positionsdifferenz für einen Punkt beschreibt, zeigt Gleichung 12 den Fall für n aufgenommene Punkte. Die Matrize V ist eine Vandermonde-Matrize, die in eine orthonormal Matrix Q und eine Dreiecksmatrize R zerlegt werden kann (Gleichung 13). Die Inverse von R multipliziert mit der Transponierten von Q und dem gemessenen Positionsunterschied Δz ergibt schließlich die Koeffizienten der Näherungsfunktion. Da die Messwerte nicht genau auf der Näherungsfunktion eines n-ten Polynoms liegen, wird mit der 'Least-Square' Optimierung, die Näherungsfunktion mit dem kleinsten Fehler gesucht. Abb. 25 zeigt das Ergebnis.

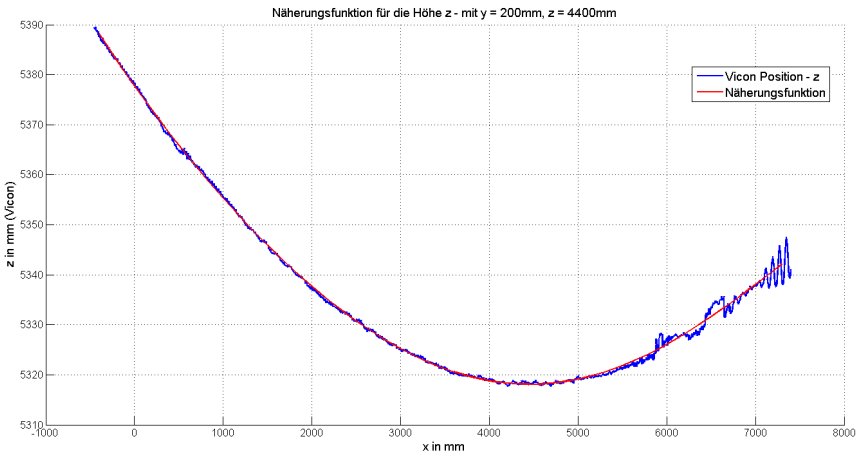


Abbildung 25: *Näherungsfunktion der Höhe der SpiderCam für z = 4400 mm, y = 200 mm*

Für den zweidimensionalen Fall, ist ein Polynom höherer Ordnung nötig. Die Koeffizienten einer Funktion 10.Grades sind dann in diesem Fall von x und y abhängig

(Gleichung 15). Die nächsten Schritte sind analog zum eindimensionalen Fall.

$$\Delta z = z_{Kabel} - z_{Vicon} = \begin{bmatrix} x^3 & x^2y & xy^2 & y^3 & x^2 & xy & y^2 & x & y & 1 \end{bmatrix} \begin{bmatrix} p_{10} \\ p_9 \\ \dots \\ \dots \\ p_1 \\ p_0 \end{bmatrix} \tag{15}$$

$$\Delta z = \begin{bmatrix} x_1^3 & x_1^2y_1 & x_1y_1^2 & y_1^3 & x_1^2 & x_1y_1 & y_1^2 & x_1 & y_1 & 1 \\ . & . & . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . & . & . \\ x_n^3 & x_n^2y_n & x_ny_n^2 & y_n^3 & x_n^2 & x_ny_n & y_n^2 & x_n & y_n & 1 \end{bmatrix} \cdot \begin{bmatrix} p_{10} \\ p_9 \\ . \\ . \\ p_1 \\ p_0 \end{bmatrix} = V_{2d} \cdot P_{2d} \tag{16}$$

$$V_{2d} = Q \cdot R \tag{17}$$

$$P_{2d} = R^{-1} \cdot Q' \cdot \Delta z \tag{18}$$

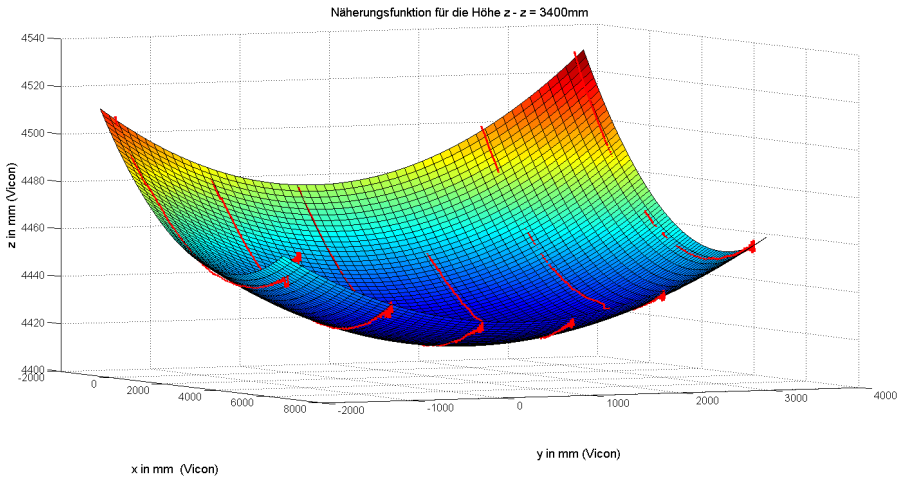


Abbildung 26: *Näherungsfunktion der Höhe der SpiderCam - z = 3400 mm*

Abbildung 26 zeigt die 2D-Näherungsfunktion für eine bestimmte Höhe. Um eine 3-D Näherungsfunktion für die z-Korrektur zu erhalten wird Gleichung 19 benötigt und

analog weiter vorgegangen wie im eindimensionalen Fall.

$$\delta z = \begin{bmatrix} x^3 & x^2y & x^2z & xy^2 & xyz & y^3 & y^2z & yz^2 & z^3 & x^2 & xy & xz & y^2 & yz & z^2 & x & y & z & 1 \end{bmatrix} \begin{bmatrix} p_{20} \\ p_{19} \\ \cdots \\ \cdots \\ p_1 \\ p_0 \end{bmatrix} \tag{19}$$

$$\Delta z = V_{3d} \cdot P_{3d} \tag{20}$$

$$V_{3d} = \begin{bmatrix} x_1^3 & x_1^2y_1 & x_1^2z_1 & x_1y_1^2 & x_1y_1z_1 & y_1^3 & y_1^2z_1 & y_1z_1^2 & z_1^3 & x_1^2 & x_1y_1 & x_1z_1 & y_1^2 & y_1z_1 & z_1^2 & x_1 & y_1 & z_1 & 1 \\ . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . & . \\ x_n^3 & x_n^2y_n & x_n^2z_n & x_ny_n^2 & x_ny_nz_n & y_n^3 & y_n^2z_n & y_nz_n^2 & z_n^3 & x_n^2 & x_ny_n & x_nz_n & y_n^2 & y_nz_n & z_n^2 & x_n & y_n & z_n & 1 \end{bmatrix} \tag{21}$$

$$V_{3d} = Q \cdot R \tag{22}$$

$$P_{3d} = R^{-1} \cdot Q' \cdot \Delta z \tag{23}$$

Das Ergebnis ist eine Korrekturfunktion (siehe Abbildung 27) für die z-Komponente in Abhängigkeit von der aktuellen xyz-Position der SpiderCam. Die Berechnungen müssen ebenfalls für die x- und y- Komponente erfolgen. Dann kann nach Gleichung 24 eine korrigierte, genauere Position aus der gemessen Kabellängenposition ermittelt werden. Der Vorteil dieser Methode ist, dass nun auch Korrekturwerte außerhalb des MTS Arbeitsbereichs vorhanden sind. Der Nachteil ist, dass durch Änderungen der Kabellängen (Verzug, Abnehmen der SpiderCam, andere Gewichte, ...) eine neue Kalibrierung und Berechnung der Koeffizienten der Näherungsfunktionen von Nöten ist.

$$\begin{bmatrix} x_{korrigiert} \\ y_{korrigiert} \\ z_{korrigiert} \end{bmatrix} = \begin{bmatrix} x_{kabel} \\ y_{kabel} \\ z_{kabel} \end{bmatrix} + \begin{bmatrix} \Delta x(x_{kabel}, y_{kabel}, z_{kabel}) \\ \Delta y(x_{kabel}, y_{kabel}, z_{kabel}) \\ \Delta z(x_{kabel}, y_{kabel}, z_{kabel}) \end{bmatrix} \tag{24}$$

Stand der Anlage im ersten Quartal 2011

Nach Hallenumbauten im ersten Quartal 2011 für die der CableRobot abgenommen werden musste, funktionierte die Kommunikation zum Mount per Glasfaser nicht mehr. Nach einer ausführlichen Problemanalyse wurde erkannt, dass der Lichtwellenleitereingang des 3Com Switches auf dem Mount einen Wackelkontakt hatte. Dieser wurde auf Garantie ausgetauscht, wodurch der Fehler behoben wurde.

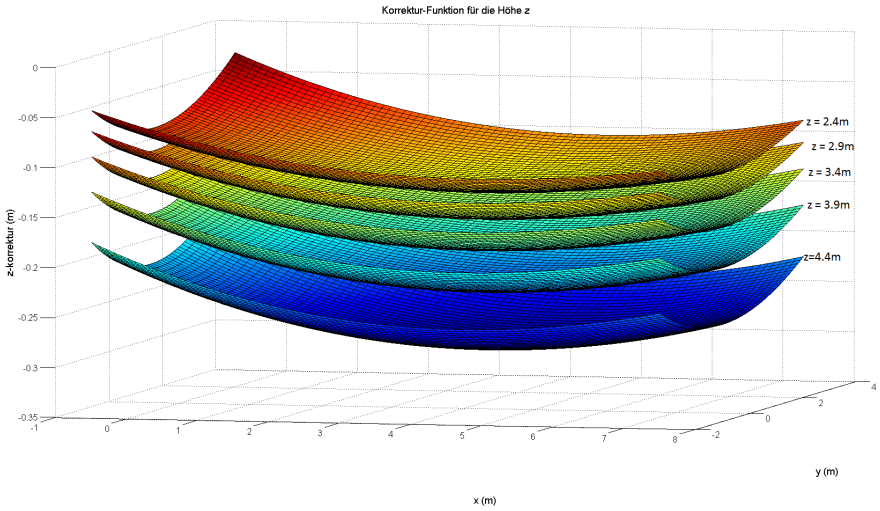


Abbildung 27: *z-Korrektur-Funktion für verschiedene Höhen*

Im zweiten Quartal wurde der Servicer von Astrium angeliefert und gewogen. Es stellte sich heraus, dass Servicer inklusive Sensoren und z-Achse 160 kg wiegt. Nach Rücksprache mit SpiderCam sind die 10 kg Extragewicht kein Problem. Der Servicer wurde an den CableRobot montiert, die fehlenden elektronischen Komponenten eingebaut und die Funktionalität getestet. Durch die größeren Abmaße musste der Arbeitsbereich des CableRobots verkleinert werden. Der Flug über die Empore ist noch in einem kleinen Höhenkorridor von 10 cm möglich.

Der Client wiegt 59 kg und besitzt einen Hebelarm zum TCP des KUKA von 600 mm. Dieser Umstand bereitet vor allem den Achsen A4 und A5 Probleme, da sie mit mehr als 200% überlastet werden. Größtenteils stellt diese Überlast kein Problem da, da der KUKA mit weniger als 5% seiner Maximalgeschwindigkeit betrieben wird und somit die dynamischen Kräfte fast keinen Einfluss haben. Dennoch hat der KUKA von Zeit zu Zeit Probleme, manche Konfigurationen anzufahren. Um diese Fälle zu umgehen, wurde ein alternatives System entwickelt, das die simulierten Bewegungen der Satelliten so auf den Kuka-Arm umsetzt, dass die oben genannten Achsen dabei in Ruheposition verharren. Der Kuka setzt somit nur noch Rotationsbewegungen um, bei denen jedoch automatisch Translationen hinzukommen, die vom CableRobot ausgeglichen werden müssen, was wiederum den Arbeitsraum einschränkt. Dies Methode wird daher nur bei Bedarf eingesetzt.

Vertikale Achse für den CableRobot

Eine zusätzliche vertikale Achse für den CableRobot wurde im ersten Quartal 2011 entwickelt. Der Motor wird zentral über dem CableRobot angebaut, um das Servicer-Mock-up um die vertikale Achse drehen zu können. Der Motor wird über CAN direkt mit dem dSpace Kern gesteuert. Da die Glasfaser zwischen dem CableRobot und dem Leitstand kein CAN Bus unterstützt, werden zwei CAN-Ethernet Converter verwendet.

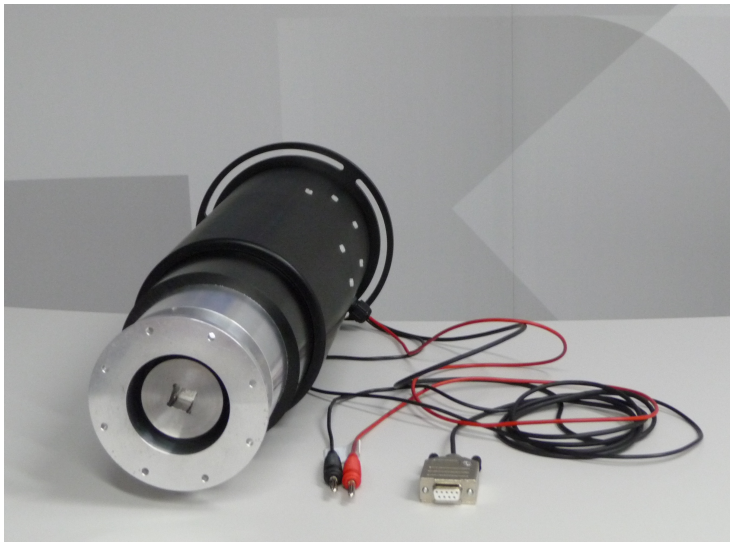


Abbildung 28: *Motor für die Steuerung einer zusätzlichen vertikalen Achse für den CableRobot*

Diese zusätzliche Z-Achse für den CableRobot vergrößert zum einen den Arbeitsbereich. Zum anderen kann die ungewollte Rotation kompensiert werden, die abhängig von der Position des CableRobots ist.

Die Integration des Motors wurde erfolgreich durchgeführt. Der Rotationswinkel wird mittels eines absoluten Encoders gemessen. Die Messgenauigkeit ist $0.035°$. Ein PI-Regler wurde für die Steuerung der vertikalen Achse implementiert. Der Regler ermöglicht eine Rotationsgenauigkeit von $0.1°$ mit einer Geschwindigkeit von $3°/s$. Der Drehbereich der vertikalen Achse liegt zwischen $-50°/s$ und $+50°/s$.

Schätzung der Position und Orientierung des CableRobots

Wie oben bereits erläutert wurde, zeigten die ersten Experimente, dass die Position des CableRobots über Kabellängen allein nicht genau genug ist. Wie bereits erläutert wurde die Lösung verfolgt, die SpiderCam im Sichtbereich des MTS zu bewegen, die Positionsabweichungen aufzunehmen und anhand dieser Kalibrierergebnisse eine Polynom-Näherungsfunktionen für jede xyz-Komponente zu berechnen, die den Abstand zwischen MTS-Messung und Seillängenmessung angibt. Eine Schätzung der Orientierung des CableRobots wurde zusätzlich auch berechnet, da die interne SpiderCam Software keine Angaben dazu macht.

Eine Kalibrierungstrajektorie wurde definiert, um Positions- und nun auch Orientierungsdaten im Arbeitsbereich des CableRobots aufzunehmen. Jede Bahn beschreibt die Bewegung in jeweils nur eine Richtung, während die anderen zwei Richtungen konstant gehalten werden (siehe Abb. 29).

Für die Näherungsfunktion der Position wurde unverändert ein Polynom dritten Grades gewählt. Dieses Polynom nimmt die Position des CableRobots über die Kabellängen-

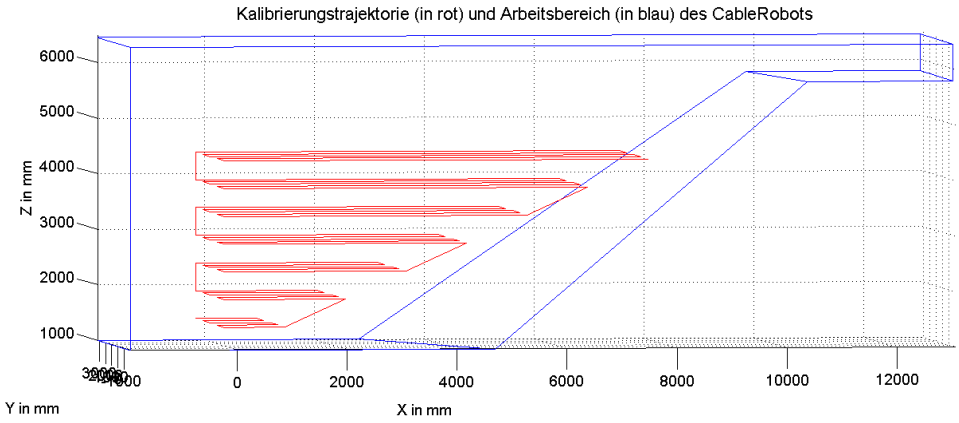


Abbildung 29: *Kalibrierungstrajektorie innerhalb des Arbeitsbereichs des CableRobots*

messung und berechnet den Positionsfehler für jede xyz-Komponente. Die Orientierungsschätzungsfunktion ist auch ein Polynom des dritten Grades. Es nimmt die echte Position des CableRobots und berechnet die xyz-Rotationskomponente.

Die Positionsfunktion wurde im Simulink-Modell als ein Messungskorrektur-Block integriert. Direkt nach dem Eingang der Positionsdaten des CableRobots über den CAN Bus wird die geschätzte Position des CableRobots berechnet. Intern wird dann nur mit der korrigierten Position weiter gerechnet.

Ein Orientierungsschätzungs-Block steht zur Verfügung für das 12D/9D Transformationsmodell. Das Ziel ist die Drehung des CableRobots mit dem Kuka Arm zu kompensieren. Dieser Block wurde getestet aber zum damaligen Zeitpunkt im ersten Quartal 2011 noch nicht integriert.

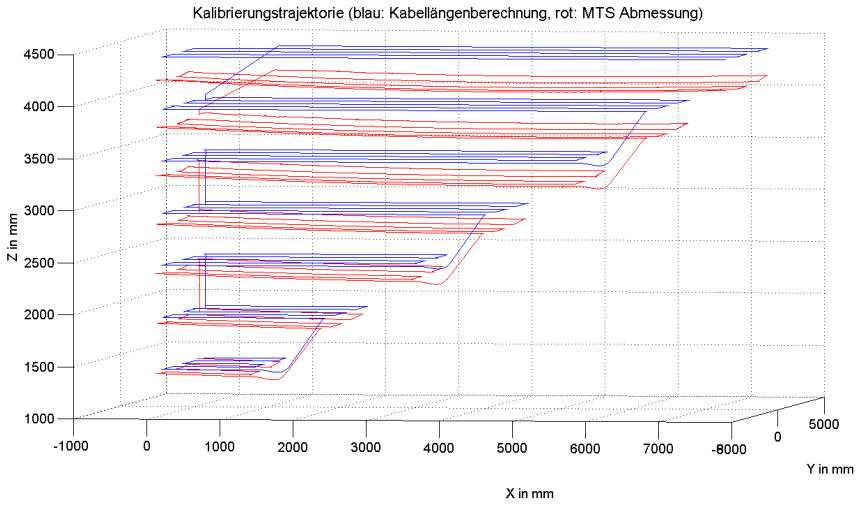


Abbildung 30: *Vergleich der internen Positionsbestimmung der SpiderCam über Kabellängen und der Positionsbestimmung mittels Motion Tracking System*

Abb. 30 beschreibt die Bewegung des CableRobots während der Kalibrierungstrajektorie. Die blaue Position ist über Kabellängen berechnet und die rote Linie ist die durch das MTS aufgenommene Position. Vor allem der Fehler in z-Richtung ist sehr markant.

Die Standardabweichung des Fehlers für die xyz-Position über Kabellängen ist in der Tabelle1 dargestellt. Kleinere Werte für die Näherungsfunktion sind nicht zu erwarten, da ein verbleibender Fehler durch das Rauschen der MTS-Messwerte bleibt.

	Kabellängen	Näherungsfunktion
X	68.10mm	5.13mm
Y	91.15mm	3.82mm
Z	203.03mm	1.59mm
RX		0.064°
RY		0.003°
RZ		0.063°

Tabelle 1: *Standardabweichung des Fehlers für die xyz-Position und die Orientierung*

Anschließend wurde die Trajektorie wiederholt aber mit korrigierter Position (Tabelle2). Die Standardabweichung der Position ist etwas größer für die Höhe z aber ist trotzdem kleiner als 8 mm. Da die Messungen des MTS rauschen ist es schwierig festzustellen, ob die Abweichung von der Ungenauigkeit des MTS herrührt oder von der Ungenauigkeit des Polynoms.

	korrigierte Position
X	5.66mm
Y	4.56mm
Z	7.76mm
RX	0.090°
RY	0.003°
RZ	0.090°

Tabelle 2: *Standardabweichung des Fehlers für die xyz-Position mit der korrigierten Position*

Wie in Abb. 31 dargestellt, decken sich Schätzung und erneute Messwerte der Rotation des CableRobots. Der Fehler bleibt für alle Achsen unter 0.1°.

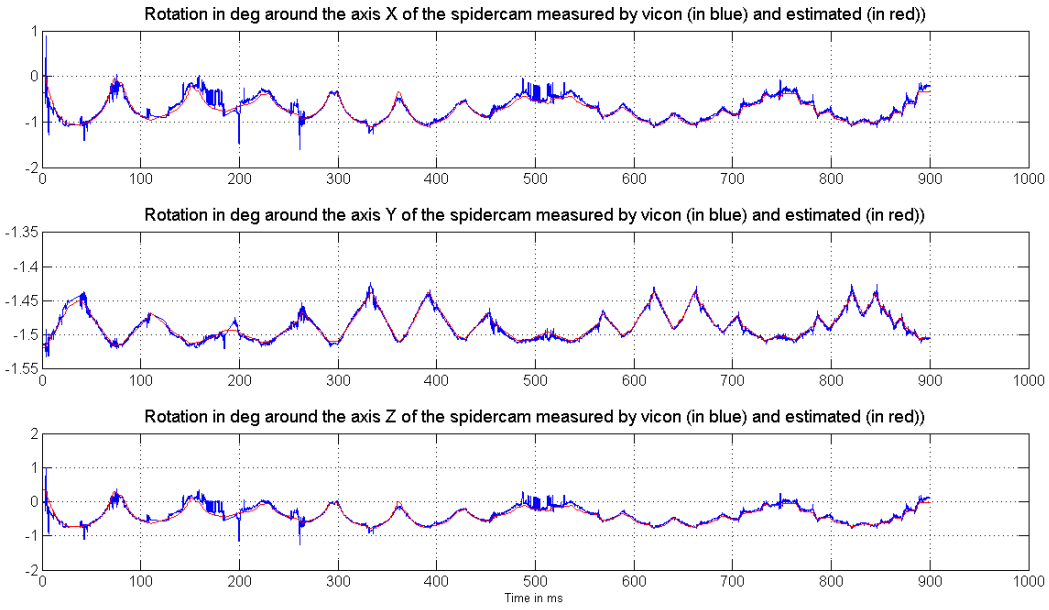


Abbildung 31: *Vergleich der Rotationsbestimmung der SpiderCam durch die Näherungsfunktion (rot) und der Rotationsbestimmung mittels Motion Tracking System (blau)*

3.1.9 AP31225-D: Demo und Verifikation Bewegungssystem

Anlagenkalibrierung

Nach Abschluss der Arbeiten am Bewegungssystem wurde die Genauigkeit der Anlage überprüft. Da das Motion Tracking System für Genauigkeitsangaben im mm-Bereich nicht ausreicht, wurde ein Laser Tracker der Firma API Metrology mit einer Genauigkeit im μm-Bereich getestet. Der Laser Tracker konnte erfolgreich in die bestehende Software integriert werden, so dass dessen Messdaten synchron mit den anderen Messdaten abgespeichert werden können. Es wurde eine Transformation erstellt, die das Laser Tracker Koordinatensystem in das Weltkoordinatensystem mittels vier bekannter Punkte in beiden Koordinatensystemen überführt. Die Funktionalität des Laser Trackers wurde für unsere Zwecke nachgewiesen.

Im nächsten Schritt wurde eine ausführliche Testkampagne zur Bestimmung der Genauigkeit der Anlage durchgeführt. Der Laser-Tracker detektiert ein spezielles, reflektierendes Target mit einer Genauigkeit von 0.007 mm bei 20 m Entfernung. Mit dessen Hilfe wurde zunächst ein neues Weltkoordinatensystem für die Explorationshalle definiert. Danach wurde die Genauigkeit des Kuka-Arms vermessen. Der Cable-Robot und das Motion-Tracking-System der Firma Vicon (MTS) wurden zuerst kalibriert und auf ihre Genauigkeit überprüft.

Neues Weltkoordinatensystem

Für das neue Koordinatensystem wurden fünf Punkte mit dem Laser Tracker eingemessen, welches ungefähr mit dem Kuka-Koordinatensystem übereinstimmt. Vier dieser Punkte liegen direkt auf dem Hallenboden (siehe Abbildung 32) und bilden annähernd einen rechten Winkel. Diese Anordnung ist notwendig, um das Koordinatensystem des MTS auf das Weltkoordinatensystem abbilden zu können. Der fünfte Marker befindet sich im Krater (siehe Abbildung 33), um einen Marker außerhalb der Hallenbodenebene positionieren zu können. Mit diesem und drei der vier weiteren Marker kann eine Transformation zwischen einer frei wählbaren Pose des Laser Trackers in der Halle und dem eingemessenen Weltkoordinatensystem berechnet werden. Gleichungen 25, 26, 27 und 28 beschreiben[3] wie anhand vier Markerpositionen im Weltkoordinatensystem (*a1, b1, c1, d1*) und im gesuchten Koordinatensystem (*a2, b2, c2, d2*) eine Rotationsmatrix *R* und ein Translationsvektor *T* zwischen beiden Koordinatensystemen bestimmt wurden. Die Weltkoordinaten der Messpunkte sind in Tabelle 3 dargestellt.

Angaben in mm	P1	P2	P3	P4	P5
x-Koordinate	1463,5	2310,7	5000,4	1489,5	6685,5
y-Koordinate	-2,7	-5,7	-38,8	2470,6	3344,8
z-Koordinate	26,0	24,3	24,1	20,6	1525,6

Tabelle 3: *Position der eingemessene Punkte im Weltkoordinatensystem*

[3] siehe Point Alignment, http://answers.google.com/answers/threadview/id/556169.html

$$P = \begin{pmatrix} bx1 - ax1 & by1 - ay1 & bz1 - az1 \\ cx1 - ax1 & cy1 - ay1 & cz1 - az1 \\ dx1 - ax1 & dy1 - ay1 & dz1 - az1 \end{pmatrix} \tag{25}$$

$$Q = \begin{pmatrix} bx2 - ax2 & by2 - ay2 & bz2 - az2 \\ cx2 - ax2 & cy2 - ay2 & cz2 - az2 \\ dx2 - ax2 & dy2 - ay2 & dz2 - az2 \end{pmatrix} \tag{26}$$

$$R = P^{-1} \cdot Q \tag{27}$$

$$T = \begin{pmatrix} ax2 & ay2 & az2 \end{pmatrix} - \begin{pmatrix} ax1 & ay1 & az1 \end{pmatrix} \cdot R \tag{28}$$

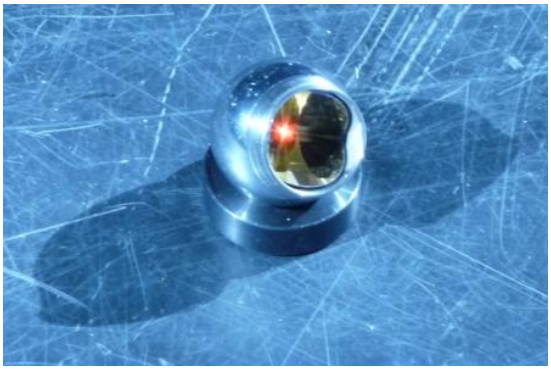


Abbildung 32: *Laser-Tracker-Marker (Radius=25mm) auf dem Boden (Referenzpunkt 1)*

Abbildung 33: *Adapter auf der Mondlandschaft (Referenzpunkt 5)*

Kuka Absolutgenauigkeit

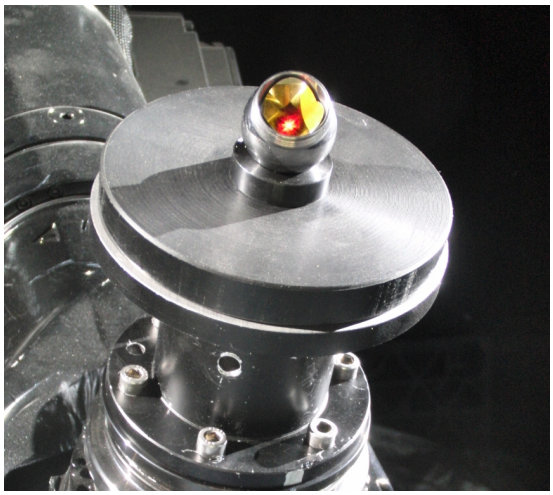


Abbildung 34: *Laser-Tracker-Marker auf dem Kuka TCP*

Die Transformation zwischen Kuka-Koordinatensystem und dem neuen Weltkoordinatensystem (T_{Kuka}^{World}) wurde ebenfalls wie oben beschrieben bestimmt, indem verschiedene Punkte im Arbeitsraum des Kuka angefahren und die angegebene Position des TCP mit der Lasermessung verglichen wurde. Gl. 29 zeigt die feste Transformation (Translation in mm).

$$T_{Kuka}^{World} = \begin{pmatrix} 0.9984 & -0.0027 & -0.0051 & 8.47 \\ 0.0032 & 1 & -0.0035 & 1.33 \\ 0.0064 & 0.0033 & 0.9997 & 3.86 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{29}$$

Danach wurde die Genauigkeit des Kuka-Arms nach der ISO 9283 Norm abgemessen. Ein Laser-Tracker Marker wurde auf dem Kuka TCP befestigt (Abbildung 34) und eine Strecke mit fünf Punkten auf einer Ebene im Arbeitsbereich des Kuka-Roboters definiert. Diese Strecke wurde 10 Mal zurückgelegt, mit einer Pause an jedem Punkt. Für jeden Punkt wurden die mittleren Positionen berechnet und die Absolutgenauigkeit und Wiederholgenauigkeit abgeleitet (siehe Tabelle 4).

Absolutgenauigkeit	Wiederholgenauigkeit
0.14 mm	0.006 mm

Tabelle 4: *Absolut- und Wiederholgenauigkeit des Kuka-Arms*

Cable-Robot- und Motion-Tracking-System-Kalibrierung

Das MTS-Koordinatensystem kann auf das Weltkoordinatensystem abgebildet werden, indem die MTS Reflektorkugeln mit 25 mm Durchmesser auf die dafür vorgesehenen Markierungen befestigt werden. Die Transformation zwischen Cable-Robot- und Weltkoordinatensystem wurde bereits mit einem Laserscanner vermessen. Der Nullpunkt des Cable-Robots wurde nun aber neu auf einen messbaren Punkt definiert. Er befindet sich im Zentrum des Cable-Robots auf Höhe der Oberkante der oberen Mount-Platte.

Eine Kalibrier-Trajektorie, die den gesamten Arbeitsbereich des MTS abdeckt, wurde abgefahren, um Polynome für die Korrekturfunktionen (Cable-Robot und MTS) zu bestimmen. Anschließend wurde die 5-Punkt Genauigkeitstrajektorie abgefahren, um die Absolut- und Wiederholgenauigkeit nach der ISO Norm 9283 zu vermessen. Die Strecke (30 Zyklen) wurde einmal ohne Korrektur und einmal mit Korrektur zurückgelegt. Die Ergebnisse für den Cable-Robot sind in Tabelle 5 dargestellt. Der sehr schlechte Wert für die Absolutgenauigkeit vor der Korrektur ist zum Teil darauf zurückzuführen, dass der Höhenunterschied zwischen wahrem Cable-Robot Nullpunkt und neu definiertem Nullpunkt nicht bekannt war. Dieser wird in Zukunft aber automatisch durch die Korrekturfunktion eliminiert.

Da die Absolutgenauigkeit des MTS kleiner als die des Cable-Robots ist, kann in Zukunft der Cable-Robot mit dem Motion-Tracking-System geregelt und neu kalibriert werden.

	Absolutgenauigkeit	Wiederholgenauigkeit
ohne Korrektur	164.3 mm	7.1 mm
mit Korrektur	10.3 mm	1.2 mm

Tabelle 5: *Absolut- und Wiederholgenauigkeit des Cable-Robots*

	Absolutgenauigkeit
ohne Korrektur	19.8 mm
mit Korrektur	4.6 mm

Tabelle 6: *Absolutgenauigkeit des Motion-Tracking-Systems*

Neue Transformationskerne

In der bisherigen 12D/9D Transformation sind während der Demonstration verschiedener Missionsszenarien verschiedene Probleme aufgetreten, weswegen nach neuen Ansätzen gesucht wurde, die komplexe Bewegung zweier Satelliten im All mit den beschränkten Möglichkeiten in der Explorationshalle abzubilden. Ein Problem ist der begrenzte Arbeitsraum von Kuka und Cable-Robot. Hierzu wurde eine neue 12D/9D Variante entwickelt bei der der Client um eine feste Position gedreht wird während der Servicer alle translatorischen Bewegungen ausführt. Ein zweites Problem stellt die Tragkraft des Kuka KR60 da. Der Client mit seinen 59 kg erfüllt zwar die Maximallast von 60 kg, sein Massenschwerpunkt liegt jedoch zu weit vom TCP entfernt (600 mm statt erlaubten 200 mm). Vor allem Achse A4 und A5 sind dadurch an ungünstigen Positionen zu stark belastet, weswegen der Kuka vor allem bei langsamen Bewegungen zur Sicherheit stoppt. Aus diesem Grund wurde eine 12D/6D Variante entwickelt, die den Ansatz verfolgt die Achsen A2, A4, und A5 fix zu lassen und mit dem Kuka lediglich die gewünschte Orientierung des Servicers mit den Achsen A1, A3 und A6 herzustellen.

12D/9D Transformationskern mit fixem Clienten

Bei dieser Variante wird zunächst aus den bekannten Posen des Servicers $T^{ECI}_{Servicer}$ und Clienten T^{ECI}_{Client} im ECI (Earth-Centered Inertial) Koordinatensystem (siehe Abb. 35) die Pose des Clienten aus der Sicht des Servicers $T^{Servicer}_{Client}$ berechnet (Gl. 30).

$$T^{Servicer}_{Client} = (T^{ECI}_{Servicer})^{-1} \cdot T^{ECI}_{Client} \tag{30}$$

Als nächster Schritt wird festgelegt, wo sich Servicer und Client in der Explorationshalle befinden sollen. Dazu wird der Client zunächst so positioniert, dass der Bewegungsbereich des Kuka möglichst viele Orientierungen erlaubt. Diese Initialposition wird zunächst manuell definiert, kann im späteren Verlauf aber verschoben werden, um den verfügbaren Arbeitsraum optimal zu nutzen.

Die Orientierung des Servicers in der Explorationshalle $R^{World}_{Servicer}$ ist bekannt (siehe

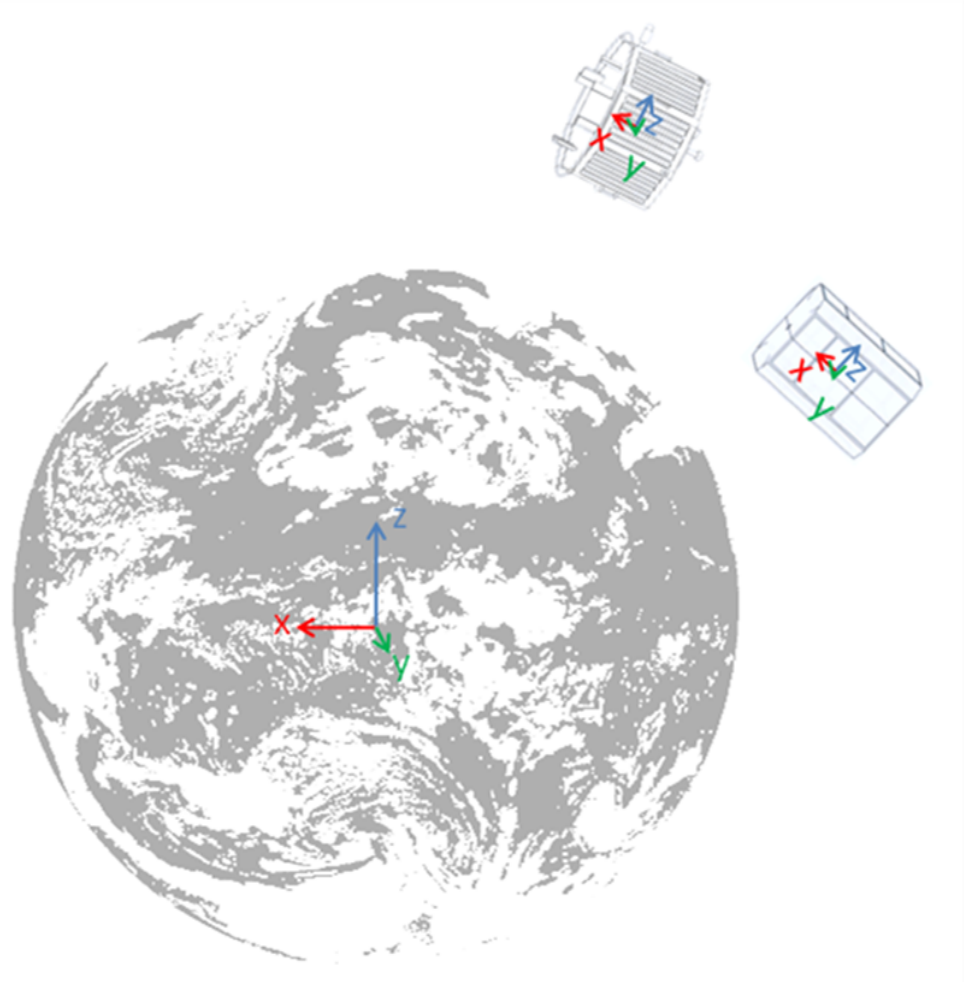


Abbildung 35: *Ausgangspunkt: Servicer und Client im ECI Koordinatensystem*

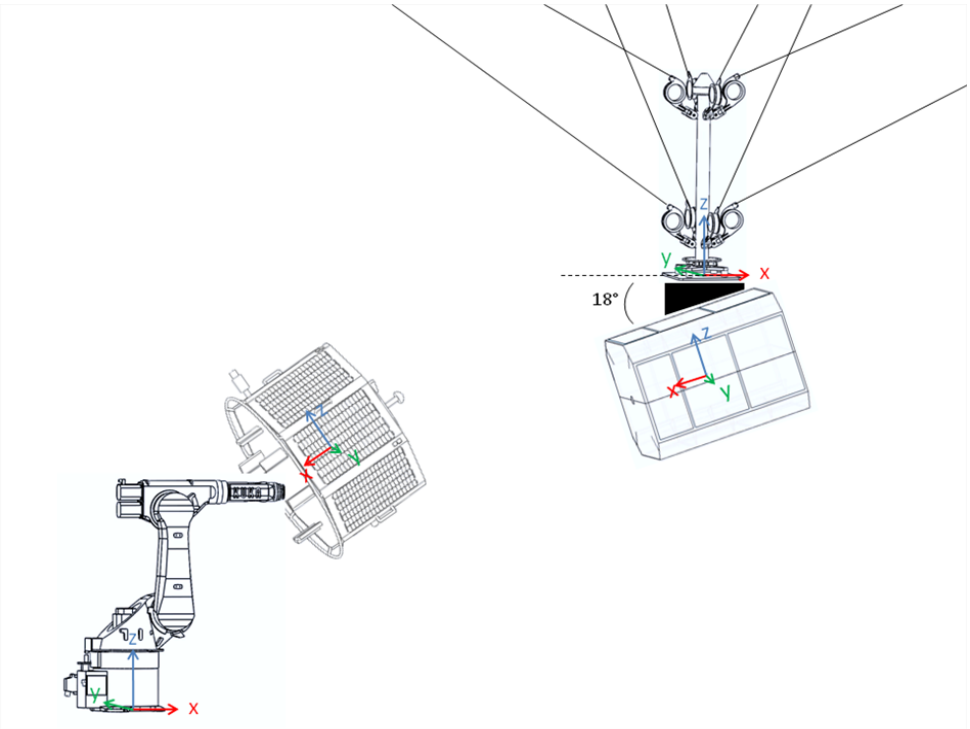


Abbildung 36: *Positionierung der Satelliten in der Explorationshalle* ($Roll = \phi = 0°, Pitch = \theta = 18°, Yaw = \psi = 180°$)

Abb. 36) und kann nach (Gl. 31) berechnet werden (ZYX-Euler Konvention).

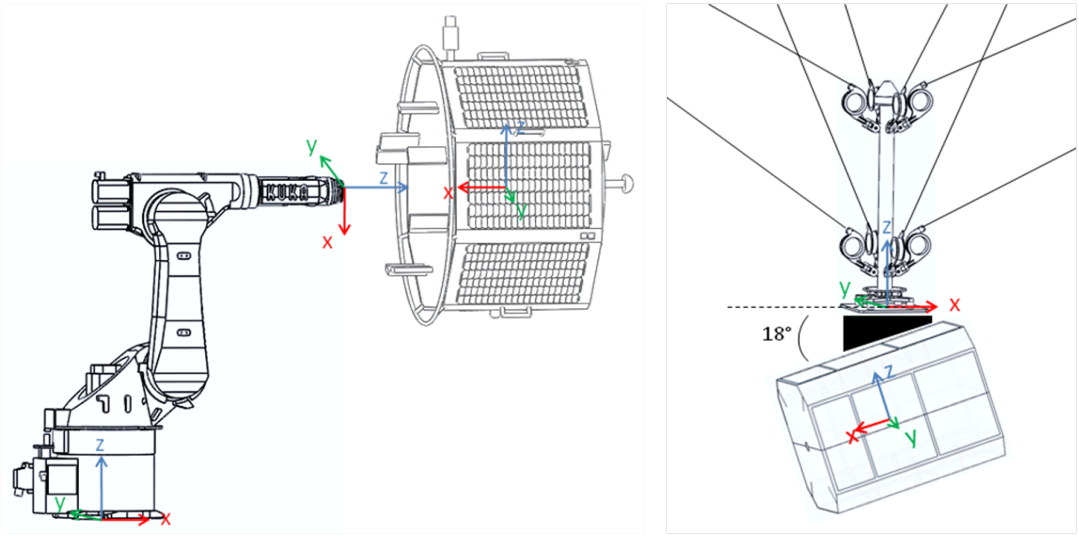
$$R^{World}_{Servicer} = \begin{pmatrix} c\psi \cdot c\theta & -s\psi \cdot c\phi + c\psi \cdot s\theta \cdot s\phi & s\psi \cdot s\phi + c\psi \cdot s\theta \cdot c\phi \\ s\psi \cdot s\theta & c\psi \cdot c\phi + s\psi \cdot s\theta \cdot s\phi & -c\psi \cdot s\phi + s\psi \cdot s\theta \cdot c\phi \\ -s\theta & c\theta \cdot s\phi & c\theta \cdot c\phi \end{pmatrix} \tag{31}$$

Aus der Orientierung des Servicers in der Explorationshalle $T^{World}_{Servicer}$ und der Orientierung des Clienten im Servicer-Koordinatensystem $R^{Servicer}_{Client}$ kann schließlich die Orientierung des Clienten in der Explorationshalle T^{World}_{Client} (Gl. 32) und die Position des Servicers in der Explorationshalle $P^{World}_{Servicer}$ (Gl. 33) berechnet werden.

$$T^{World}_{Client} = R^{World}_{Servicer} \cdot R^{Servicer}_{Client} \tag{32}$$

$$P^{World}_{Servicer} = P^{World}_{Client} - R^{World}_{Servicer} \cdot P^{Servicer}_{Client} \tag{33}$$

Als finalen Schritt müssen für die Anlagensteuerung die Posen vom Kuka TCP und vom CableRobot berechnet werden. Für die Berechnung der Pose des Kuka TCP T^{Kuka}_{TCP} (Gl. 36) im Kuka-Koordinatensystem, welches den Input für die inverse Kinematik liefert, sind nun alle Größen bekannt, bis auf die Transformation zwischen TCP und Client aus Sicht des Clienten T^{Client}_{TCP}. Diese konnte jedoch anhand von CAD Daten bestimmt und anhand der ZYX-Euler Konvention als Transformationsmatrix dargestellt werden (Gl. 34) (Translation in mm) (siehe Abb. 37(a)).



(a) Client und Kuka TCP ($Roll = \phi = 0°, Pitch = \theta = 90°, Yaw = \psi = 180°$)

(b) Servicer und Cable-Robot

Abbildung 37: *Transformation zwischen Anlagenkomponente und Satelliten Mockup*

$$T_{TCP}^{Client} = \begin{pmatrix} 0 & 0 & -1 & 600 \\ 0 & -1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{34}$$

$$T_{TCP}^{Kuka} = (T_{Kuka}^{World})^{-1} \cdot T_{Client}^{World} \cdot T_{TCP}^{Client} \tag{35}$$

Die Position des Cable-Robots P_{SC}^{World} (siehe Abb. 37(b)) berechnet sich aus der Position des Servicers $P_{Servicer}^{World}$, der Höhe des Servicers $H_{Servicer}$ multipliziert mit der eingestellten Neigung ($Pitch = \theta = 18°$) und dem Offset zwischen Cable-Robot-Nullpunkt und Servicer Anflanschpunkt.

$$P_{SC}^{World} = P_{Servicer}^{World} + H_{Servicer} \cdot \begin{pmatrix} cos(\theta) \\ 0 \\ sin(\theta) \end{pmatrix} + \begin{pmatrix} offset_x \\ 0 \\ offset_z \end{pmatrix} \tag{36}$$

12D/6D Transformationskern

Wie schon erwähnt wurde die 12D/9D-Transformation entwickelt, um eine zu hohe Belastung der Kuka-Gelenke zu vermeiden, welche durch die Last des Clienten entsteht. Betroffen sind hierbei lediglich die Achsen vier und fünf, deren Arbeitsbelastung (It-Fehler) zeitweise bis auf 200 Prozent der Maximalbelastung ansteigt. Die Achsen zwei und drei sind hiervon kaum betroffen, da deren Hebel weit weniger ungünstig liegt. Achse eins und sechs werden von diesem Problem erst gar nicht tangiert. Die Lösung muss also eine Entlastung der Achsen vier und fünf bieten. Hierbei ist zu erwähnen

dass bedingt durch die Kuka-Kinematik diese beiden Achsen in einer definierten Winkelstellung ebenfalls von der Last des Clienten nicht tangiert werden. Aufgrund dieser Erkenntnisse wurde folgende Konvention getroffen: 1. Die Achsen vier und fünf werden in eine Winkelstellung gebracht, in der sie von der Gravitationslast des Clienten befreit sind und dort determiniert. 2. Die Achse zwei wird in eine beliebige günstige Winkelstellung gebracht, in welcher sie dem Kuka den größten Arbeitsraum lässt. 3. Die Achsen eins, drei und sechs werden angesteuert und sind fortan für die Lage des Clienten verantwortlich. Die Position des Clienten steht somit in direkter Abhängigkeit von der Lage des Clienten, was einen relativen Positionsfehler zwischen Client und Servicer verursacht. Um diesen Fehler eliminieren, wird die Differenz aus reell erreichter Position und eigentlich erwarteter Position (je nach Berechnungsmethode verschieden) gebildet und invers auf die SpiderCam gegeben. Anders ausgedrückt wird die SpiderCam den Positionsfehler des Kukas fortan korrigieren. Mit dieser Methode erhält man ein System, welches durchgehend eine zu hohe Belastung jeglicher Gelenke vermeidet.

Nachteil dieses Systems ist allerdings der nun gestiegene Arbeitsraumbedarf, einerseits vom Kuka, welcher seine Position mit jeder Lageänderung korrigieren wird und andererseits von der SpiderCam, welche den Positionsfehler korrigieren muss. Eine Lösung dieses Problems kann die Nutzung der Achse zwei sein, welche nicht von einer Überlastung betroffen ist und in diesem System ein redundantes Gelenk darstellt. Diese Achse kann somit für eine übergeordnete Positionsänderung vom Clienten und des CableRobots sorgen. Gerät z. B. eines der beiden Objekte an die obere Grenze des jeweiligen Arbeitsraumes, so würde sich der Kuka mit Hilfe von Achse zwei absenken und mehr Spielraum schaffen. Gerät hingegen eines der beiden Objekte an den unteren Rand, so würde sich der Kuka aufstellen. Eine adaptive Regelung der Achse zwei kann das System somit sehr leistungsfähig halten und ohne mechanische Korrekturen das Gewichts- und Überlastungsproblem lösen. Weitere positive Effekte wären eine vereinfachte numerische Berechnung, da keine Sechs-Achsen-Invers-Kinematik mehr nötig wäre und eine kinematische Kollisionsvermeidung, da ein Zusammenstoßen zwischen Client und Kuka sehr unwahrscheinlich, wenn nicht unmöglich wird. Nachteilig wäre allerdings, das die Z-Stabilisierung der SpiderCam auf der Empore nicht mehr benutzbar wäre.

Verifikation der Transformationskerne

Um die 12D-9D Transformationen zu prüfen und eventuell Fehler oder andere Effekte aufzudecken wurden zwei Closed-Loop-Verifikationen als Simulink-Modell entwickelt.

Zum einem werden alle Transformationen beginnend bei den Positionen der Satelliten im Weltraum bis zu den dafür notwendigen Positionen des Kukas und des CableRobots durchgeführt und anschließend die Transformationen invertiert und zurück gerechnet. Die sich daraus ergebene Relativpose zwischen Servicer und Client wird mit der der Originaldaten verglichen. Mit diesem Modell können die Transformationskerne erfolgreich mathematisch getestet und die Effekte der Interpolation sowie der numerischen Fehler könnten analysiert werden.

Dabei wurde entdeckt, dass die Transformation zwischen Welt- und Kuka-Basis-

Koordinatensystem nicht exakt bestimmt wurde. Der Fehler war auf die experimentelle Bestimmung dieser Transformation zurückzuführen (ungenaue Messwerte), weswegen diese wiederholt werden musste. Alle anderen Fehlerquellen sind vernachlässigbar gering.

Ein zweites Closed-Loop-Verifikations-Modell wurde für die Hardware-Verifikation verwendet. Dieses nimmt während der Hardware-Simulation die gemessene Position des CableRobots und die Gelenk-Winkel von Kuka und berechnet daraus die Position von Servicer und Client in der Explorationshalle. Die Ergebnisse werden ebenfalls mit den originalen Daten verglichen. In diesem Fall kann man den Einfluss der Regelung oder die Ungenauigkeit der Systeme erkennen. Nach den ersten Experimenten wurde daraufhin der Regler des Kukas verbessert. Die Geschwindigkeitsbegrenzung des Reglers war in manchen Fällen zu niedrig und die Drehung einiger Achsen war dadurch zu langsam. Der Regler wurde angepasst, wodurch nun alle Winkel die nominalen maximalen Geschwindigkeiten während einer Trajektorie erreichen können. Für die manuelle Fahrt wird weiterhin der alte Regler verwendet, der die Geschwindigkeit so begrenzt, dass der Operator stets eingreifen könnte.

Realisierbarkeit der GNC Trajektorien

Als Vorbereitung für die Tests im Closed-Loop wurden die Missionen im Open-Loop getestet. D.h. die im GNC berechneten Trajektorien wurden abgeflogen, alle Schnittstellen mit Daten versorgt, aber ohne aktives Eingreifen des Servicers aufgrund von potentiellen Sensordaten. Für jede Trajektorie wurde bestimmt, welche 12D9D-Transformation am geeignetsten ist. Dabei wurde die beste Konfiguration der Anlage ermittelt, die einen möglichst großen Teil der Trajektorie ohne Umkonfiguration erlaubt. Der Arbeitsraum des CableRobots und des KUKAs waren dabei maßgeblich die beschränkenden Faktoren. Hinzu kam das zu hohe Gewicht des Clienten, welches durch die Pappwände zusätzlich erhöht wurde. Dadurch lassen sich theoretisch machbare Lösungen nicht in der Halle realisieren, da entweder Achse A4 oder A5 zu hohe Belastungen vor allem bei langsamen Bewegungen hat. Eine genaue Analyse der einzelnen Missionen findet sich in Tabelle 7 und Tabelle 8. Dabei ist der CableRobot mit CB abgekürzt, p dessen Position im Weltkoordinatensystem und γ der Winkel des CableRobots um dessen z-Achse.

Die Missionen entsprechen verschiedenen Raumfahrt-Manövern. Die Mission 101 simuliert ein Station-Keeping, wobei der Client lagestabilisiert sich in nahezu konstanter Entfernung zum Servicer befindet. Die Missionen 102 bis 104 bilden auch ein Station-Keeping Szenario ab, aber der Client taumelt in verschiedenen Drehraten. Die Mission 105 simuliert einen geradlinigen Anflug, wobei der Client wieder lagestabilisiert ist. In den Missionen 106 bis 108 taumelt der Client während sich der Servicer auf ihn zu bewegt. Die Mission 109 ist ein Docking und die letzten vier Missionen sind ein Fly-Around ohne und mit Taumelbewegung des Clients.

Für die Abschlussexperimente muss das Bewegungssystem so genau wie möglich kalibriert sein, damit die Fehler basierend auf der Anlagengenauigkeit minimiert werden, um somit eine Analyse der eigentlichen Experimentierergebnisse und den Closed-

Tabelle 7: *Realisierung der GNC trajektorien*

Mission	Konfiguration	Zeitinter-vall in s	Ursache Start	Ursache Ende	Bemerkung
101	12D/6D A2=-134° InvWrist=Nein A6=0° γ=0°	0-600 von 600	-	-	funktioniert komplett
102	12D/6D A2=-134° InvWrist=Nein A6=-360° γ=0°	0-300 von 600	-	Grenze CR	würde komplett funktio-nieren, wenn Abstand zwischen Servicer und Client 1,0 m kleiner ClientFix könnte theo-retisch komplett funktio-nieren, aber Client zu schwer
103	12D/6D A2=-134° InvWrist=Nein A6=-360° γ=0°	0-300 von 600	-	Grenze CR	würde komplett funktio-nieren, wenn Abstand zwischen Servicer und Client 1,0 m kleiner ClientFix könnte theo-retisch komplett funktio-nieren, aber Client zu schwer
104	12D/6D A2=-134° InvWrist=Nein A6=-360° γ=0°	0-300 von 600	-	Grenze CR	würde fast komplett funktionieren, wenn Ab-stand zwischen Servicer und Client 1,8 m kleiner ClientFix könnte theo-retisch fast komplett funktionieren, aber Client zu schwer
105	12D/6D A2=-134° InvWrist=Nein A6=0° γ=0°	0-1000 von 1000	-	-	funktioniert komplett
106	12D/6D A2=-68° InvWrist=Nein A6=-360° γ=-9°	30-800 von 800	Grenze Kuka A6	-	Umkonfiguration wäre nötig für vollständige Trajektorie

Tabelle 8: *Realisierung der GNC trajektorien*

Mission	Konfiguration	Zeitinter-vall in s	Ursache Start	Ursache Ende	Bemerkung
107	12D/6D A2=-65° InvWrist=Nein A6=-360° γ=0°	171-711 von 800	-	-	ClientFix könnte theo-retisch mehr abfahren, aber Client zu schwer
108	12D/9D ClientFix p=[1,80 0,0 2,05] InvWrist=Nein A4=-360° A6=-360° γ=0°	140-687 von 600	-	-	Umkonfiguration wäre nötig für vollständige Trajektorie
109	-	-	-	-	wurde nicht getestet, da Docking nicht phy-sisch im Demonstrator betrachtet wird
110	12D/9D ClientFix p=[1,45 -1,4 1,95] InvWrist=Nein A4=0° A6=0° γ=4°	0-2000 von 2000	-	-	funktioniert komplett
111	-	-	-	-	noch nicht getestet, da A6 sehr viele Umdrehun-gen benötigt und somit nur kleine Abschnitte ge-zeigt werden können.
112	-	-	-	-	noch nicht getestet, da A6 sehr viele Umdrehun-gen benötigt und somit nur kleine Abschnitte ge-zeigt werden können.
113	-	-	-	-	noch nicht getestet, da A6 sehr viele Umdrehun-gen benötigt und somit nur kleine Abschnitte ge-zeigt werden können.

Loop Betrieb zu erlauben. Auf Grundlage vorangegangener Experimente wurde bereits ersichtlich, dass das Motion Tracking System (MTS) nur durch Erhöhung dessen Abdeckungsbereichs eingesetzt werden kann. Aus diesem Grund wurden im ersten Quartal 2012 die Kameras neu positioniert und ausgerichtet sowie eine weitere MTS Kamera integriert. Da diese MX-T160 Kamera von Vicon die 16-fache Auflösung bietet, konnte damit zeitgleich die Genauigkeit verbessert werden.

Für das Prüfen der Genauigkeit wurden erneut mit einem submillimeter-genauen Laser Tracker der Kuka, der CableRobot und das MTS vermessen. Da der Client auf dem Kuka nicht mit MTS Markern ausgestattet werden kann, weil diese die Messung des LIDARs unbrauchbar machen würden, muss der Client Open-Loop bewegt werden, wodurch dessen Positioniergenauigkeit von der Genauigkeit des Kuka Manipulators abhängt. Aus diesem Grund wurden vor der Laser Tracker Kalibrierung die Achsen des Kukas durch Fachkräfte neu justiert. Mit dem Laser Tracker wurde dann das Koordinatensystem des Kukas eingemessen.

Danach wurde nach der Norm ISO 9283 die Wiederhol- und Absolutgenauigkeit gemessen, indem eine 5-Punkt-Trajektorie abgefahren wurde und die Koordinaten der Eckpunkte mit den Messungen des Laser Trackers verglichen wurden. Die Ergebnisse können Tab. 9 entnommen werden. Allerdings musste aufgrund der Laser Tracker Markerplatzierung auf Volllast verzichtet werden. Außerdem wurde die Verfahrgeschwindigkeit auf die in den Missionen zu erwartende Geschwindigkeit gedrosselt.

Durch die Erbnisse früherer Experimente war bereits ersichtlich, dass Positions- bzw. Rotationskorrekturfunktionen die Genauigkeit des CableRobots und des MTS verbessern können, da beide Systeme eine relative hohe Wiederholgenauigkeit besitzen. Deshalb wurde zunächst eine neue Kalibrierungstrajektorie erstellt, die aufgrund der neuen MTS Konfiguration mehr Arbeitsbereich des CableRobots abdeckt. Außerdem wurde ein Trajektoriengenerator verwendet, der ruck- und beschleunigungsbegrenzte Trajektorien generiert, um somit ein Aufschwingen des CableRobots beim Anfahren und Verlassen von Wegpunkten stark zu reduzieren (Abb. 38).

Abb. 39 zeigt die Kalibriertrajektorie und die Messergebnisse des Laser Trackers. Da die Abweichung pro Durchlauf nahezu konstant ist, konnte ein Korrekturpolynom für die Position und Rotation des CableRobots und für die Positionsbestimmung des MTS berechnet werden. Anschließend wurde anhand der Norm ISO 9283 die Wiederhol- und Absolutgenauigkeit des CableRobots und des MTS mit und ohne Korrekturfunktion gemessen. Die Ergebnisse zeigen (Tab. 9), dass die Kalibrierung die Genauigkeit des CableRobots deutlich erhöht. Im Vergleich zur früheren Laser Tracker Kampagne ist deutlich, dass bessere Korrekturfunktionen berechnet werden. Die MTS Genauigkeit ohne Korrektur hat sich ebenfalls verbessert. Allerdings hat sich die Genauigkeit des CableRobots ohne Korrektur verschlechtert, was darauf hindeutet, dass der CableRobot sich im Laufe der Zeit durch Temperaturschwankungen, Notabschaltungen oder Gewichtsänderungen verändert und somit die Kalibrierung nach und nach an Relevanz verliert. Deswegen muss das MTS als ständige Positionsmessung des CableRobots in Betracht gezogen werden. Damit beschränkt sich die Genauigkeit der Anlage auf die des MTS.

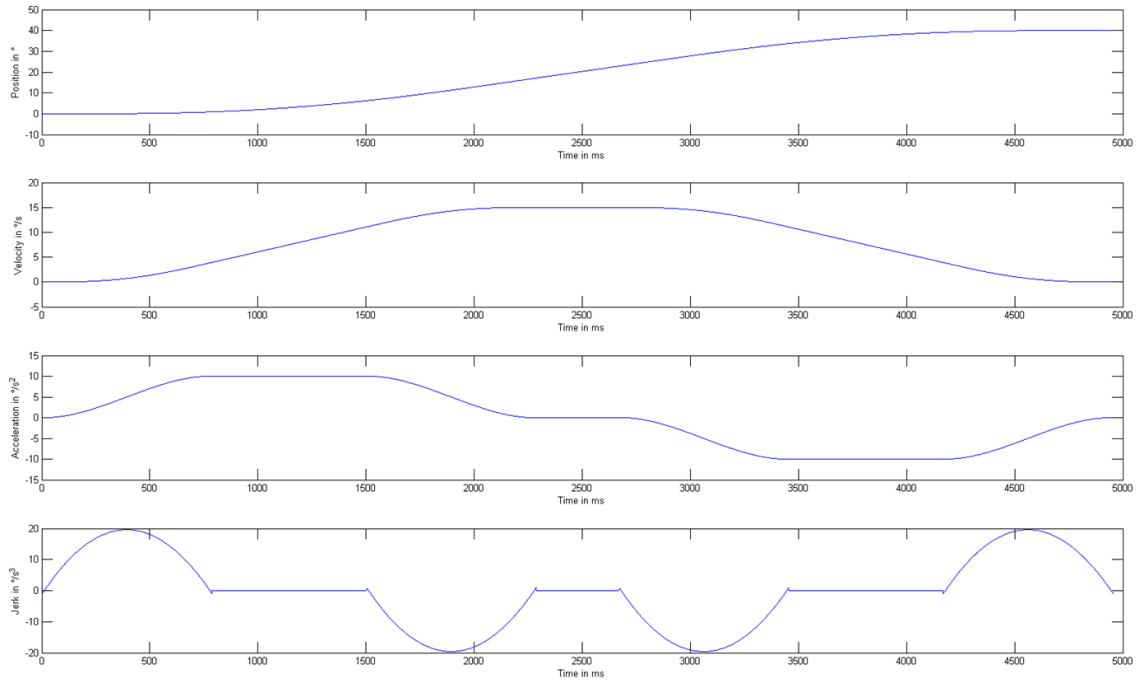


Abbildung 38: *Beispielbewegung des Trajektoriengenerators - Ruck und Beschleunigung sind begrenzt, welches ein sanftes Anfahren und Bremsen gewährleistet.*

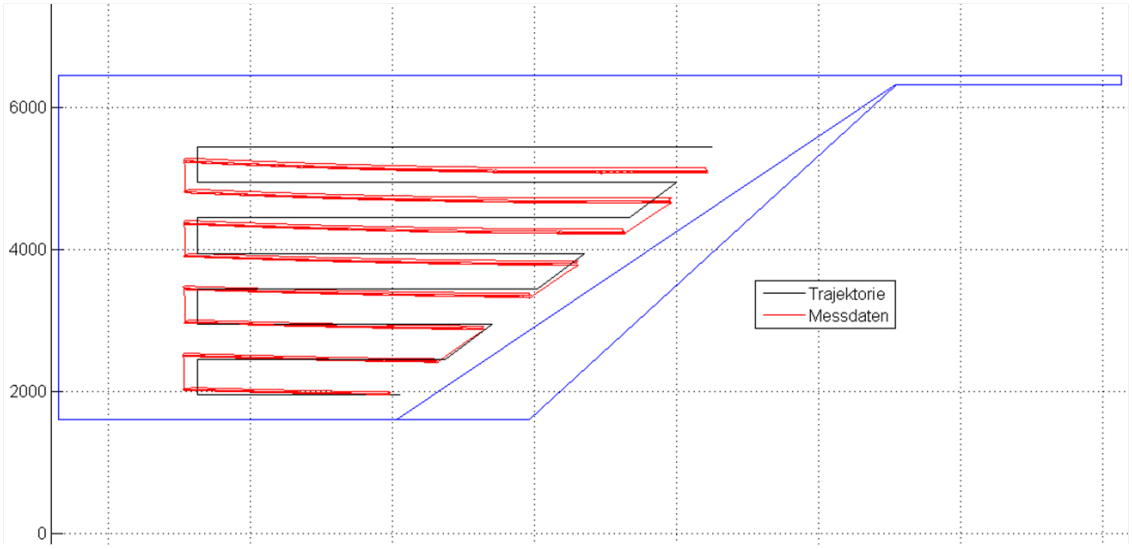


Abbildung 39: *Kalibriertrajektorie und Messergebnisse*

	Absolutgenauigkeit	Wiederholgenauigkeit
KUKA (ohne Korrektur)	1.6 mm,	0.01 mm
CableRobot ohne Korrektur	267.4 mm	1.1 mm
CableRobot mit Korrektur	6.6 mm	1.1 mm
MTS ohne Korrektur	8.9 mm	0.6 mm
MTS mit Korrektur	5.3 mm	0.6 mm

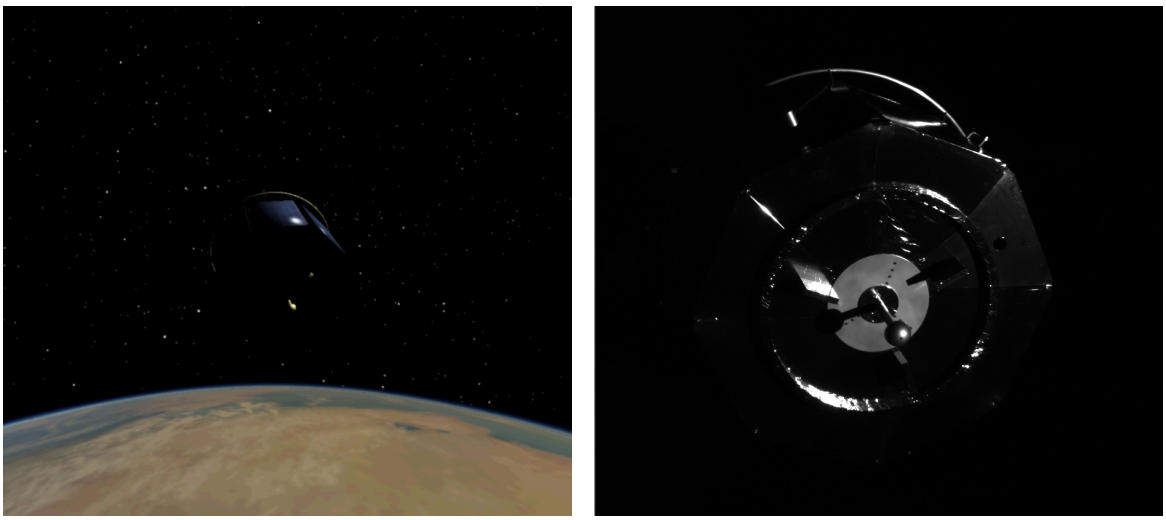
Tabelle 9: *Absolut- und Wiederholgenauigkeit des Cable-Robots*

3.1.10 AP31234-D: Demo Lageschätzung/Prädiktion/Abstandsregelung

In 2011 fand die Anbindung der Astrium-Simulationskomponenten an das DFKI Simulationssystem statt. Erst nachdem dies und die Anbindung der Servicer-Rechner an das Gesamtsystem abgeschlossen waren, konnte das autonome Verhalten des Servicers im Closed-Loop-Betrieb getestet werden. Durch die nicht sicher vorhersehbaren Trajektorien, die dabei entstehen können, waren hierzu jedoch Sicherheitsvorkehrungen und umfangreiche Vorbereitungen erforderlich.

In diesem Arbeitspaket wurden alle Voraussetzungen geschaffen, um die Visual Navigation (VN) und Guidance Navigation Control (GNC) mit dem Bewegungssystem zu testen. Bevor die Experimente dazu starten konnten, musste durch qualitative Vergleiche verifiziert werden, ob Realität in der Explorationshalle und Simulation gut genug übereinstimmen.

Als erstes wurden die gleichen Trajektorien sowohl mit Hardware in der Halle als auch im Astrium Simulator abgeflogen. Von beiden wurden Videos mit den echten Kameras bzw. den simulierten Sensoren erstellt und miteinander verglichen. Dies gab zum einen die Möglichkeit grobe Fehler, wie unterschiedliche Koordinatensysteme, Transformationen oder Dimensionen zu erkennen. Zum anderen konnte überprüft werden, ob die Konfiguration und alle Parameter für beide Teststände identisch sind. Abb. 40 zeigt beispielsweise, dass sich die Kamerakonfiguration in der Simulation von der realen unterschied, wodurch der Client in der Simulation weiter entfernt zu sein scheint.



(a) Simulation (Rechte Kamera im VESTA Viewer)

(b) Realität (Rechte Kamera des Closed Range Stereokamerasystems)

Abbildung 40: *Qualitativer Vergleich zwischen Simulation und Realität*

Als nächster Schritt wurde der LIDAR auf dem Servicer aktiviert und verschiedene Experimente durchgeführt, um die Güte des Scans zu maximieren.

Als zweiter Schritt konnte der Scan mit der Ground Truth verglichen werden. Dafür wurden Servicer und Client in einem konstanten Relativlage gehalten, die einen kom-

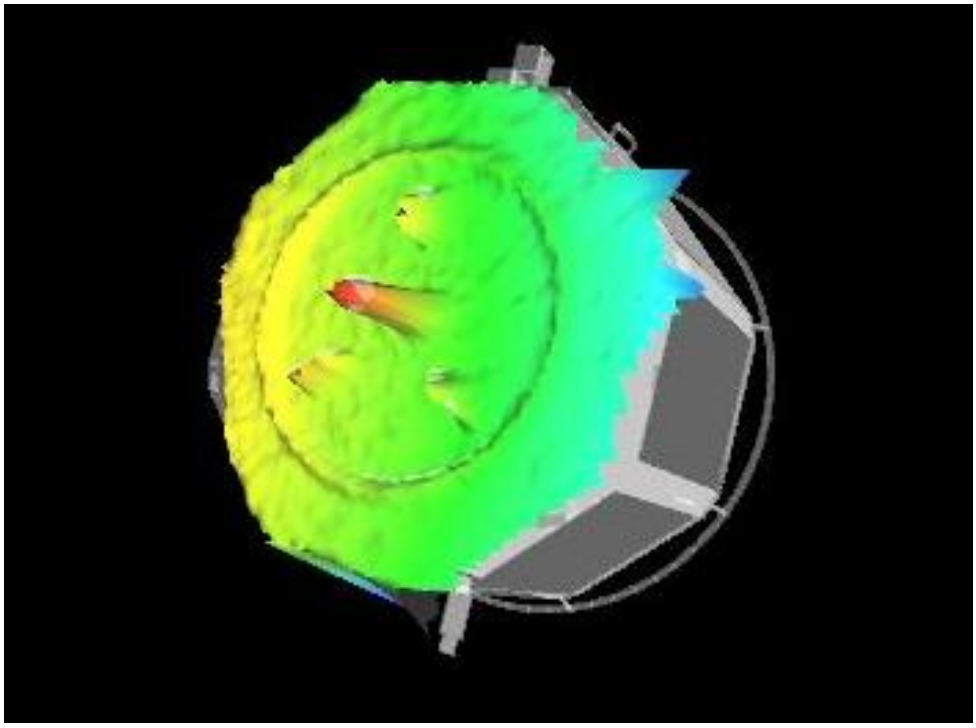


Abbildung 41: *Vergleich zwischen LIDAR Scan und Ground Truth Daten (Modell)*

pletten Scan des Clienten erlaubt. Das Ground Truth Modell, was mit dem LIDAR Scan theoretisch übereinstimmen müsste, konnte aus den aktuellen Posen des KUKA und des CableRobots sowie mit Hilfe der CAD Daten des Clienten berechnet werden.

Abweichungen zwischen Modell und Scan konnten in einem dritten Schritt minimiert werden, in dem die Pose des LIDARs in der Software leicht variiert wird, sodass Modell und Scan in Deckung gebracht werden. Durch diesen dritten Schritt konnte die Lücke zwischen Soll- und Istpose des Sensors geschlossen werden. Wie Abb. 41 zeigt, konnten LIDAR Scan und Sollpose des Clienten qualitativ relativ gut in Deckung gebracht werden.

Um jetzt quantitative Ergebnisse für die VN und das Verhalten der GNC erzeugen zu können, sollten Trajektorien verschiedener Missionen wiederholt abgeflogen werden. Es wurden vorher außerhalb dieses APs alle verfügbaren Missionen absolviert und auf ihre Darstellbarkeit in der Explorationshalle geprüft. Im Zuge dieses Arbeitspakets wurden dann die Missionen ausgewählt, die die aussagekräftigsten Ergebnisse liefern können. Ausgewählt wurden:

- Mission 101: Station Keeping in einer Distanz von 8 m zum Untersuchen der Lageschätzung bei konstantem Abstand und ohne Rotation.
- Mission 104: Station Keeping in einer Distanz von 8 m während der Client in verschiedenen Drehraten taumelt. Es soll untersucht werden ob die Drehung des Clienten die Lageschätzung beeinflusst.
- Mission 108: Annäherung von 8 m auf 3.5 m und Einlauf in stationäre Phase, wäh-

rend der Client in verschiedenen Drehraten taumelt. Diese Mission soll zusätzlich den Einfluss des Relativabstandes auf die Lageschätzung beurteilen. Da diese Mission die wichtigsten Aspekte beinhaltet, wurde sie als Standardtest ausgewählt.

- Mission 113: Fly Around des Servicers um den Clienten während der Client in verschiedenen Drehraten taumelt. Dieses Szenario soll zeigen, wie performant die Lageschätzung beim Übergang von der Front- zur Seitenansicht des Clients ist.

3.1.11 AP31333-D: Demo und Verifikation SOM

Analog zu AP31234-D fand in 2011 die Anbindung der Astrium-Simulationskomponenten an das DFKI Simulationssystem statt. Erst nachdem dies und die Anbindung der Servicer-Rechner an das Gesamtsystem abgeschlossen waren, konnte das autonome Verhalten des Servicers im Closed-Loop-Betrieb getestet werden. Durch die nicht sicher vorhersehbaren Trajektorien, die dabei entstehen können, waren hierzu jedoch Sicherheitsvorkehrungen und umfangreiche Vorbereitungen erforderlich.

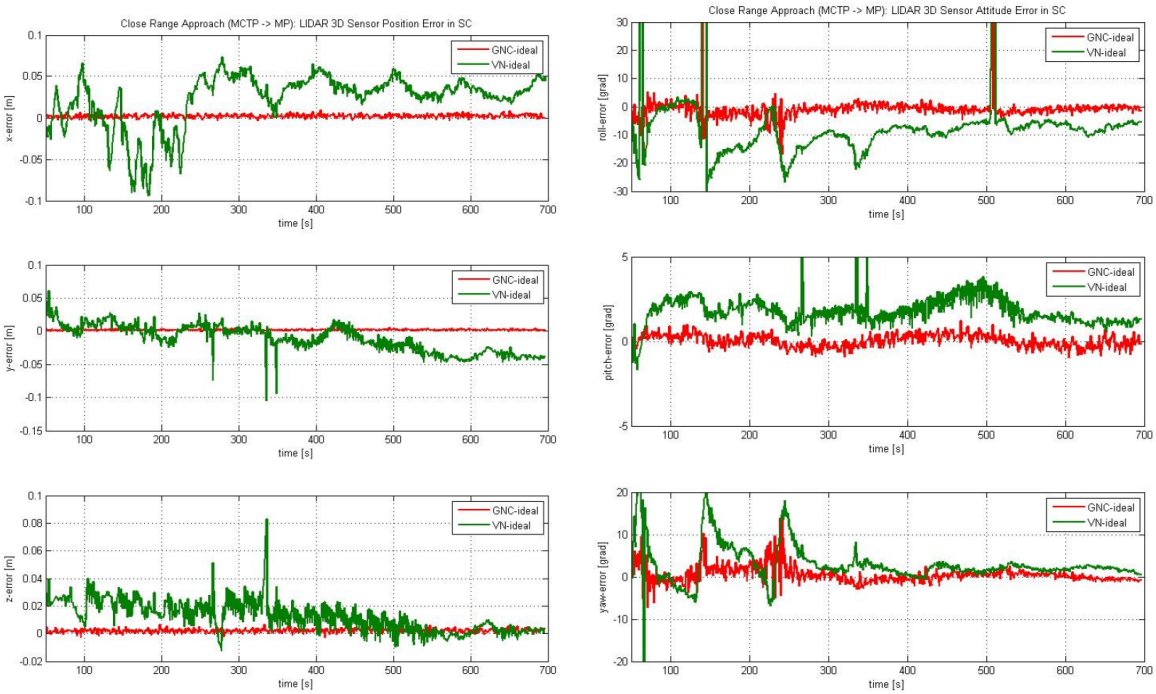
Das Bewegungssystem konnte im ersten Quartal 2012 für Open- und Closed-Loop Tests seitens Astrium genutzt werden. Open-Loop bedeutet in diesem Zusammenhang, dass die Guidance Navigation Control (GNC) von Astrium eine Trajektorie generiert, welche einer möglichen Weltraummission entspricht. Diese wird an den DFKI Kern geschickt, der die generierten Posen von Client und Servicer in der Explorationshalle mit Hilfe des Kuka und CableRobots in die Realität umsetzt. Der auf dem Servicer mitfliegende LIDAR übermittelt während der Mission Messdaten zurück an die GNC. Im Closed-Loop Betrieb werden zusätzlich Korrekturbewegungen von der GNC initiiert, die durch Orbitaldynamikberechnungen in Abweichungen von der Ursprungsbewegung resultieren, um somit den Fehler zwischen Soll- und Istposition des Servicers auszuregeln.

Für die folgenden Experimente wurde die Mission 108 ausgewählt bei der sich der Servicer dem Clienten von 8 m bis 3.5 m annähert und dann den Abstand stationär hält. Der Client taumelt währenddessen in verschiedenen Drehraten. So konnte das Verhalten des Gesamtsystems während der Annäherung und in einer stationären Phase analysiert werden.

Im Open-Loop konnte die Lageschätzung des LIDARs analysiert werden. Die Lageschätzung der Visual navigation (VN) zeigt eine quantitative Abweichung von wenigen cm. Die Abweichung ist am Anfang der Trajektorie am größten (siehe Abb. 42), da zum einen die Messgenauigkeit des MTS dort an der Grenze des Messbereiches am niedrigsten ist und zum anderen sich vor allem Fehler in der Rotationsmessung aufgrund des hohen Abstands stark bemerkbar machen. Außerdem verringert sich die Auflösung des LIDARs mit wachsendem Abstand. Nach Erreichen der stationären Phase stellt sich ein Abstandsfehler von ca. 5 cm (x-Komponente) und ein lateraler Fehler von 4 cm (y-Komponente) ein, während der Fehler in z-Richtung eine Potenz niedriger ist.

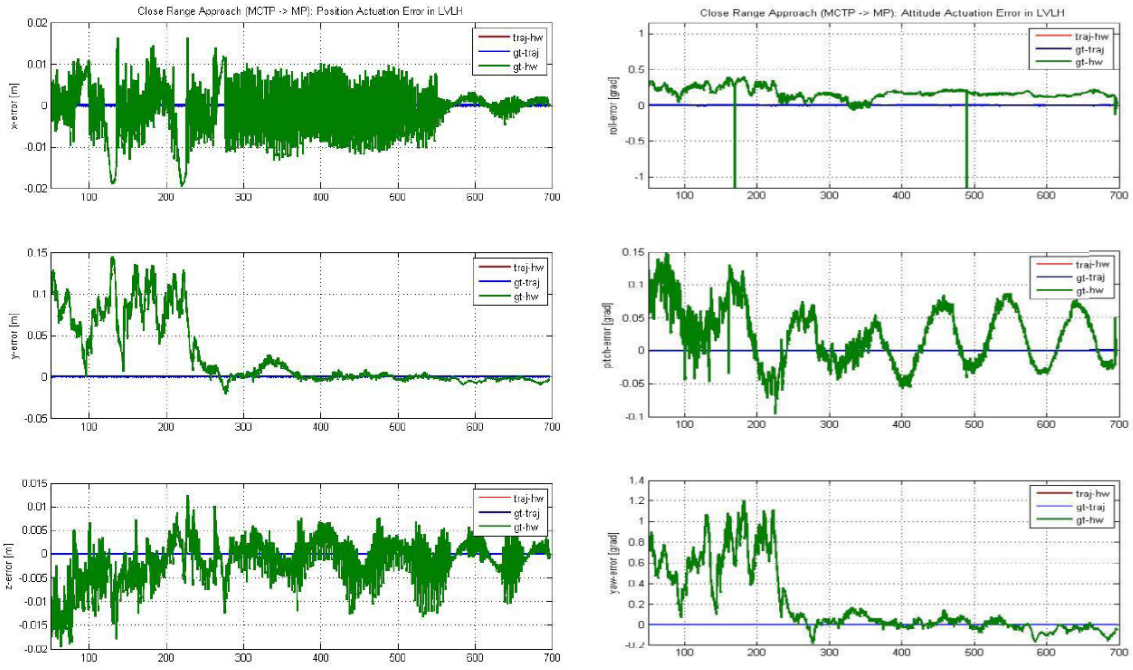
Der Vergleich zwischen der Soll- und Istrelativpose (gemessen anhand der Kuka Winkel und der Kalman-gefilterten CableRobot Pose) (Abb. 43) spiegelt den Verlauf wieder. Anlagenseitig spielen hier Fehler durch das Open-Loop Positionieren des Clienten und teilweise auch des Servicers eine Rolle, da hier beispielsweise Ungenauigkeiten der Mockups oder deren Verzug nicht detektiert werden.

Der Fehler zwischen Ground Truth und LIDAR Daten ist hinreichend klein, um den Closed-Loop Betrieb gefahrlos zu aktivieren. Abb. 44 zeigt in blau den Navigationsfehler aufgrund der LIDAR Daten, der versucht wird auszuregeln. Das System ist stabil, ein Restfehler bleibt jedoch. Trotzdem gilt dieser Test als Erfolg, da alle Einzelkomponenten erfolgreich zusammen gearbeitet haben und ein autonomes Verhalten des



(a) Position (b) Rotation

Abbildung 42: *Vergleich zwischen LIDAR Messung und Ground Truth*



(a) Position (b) Rotation

Abbildung 43: *Fehler zwischen Soll- und Istpose des Bewegungssystems*

Servicers mit dem Bewegungssystem gezeigt werden konnte.

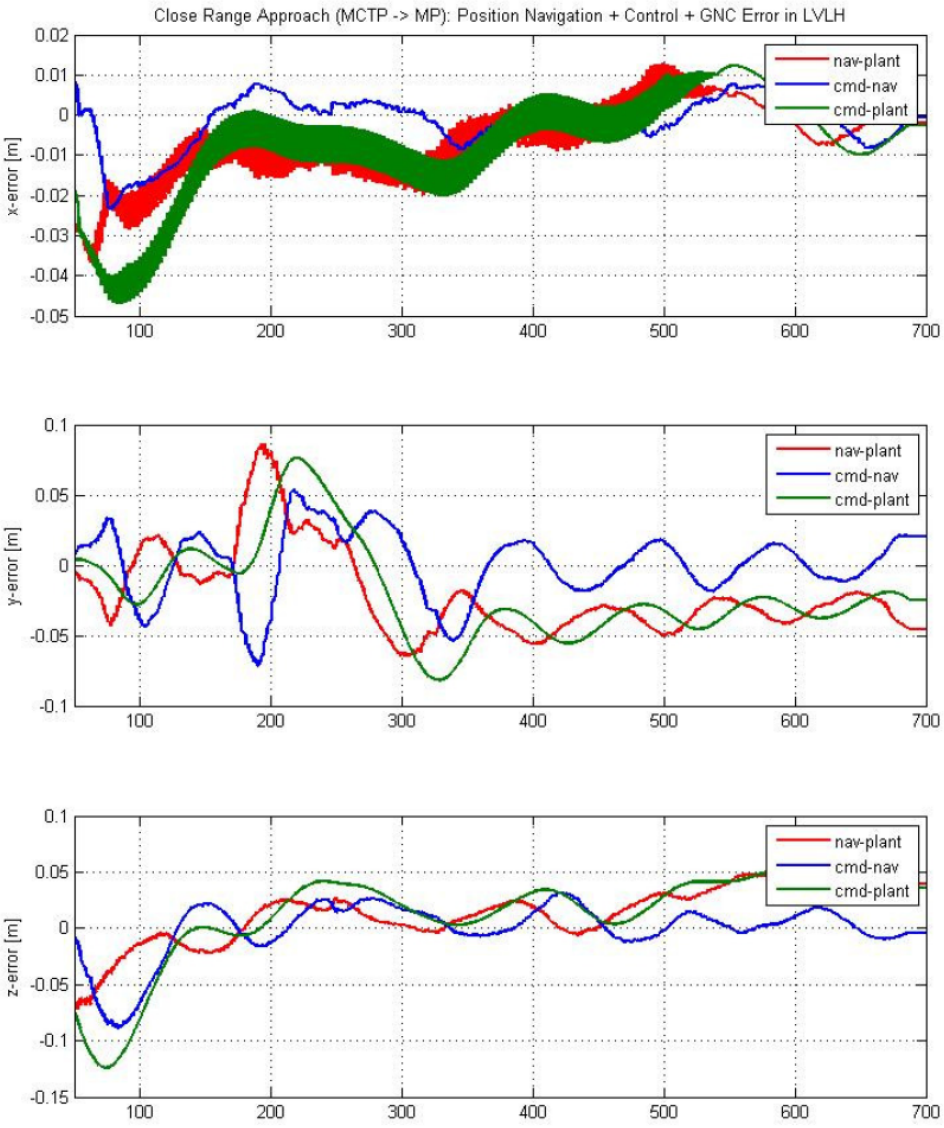


Abbildung 44: *Navigationsfehler (blaue Linie) während des Closed-Loop Betriebs*

3.1.12 AP41100: Spezifikation Kernsystem und Evaluierung vorhandener Systeme

In diesem Arbeitspaket wurde die Spezifikation des Kernsystems erarbeitet.

Vorher wurden die Simulationsysteme COSIMIR und MARS auf ihre Tauglichkeit als Simulationskern untersucht. Neben MARS wurde schließlich auch der VESTA-Viewer von Astrium für Visualisierungen der Simulation eingesetzt, während die Simulationen selbst in speziell entwickelten Matlab-Simulationen auf dem eingesetzten dSPACE-System laufen.

Das Kernsystem wurde so spezifiziert, dass grundsätzlich verschiedene Module mit dem Kern kooperieren können. Durch diese modulare Architektur (siehe Abbildung 45) soll es insbesondere ermöglicht werden, verschiedene Dynamik- und Steuerungssimulationen mit dem Simulationssystem verbinden zu können, um auch in Zukunft flexibel zu sein.

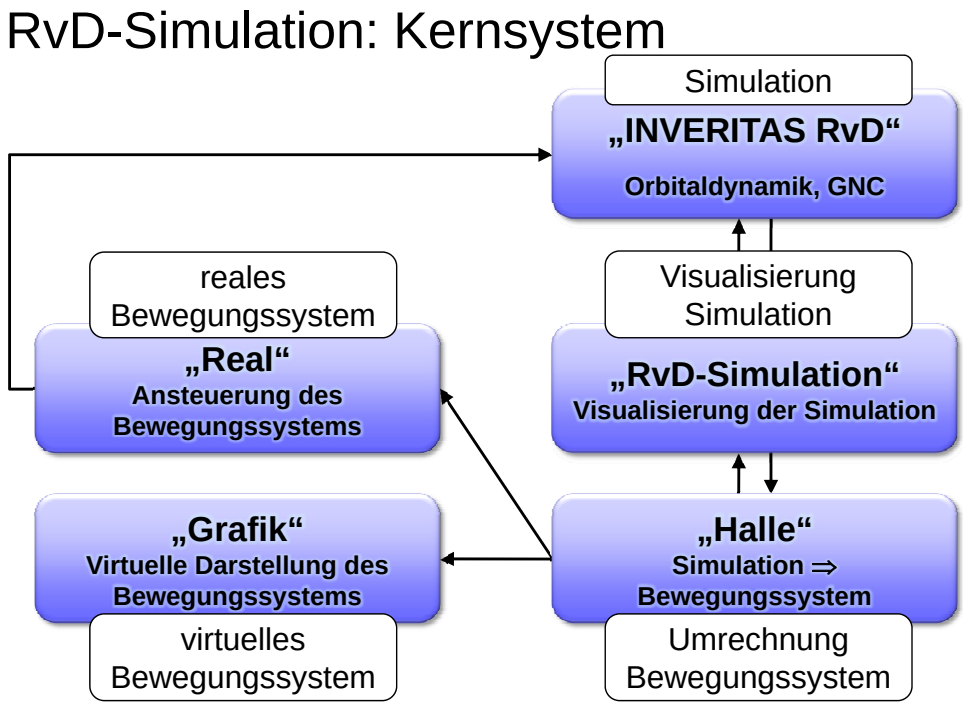


Abbildung 45: *Überblick des Kernsystems*

Das Modul "INVERITAS RvD" simuliert die Steuerung des Chasers (GNC) sowie die Orbitaldynamik der bewegten Objekte. Dies ist der für INVERITAS spezielle Teil des Gesamtsystems. "RvD-Simulation" übernimmt die Daten aus dieser speziellen Komponente und kann den aktuellen Zustand der simulierten Objekte grafisch darstellen. Diese Darstellung kann zur Evaluierung und für virtuelle Sensoren verwendet werden, deren Daten an "INVERITAS RvD" zurückgegeben werden können. Das Modul "Halle" ist daran angeschlossen und übersetzt diese Objektzustände in neue Konfigurationen

der Komponenten des Demonstratorsystems, also des KUKA-Arms und des Kabelroboters. Diese Konfigurationen können einerseits über das Modul "Grafik" dreidimensional dargestellt werden, andererseits auch über das Modul "Real" auf die entsprechenden Hardwarekomponenten übertragen werden, so dass die am KUKA-Arm und an dem Kabelroboter befestigten Objekte die gleiche relative Position und Ausrichtung zueinander einnehmen wie die simulierten Objekte in dem Modul "RvD-Simulation". Die realen Sensoren am Servicer können ihre Daten an das Modul "INVERITAS RvD" übergeben, so dass der Servicer sich entsprechend dieser Sensordaten verhält.

3.1.13 AP41200: Implementierung und Kalibrierung des Kernsystems

Klient

Der eingesetzte Demonstrator für den Klienten in der Hardwaresimulation des RvC-Vorgangs ist der KUKA KR 60-3. Für die Softwaresimulation ist aus diesem Grund eine Nachbildung dieses Manipulatorsystems erforderlich. Dazu wurde zunächst in der Simulationsumgebung MARS ein Massenmodell des KUKA KR 60-3 erstellt. Die notwendigen Maße und Massen der Einzelelemente wurden weitestgehend aus dem dazugehörigen Datenblatt entnommen. Die für die Visualisierung wichtigen Koordinaten der sechs Gelenke sind bekannt, wodurch eine funktionale Nachbildung erreicht werden konnte, die die gleiche Kinematik des realen Systems aufweist. Die Implementierung maßgetreuer CAD-Zeichnungen verbessert die Visualisierung und erhöht damit die Authentizität der Simulation. Aus diesem Grund wurde ein 3D-Modell der Umgebung eingebunden sowie das 3D-Modell des KUKA KR 60-3. Letzteres wurde in grobe Einzelelemente zerlegt, sodass jede primitive Form des Massenmodels durch das entsprechende Teilmodell ersetzt werden konnte. Abbildung 46 zeigt die beschriebene Vorgehensweise und das Resultat.

Für die Simulation ist es von Bedeutung, dass die Position und Orientierung des Endeffektors des KUKA Roboters während jeder beliebigen Achskonfiguration bekannt ist. Deswegen ist die Berechnung der Vorwärtskinematik erforderlich. Hier wurde der Standardansatz mit Hilfe der Denavit-Hertenberg (DH) Parameter gewählt. Dieser ermöglicht eine simple Berechnung der Transformationsmatrix vom Basis- zum Endeffektorkoordinatensystem. Die benötigten DH-Parameter leiten sich aus den festen geometrischen Zusammenhängen der aufeinander folgenden Gelenke und den variablen Gelenkwinkeln ab. Die sich ergebende Transformationsmatrix T_6^0 (4x4) beinhaltet alle rotatorischen und translatorischen Informationen. Während letztere direkt ablesbar sind, müssen die Roll-Nick-Gier-Winkel aus den Elementen berechnet werden. Diese entsprechen ebenfalls den ZYX-Euler Winkeln. Zur Evaluierung des berechneten Ergebnisses wird das Endeffektorkoordinatensystem in die Simulationsumgebung eingezeichnet. Durch die exakte Übereinstimmung mit der Bewegungssimulation ließ sich eine korrekte Berechnung verifizieren.

In der Demonstratorsimulation werden jedoch nicht Gelenkwinkel vorgeben sondern kartesiche Trajektorien, auf denen sich der Klient bewegen soll. Aus diesem Grund müssen aus den Koordinaten geeignete Achskonfigurationen für den KUKA Roboter mit Hilfe einer inversen Kinematik berechnet werden. Hier existieren zwei grundlegen-

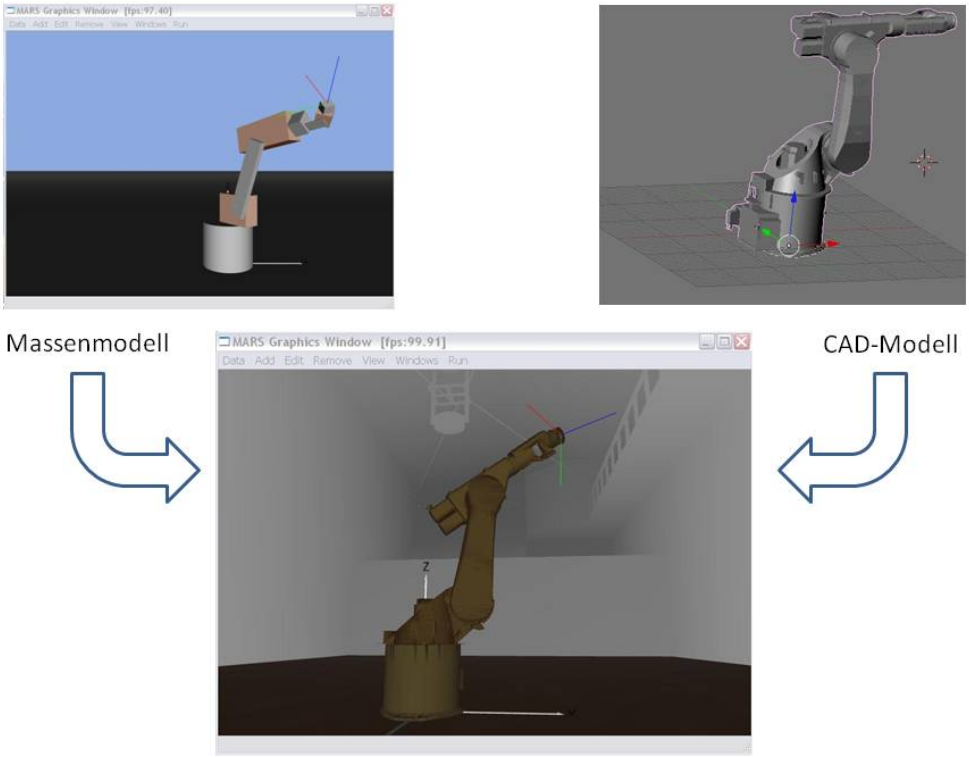


Abbildung 46: *KUKA Visualisierung in MARS*

de Verfahren. Zum einen versucht die numerische Lösung durch iterative Berechnungen von Vorwärtskinematiken mit variierenden Gelenkwinkeln eine Kombination zu finden, die einen minimalen Fehler zur vorgegeben Position und Orientierung aufweist. Diese Variante ist jedoch rechenintensiv und relativ ungenau. Zum anderen existieren analytische Methoden, die die Gelenkwinkel direkt berechnen und damit genauer sind sowie weniger Rechenzeit benötigen. Hier werden entweder durch algebraische Umformungen der DH-Matrizen und Lösung der entstehenden Gleichungssysteme Gelenkwinkel berechnet oder geometrische Ansätze verfolgt. Aufgrund der Vorteile wurde nach einer analytischen Lösung gesucht. Zunächst wurden Lösungswege verfolgt, die eine generelle Lösung für alle Roboter mit sechs Freiheitsgraden ergeben sollten. Keine der Methoden konnte jedoch bis zum Ende nachverfolgt werden. Aus diesem Grund wurde nach speziellen Lösungen für ähnliche Roboter gesucht, um diese für den KUKA KR 60-3 zu adaptieren. Die genauere Untersuchung verschiedener Lösungswege zeigte jedoch, dass algebraische Lösungen spezielle Sonderfälle in den Gleichungen ausnutzen, die nicht auf die DH-Konfiguration des KUKA KR 60-3 übertragbar sind. Zuletzt wurde eine inverse Kinematik für den KUKA KR 3 gefunden, die geometrisch die passende Achskonfiguration ermittelt. Hier war eine Adaption an die Geometrie des Kuka KR 60-3 möglich. So wurde ein geometrischer Sachverhalt aufgestellt, der die Lösung der ersten drei Gelenkwinkel ermöglicht, die im Wesentlichen für die Position des Endeffektors verantwortlich sind. Die passende Orientierung wird durch die letzten drei Gelenkwinkel bestimmt. Da die Achsen dieser Gelenke sich schneiden, gleichen sie den ZYZ-Euler Winkeln und können anhand der Rotationsmatrix von der Orientierung des Schnittpunktes der Gelenke zur Endeffektororientierung berechnet werden. Die Anzahl der möglichen Achskonfigurationen, die durch die inverse Kinematik errechnet werden, hängt von der Positions- und Orientierungsvorgabe ab. So kann eine bestimmte Vorgabe vom KUKA Roboter gar nicht oder mit maximal 32 verschiedenen Achskonfigurationen erreicht werden.

Servicer

Die Bewegung des Servicers wird durch einen Kabelroboter der Firma SpiderCam demonstriert (folgend als Kabelroboter, CableRobot oder auch SpiderCam bezeichnet). Das System wird anhand von acht Winden, welche die Seillängen variieren, betrieben. Da Seilkräfte in MARS nicht ohne weiteres simuliert werden können, wurde die SpiderCam in der Simulation an einen Portalkran gehängt, der alle translatorischen Bewegungen ausführen kann. Dargestellt wird dieser nicht. Die Seile werden nur eingezeichnet und besitzen kein physisches Modell. An der SpiderCam befindet sich optional der Leichtbauarm KUKA LBR-4. Für diesen 7-gelenkigen Roboterarm wurde ähnlich der Vorgehensweise beim KUKA KR 60-3 eine direkte und inverse Kinematik entwickelt und implementiert. Zur Darstellung des Servicers wurden ebenfalls CAD-Modelle der Systemkomponenten eingebunden.

Gesamtsystem

Die Explorationshalle wurde ebenfalls in der Simulation nachgebildet. Dadurch sind die Grenzen der Anlage ersichtlich und geben der Simulation mehr Authentizität. Abbildung 47 zeigt die fertiggestellte Visualisierung der INVERITAS Hardware.

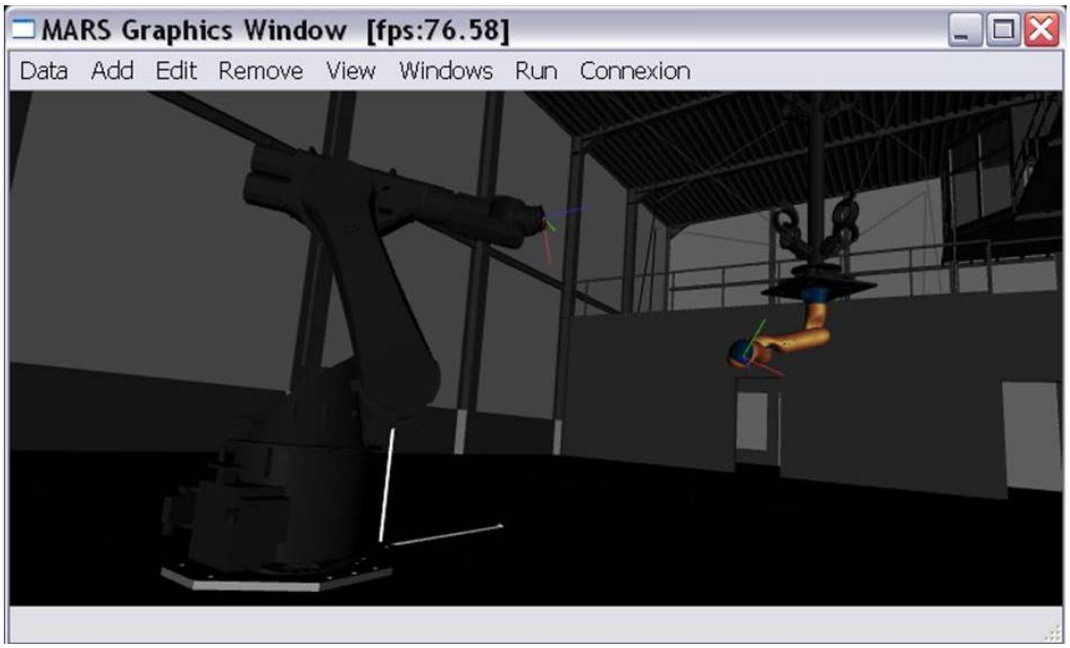


Abbildung 47: *Simulation der INVERITAS Demonstratoren in MARS*

Das Programm zur Berechnung der direkten und inversen Kinematik wurde als Plugin für MARS erstellt und in C++ mit QT geschrieben. Zusätzlich wurde eine GUI erstellt die das Steuern des KUKA KR60-3, des KUKA LBR und der SpiderCam in der Simulation ermöglicht. Neben den Vorgaben der Gelenkwinkel können auch kartesische Positionen und Orientierungen eingegeben werden, die die Roboter in eine Bewegung umsetzen. Zusätzlich können Parameter eingestellt werden, die die Auswahl der gewünschten Achskonfiguration erlaubt. Im Standardfall wird die Achskonfiguration eingenommen, welche die geringsten Gelenkwinkeländerungen nach sich zieht. In der GUI ist außerdem die Möglichkeit gegeben, einen definierten Endpunkt anzugeben, der auf einer bestimmten Trajektorie erreicht werden soll. Ebenso werden die aktuellen Gelenkwinkel des Roboters sekündlich angezeigt.

Bewegungsabbildung vom realen Raum (12D) auf die Langstrecken-Simulationsanlage (9D)

Als zentrale Komponente des Kernsystems wurde im ersten Quartal 2010 ein mathematisches Verfahren entwickelt, welches die Bewegungsabläufe der realen Welt von zwei Objekten mit zwölf unbeschränkten Freiheitsgraden auf die Simulationsanlage mit ihren zum damaligen Zeitpunkt neun, durch die technischen Gegebenheiten der SpiderCam und des KR60 beschränkten Freiheitsgraden abbilden kann. Das Ergebnis dieser Umwandlung ist eine hochgenaue Nachbildung der relativen Lage-

und Positionswerte von Chaser (=Servicer) und Target (=Client) auf der Hardware der Langstrecken-Simulationsanlage. Das Ausgangskoordinatensystem ist hierbei beliebig wählbar und unterliegt keinerlei Einschränkungen. Folgende Funktionalitäten wurden in das System implementiert:

- Eine adaptive Ausrichtung des Explorationshallensystems (Weltsystem). Hierdurch kann das Verfahren ohne weitere Transformation bei beliebiger Ausrichtung des Weltsystems direkt genutzt werden.
- Eine konfigurierbare Ausrichtung der Lage und Position der Messsysteme (Kameras, Lidar, etc.) relativ zur SpiderCam. Hierdurch kann der Anflugwinkel der SpiderCam zum KR60 determiniert und je nach Anwendung angepasst werden, um u. U. einen größeren Arbeitsbereich zu erhalten.
- Einen zusätzlichen Freiheitsgrad um die Z-Achse der SpiderCam. Diese Implementierung kann in den meisten Fällen eine Verringerung der translatorischen Ausgleichsbewegung der SpiderCam bewirken. Sie ist bei Bedarf eliminierbar.
- Eine automatische Eliminierung von fehlerhaften Lageabweichungen der SpiderCam (z. B. durch ungleiche Nutzlastenverteilung oder durch systematische Fehler der Anlage). Diese, wenn auch sehr kleinen möglichen Abweichungen, produzieren, gerade bei großen Entfernungen, einen nicht unbeachtlichen Fehler der relativen Positions- und Lage-Abbildung vom Chaser und vom Target auf das Simulationssystem. Diese Fehler werden hierbei durch eine Positionsverschiebung der SpiderCam und des KR60 heraus gerechnet.
- Einer Verminderung der von der Steuerung generierten hochfrequenten Schwingungen auf der SpiderCam mit Hilfe einer inversen Überführung dieser Schwingungen auf den dynamisch stabileren KR60. Hierdurch wird ein Aufschwingen des SpiderCam-Systems vermieden und die Genauigkeit der gesamten Simulation erhöht.

Das Verfahren basiert bisweilen darauf, dass Position und Lage mithilfe des Bewegungsfolgesystems Vicon direkt an der SpiderCam durch ein eigens zu diesem Zweck angebrachtes Kreuz, welches mit Markern versehen ist, gemessen wird. Eine Rückrechnung der ermittelten Daten des zu simulierenden Chasers (abstrahiert z. B. durch die Messsysteme) findet über die baulich festgelegte Position und Lage der Messsysteme an der SpiderCam relativ zur SpiderCam, über die rotatorische Auslenkung des zusätzlichen Freiheitsgrades an der SpiderCam sowie über die fehlerhafte Lageabweichung der SpiderCam statt und liefert anschließend die Soll-Position der SpiderCam. Es ist hier allerdings ebenfalls möglich die Position der Messsysteme direkt mit Vicon zu bestimmen, wobei hierbei geeignete Markerpositionen Voraussetzung sind. Neben der Position der SpiderCam liefert das System die Winkel des zusätzlichen rotatorischen Freiheitsgrades der SpiderCam sowie die Transformationsmatrix des TCP (Tool Center Point) des KR60. Als reale Eingänge benötigt das System zwei Transformationsmatrizen, eines von jedem Objekt.

Das mathematische Verfahren selbst wurde mit Hilfe der Implementation von zehn Koordinatensystemen entwickelt. Dies begünstigt eine gute Modularität, eine große

Übersichtlichkeit sowie eine hohe Adaptivität des Systems. Die Koordinatensysteme liegen zum Teil Anlagenseitig, zum Teil weltraumseitig. Das Ausgangssystem wird hier als Erdsystem beschrieben und besitzt seinen Koordinatenursprung im Erdmittelpunkt. Eine Ersetzung dieses Systems durch ein beliebig anderes Ausgangssystem ist ohne Einschränkung möglich. Ein wichtiges anlagenseitiges Koordinatensystem erhielt die Bezeichnung Basissystem und beschreibt zwei synchron zueinander rotierende Sphären variablen Ausmaßes, mit dem Mittelpunkt in der Roboterbasis des KR60, auf dessen kleinerer Sphäre der TCP des KR60 verankert ist und auf der größeren das Messsystem der SpiderCam. Die Abbildungen 48 und 49 zeigen eine Übersicht aller Koordinatensysteme. Durch die hohe Adaptivität des Verfahrens lassen sich zukünftig weitere Objekte beliebig in das System einfügen, deren Positionen relativ zum KR60 und zur SpiderCam ebenfalls richtig bestimmt werden. Eine Rotation dieser zusätzlichen Objekte ist allerdings nicht mehr möglich.

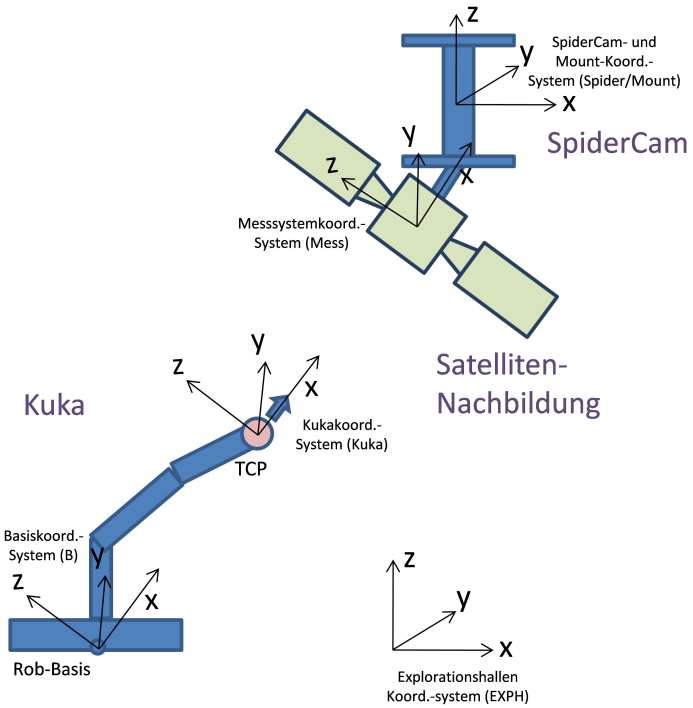


Abbildung 48: *Koordinatensysteme in der Explorationshalle*

Das gesamte mathematische Verfahren wurde in Matlab und Simulink programmiert. Eine Verifizierung des Verfahrens erfolgte im ersten Quartal 2010 durch Simulation. Der Fehler überschritt dabei niemals 10^{-12} Millimeter bzw. Radianten.

Architektur der Steuerungs- und Überwachungssoftware

Eine Software für die Steuerung und Kommunikation zwischen den Systemen wurde einschließlich bis zum zweiten Quartal 2010 entwickelt. Abb. 50 zeigt die Kommuni-

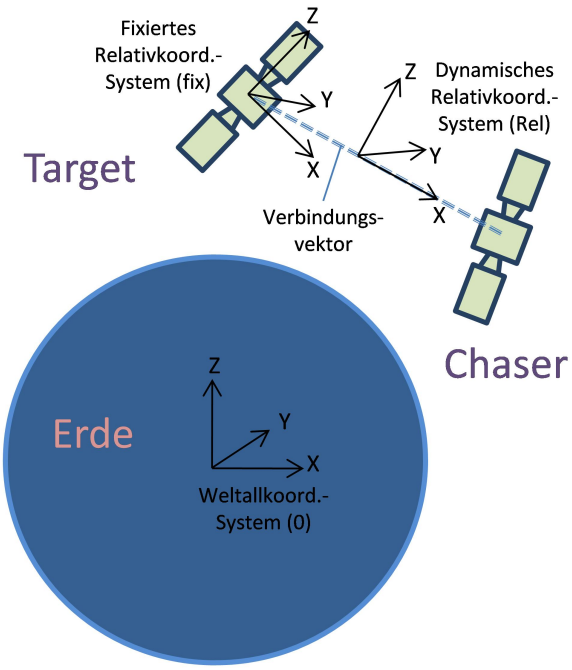


Abbildung 49: *Koordinatensysteme im Erdsystem*

kationen zwischen den Systemen und dem Hauptprogramm. Die Systeme sehen wie folgt aus:

- Motion Tracking System Vicon (MTS). Das MTS sendet alle 4 oder 12 ms die Position und Orientierung der SpiderCam. Die Daten werden über eine Ethernet Verbindung geschickt.
- Cable Robot SpiderCam. Die SpiderCam wird über einen CAN Bus gesteuert. Das Programm erhält alle 4 ms Statusinformationen der SpiderCam und sendet relative Positionänderungen zurück.
- Manipulator KUKA KR-60. Der Manipulator wird über Ethernet mit einem Remote Sensor Interface (RSI) gesteuert. Das Programm liest alle 12 ms die Position und Orientierung des Kuka-Roboters aus einer XML Nachricht und schickt Positions- oder Achskorrekturen zurück.
- Satellitenmodell
- Mars Simulation: Die gesamte Halle inclusive Manipulator und Cable Robot wird in Mars simuliert. Die Konfiguration wird über Ethernet in Echtzeit geschickt.
- Graphische Benutzeroberfläche (GUI) und Steuerungsprogramm

Die graphische Benutzeroberfläche GUI (Abbildung 51), hatte bis zum zweiten Quartal 2010 die folgenden Funktionalitäten:

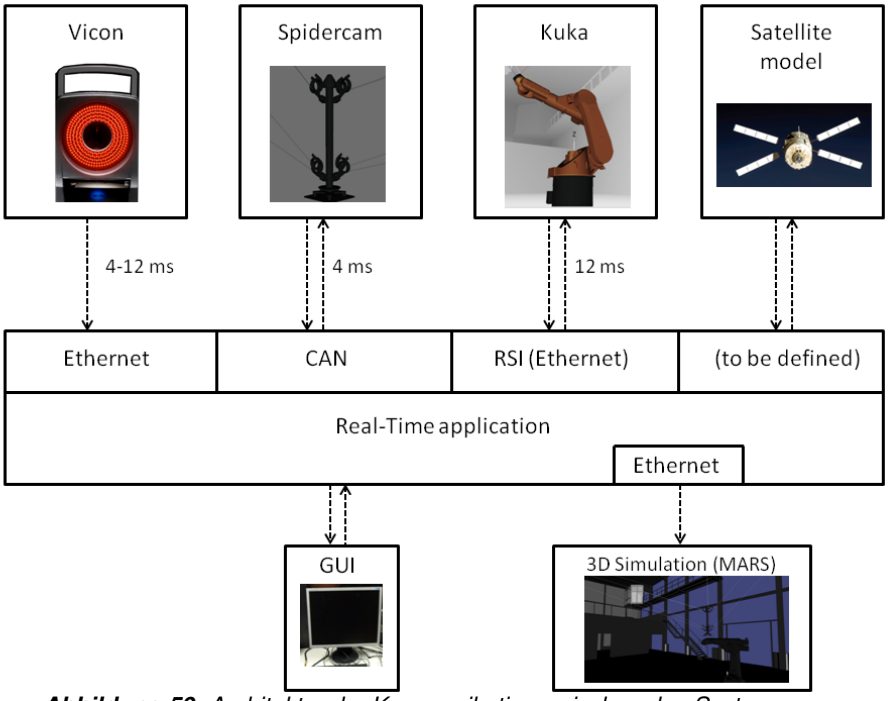


Abbildung 50: *Architektur der Kommunikation zwischen den Systemen*

- Status der Kommunikation zwischen den Systemen. Wurde eine Kommunikation zu einem System erfolgreich hergestellt, wird dieses grün hinterlegt. Wenn zu einem System keine Verbindung besteht, wird dieses rot dargestellt.
- Funktionsablauf des Programms. Nur ein Modus ist erlaubt. PASSIVE MODE ist nur für Überwachung. Mit ACTIVE MODE kann der Benutzer eine Trajektorie fahren (TRAJECTORY MODE), eine einfache Bewegung mit Positions- oder Geschwindigkeitsregelung des Cable Robots (SIMPLE MODE) vollziehen oder eine Interaktion mit der Kuka Bewegung (INTERACTION MODE) durchführen.
- Steuerung des Cable Robots. Der Benutzer des Programms kann die Zielposition für den Cable Robot durch Klicken auf den Lageplan bestimmen (X und Y Wert im Zentrum, Z Wert an der rechten Seite). In schwarz ist die aktuelle Position und in rot ist die Sollposition dargestellt. Die Sollpositionskoordinaten sind auf der linken Seite angezeigt. Die Werte können manuell verändert werden. Die gewünschte Position kann entweder per Positionsregler, per Sollwertgenerator oder per Sollgeschwindigkeitsgenerator erreicht werden. Außerdem kann eine konstante Geschwindigkeit vorgegeben werden.

Erstellung einer Simulationsumgebung für orbitale und lunare Manöver mithilfe von Matlab/Simulink und Mars

Für das Vorbereiten und für ein erfolgreiches Durchführen von Simulationsvorgängen auf der robotischen Langstreckensimulationsanlage, welche die relativen Bewegungsvorgänge weltraumgebundener Manöver physisch simulieren wird, ist es notwendig,

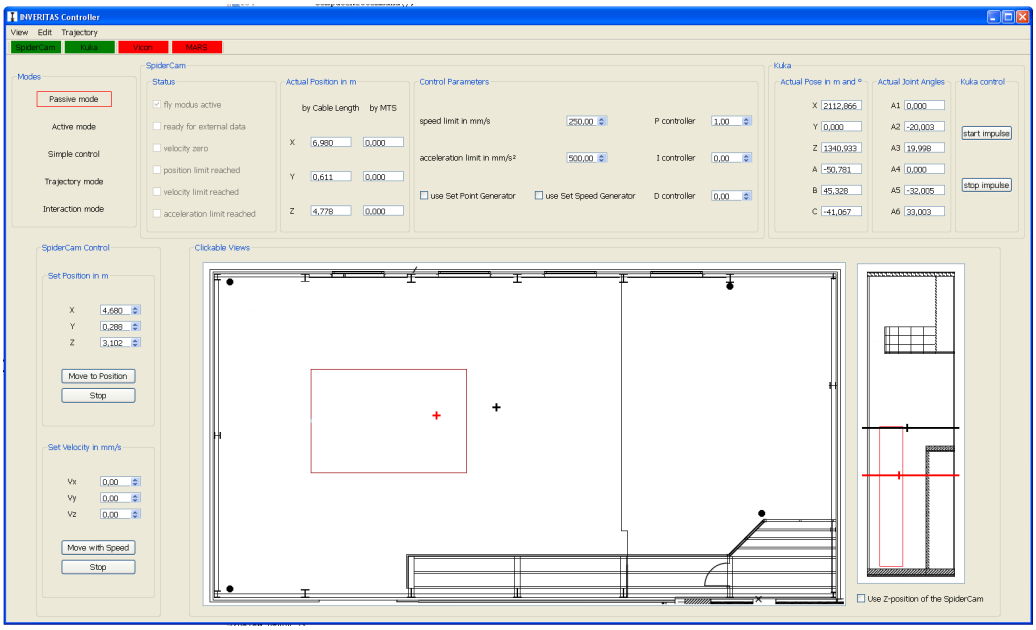


Abbildung 51: *GUI des Überwachungs- und Steuerungsprogramms*

ein geeignetes Simulationswerkzeug der Simulationsanlage zu besitzen. Vor dem eigentlichen Durchführen der physischen Simulationen kann mithilfe eines solchen Simulationswerkzeuges numerisch das Verhalten der Anlage vorab untersucht und beurteilt werden und das Werkzeug somit als wichtiger Teil der Verifikationskette dienen.

Die Simulationsumgebung wurde aus mehreren Teilen, welche zum Teil bereits existierten, zusammengestellt und diese untereinander verknüpft. Als grundlegende Softwaremodule dienen die Programme Matlab/Simulink von Mathworks und Mars vom DFKI. Folgende Funktionalitäten wurden bis jetzt umgesetzt:

- Aufbau der Simulationsumgebung in Mars
  Das vorhandene Mars-Modell der Explorationshalle (inklusive der Bewegungsanlage) wurde aktualisiert und grundlegend überarbeitet. Hierbei sei vorallem die Erweiterung der Hallenumgebung um wichtige Anlagenteile sowie eine Neuausrichtung der Kabel des CableRobot erwähnt. Desweiteren wurde eine farbliche Überarbeitung der Objekte vorgenommen, deren Schwerpunkt jetzt auf gute optische Verhältnisse statt auf Realismus liegt.

- Erweiterung des INVERITAS-Basis-Plugins für Mars
  Das Plugin wurde daraufhin erweitert, dass nun ein Laden mehrerer Szenen sowie eine farbliche Neuausrichtung der Kabel während der Simulation ermöglicht wird. Zusätzlich integrierte Sicherheitsmechanismen verhindern darüber hinaus eine falsche Bedienung der Oberfläche.

- Erweiterung des INVERITAS-Mars2Matlab-Plugins für Mars und Matlab
  Dem Plugin wurde eine Kontrolloberfläche hinzugefügt welche ein Steuern und Überwachen des Simulationsvorgangs zulässt. Darüber hinaus wurden wichtige

Komponenten wie die Inverse-Kinematik des KR60 und der PID-Regler des CableRobots eingebunden, welche erst eine realistische Simulation ermöglichen.

- Erweiterung des Simulinkmodells sendData2Mars
  Um überhaupt realistische Bewegungsdaten in Mars simulieren zu können, mussten im Modell die bereits entwickelten 12D zu 9D-Skripte integriert werden. Desweiteren war eine Umstrukturierung der TCP/IP-Verbindung erforderlich, um die Datenübertragung von Simulink zu Mars echtzeitfähig zu machen. Weiterhin war die Integration der Astrium-Trajektorien erforderlich.

Mit der Implementation der hier aufgelisteten Komponenten steht ein Simulationswerkzeug zur Verfügung, mit dem das Bewegungsverhalten der Langstreckensimulationsanlage bei der Wiedergabe beliebiger Trajektorien untersucht und beurteilt werden kann. Die Simulationsumgebung benötigt neben den Programmen Matlab/Simulink und Mars die Szenen-Datei “INVERITAS_I9.scene”, die Plugins “INVERITASBasis.dll”, “Mars2MatlabPlugin.dll”, das Simulinkmodell “sendData2Mars.mdl” sowie das M-File “MyTcpHandelerFn.m” und den Ordnerinhalt “NourTCP”. Darüber hinaus besteht keinerlei Abhängigkeit zur Anlage.

Um eine beliebige Trajektorie simulieren zu können ist es matlab-seitig lediglich nötig, das Modell “sendData2Mars” zu öffnen. Die zu simulierenden Größen sind entweder im Modell direkt zu erzeugen oder durch entsprechende Schnittstellen von außen ins Modell einzulesen. Die Schnittstelle im Modell stellt der Block “d” dar, an dem sich bereits zwei vorgegebene Blöcke namens “Eingangsgrößen” anschließen lassen. Mars-seitig ist es notwendig den Simulator mit eingebundener “InvertitasBasis.dll” zu starten. Als nächstes folgt ein Laden der Szene, üblicherweise “INVERITAS_I9”. Darauf folgend wird das “Start/Stop”-Kommando betätigt und die Funktion “12D/9D-Calculation” aktiviert, welche die TCP/IP-Verbindung herstellt und ein Kontrollfenster öffnet. Wird nun die Simulation im Simulinkmodell gestartet, startet ebenfalls die Marssimulation. Durch das Kontrollfenster der GUI in Mars sowie über das Simulinkmodell selber kann die Simulation in Echtzeit beeinflusst werden.

Erste Tests mit einer bereits eingebundenen orbitalen Rendezvous-Trajektorie von Astrium verliefen 2010 bezüglich des Arbeitsraumes der Simulationsanlage kritisch. Die Anlage geriet bei der Simulation der beiden Objekte an die Grenzen ihres Arbeitsbereiches. Durch die Ausnutzung aller Optimierungsverfahren der 12D/9D-Skripte sowie das Testen verschiedener Initial-Konfigurationen der Manipulatoren sollte anschließend untersucht werden, ob es prinzipiell möglich ist, eine so geartete Trajektorie in einem “Stück” zu Simulieren.

Erweiterungen an der Simulations- und Visualisierungs-Software MARS

Da die MARS Software zum einen als reine Visualisierung und zum anderen als komplexe Simulationssoftware verwendet werden soll, wurde eine Entkoppelung der 3D Visualisierung vorgenommen. Die 3D Visualisierung (ehemals die GraphicsFactory nun MarsGraphics) wird nun als eigene Bibliothek gebaut, gegen die von jedem beliebigem Programm gelinkt werden kann. Sie ist vollständig losgelöst von der restlichen

MARS Architektur und bietet einfache und dennoch umfangreiche Schnittstellen zur 3D Visualisierung. Grundsätzliche Neuerung ist die Verwaltung der 3D Objekte. Es existieren verschiedene Typen von 3D Objekten die jeweils durch eine eigene Klasse implementiert sind. Jede 3D Objekt Implementierung erbt von einer DrawObject Klasse die zum einen als Interface dient und zum anderen generelle Implementierungen wie die Positionierung, Skalierung und Materialzuweisungen beinhaltet. Alle generellen Implementierungen können von eine 3D Objektklasse überschrieben und an das jeweilige Objekt angepasst werden. Die 3D Objekte werden in einer Liste verwaltet und bekommen beim Erstellen eine ID zugewiesen. Über die IDs kann von außen auf die Objekte zugegriffen werden. Im folgenden sind die bisher implementierten 3D Objekte aufgelistet:

- CubeDrawObject
- SphereDrawObject
- CylinderDrawObject
- CapsuleDrawObject
- PlaneDrawObject
- TerrainDrawObject (Heightfield)
- LoadDrawObject (Import von 3D Modellen)

Des weiteren beinhaltet die MarsGraphics Bibliothek die Verwaltung eines konfigurierbaren Head Up Displays, mehrerer Fenster und Umgebungsvariablen. Diese Implementierungen wurden weitestgehend aus der vorherigen MARS Implementierung übernommen. Eine weitere Neuerung ist die Implementierung der GraphicsCameraInterface Klasse. Diese erlaubt den vollen Zugriff auf die Kamera Implementierung der 3D Fenster.
Im Zuge der MarsGraphics Implementierungen wurde die Anbindung der 3D Connexion Maus überarbeitet. Das ConnexionPlugin arbeitet jetzt nicht mehr mit absoluten Kamerapositionen sondern addiert relative Werte auf die aktuellen Werte einer Kamera. In Folge dessen kann nun die 3D Maus gleichzeitig mit der normalen Mayssteuerung verwendet werden. Zudem wurde es um ein Fenster erweitert, welches das Konfigurieren des Plugins ermöglicht (siehe Abbildung 52). In zwei DropDown Menus stehen die Verfügbaren 3D Fenster und eine Liste von den Objekten der Simulation zur Verfügung. Das ConnexionPlugin kann entweder im Modus “Camera Control” oder im “Object Control” Modus betrieben werden. Im “Camera Control” Modus wird die Kamera des ausgewählten 3D Fensters gesteuert und im “Object Control” Modus wird das ausgewählte Objekt relativ zu der Kamera des ausgewählten 3D Fensters positioniert.

Gleichzeitige Visualisierung mit MARS der Satelliten im Weltraum und in der Explorationshalle

Die relativen Bewegungen der Satelliten im Weltraum und in der Explorationshalle können mit dem DFKI-Simulationssytem MARS verglichen werden. Ein Simulink-Modell

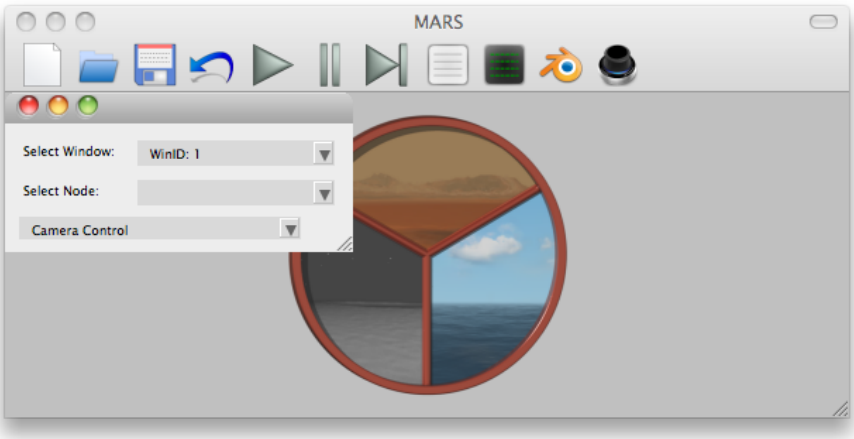


Abbildung 52: *Bildschirmfoto des ConnexionPlugin Fensters.*

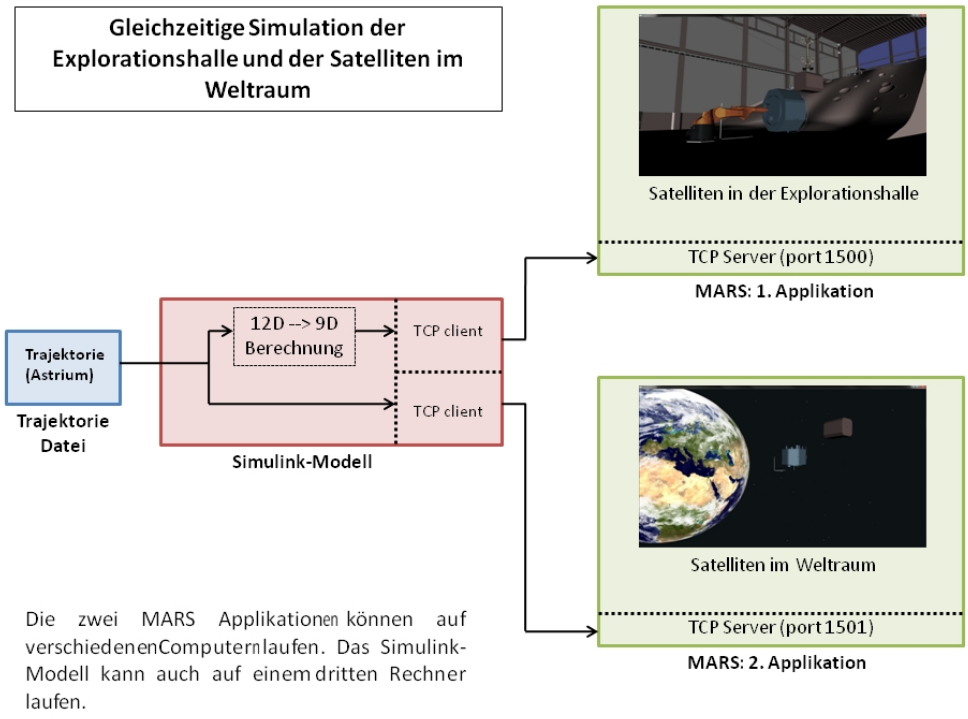


Abbildung 53: *Architektur der gleichzeitigen Simulation der Satelliten im Weltraum und in der Explorationshalle*

liest eine Trajektorie, die Astrium geliefert hat. Das Modell sendet die Position und Orientierung jedes Satelliten per TCP an eine MARS-Applikation. Diese erste Applikation simuliert die zwei Satelliten im Weltraum. Das Simulink-Modell berechnet auch die 12D nach 9D Transformation. Nach der Berechnung werden die Daten per TCP zu einer zweiten MARS-Applikation, die die Satelliten in der Explorationshalle simuliert, geschickt. Da die Daten gleichzeitig an beide Applikationen gesendet werden, laufen die zwei Simulationen synchronisiert. Jede MARS-Applikation zeigt, wie der Chaser den Client sieht. Wenn die zwei Simulationen genau gleichlaufend sind und die 12D nach 9D Berechnung eine richtige Lösung findet, sind die zwei Bilder, welche die Servicer-Kamera simulieren, identisch. Nur die Umbegung der Satelliten darf dabei unterschiedlich erscheinen.

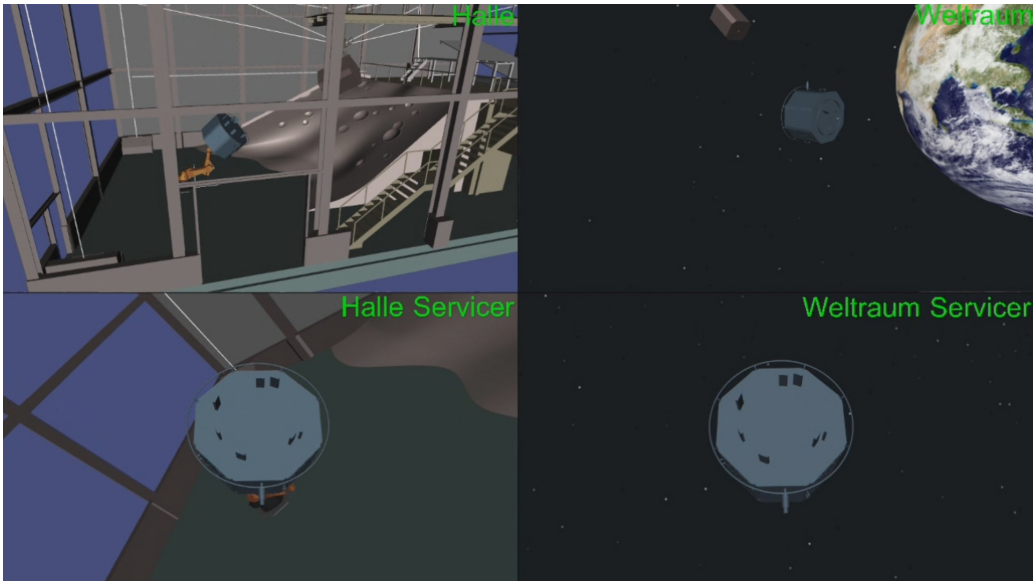


Abbildung 54: *Gleichzeitige Simulation der Satelliten im Weltraum und in der Explorationshalle*

Die Abbildung 54 zeigt die zwei synchronen Simulationen. Auf der linken Seite wird die Simulation der Explorationshalle gezeigt. Im unteren Teil wird die Sicht vom Chaser auf den Client simuliert. An der rechten Seite wird die Simulation der Satelliten im Weltraum dargestellt.

Weiterentwicklung des Simulink-Kernsystems (12D - 9D-Skript) für eine Nutzung auf der dSPACE-Hardware

Neben der Entwicklung einer Simulationsumgebung zum Zwecke der Simulations-Verifikation und -Konfiguration soll das Simulink-Kernsystem mit Namen “INVERITAS-BasicModel.mdl”, in dem das 12D zu 9D-Skript integriert wurde, auch als Steuereinheit für die reale Anlage dienen. Hierzu soll das Modell auf dem dSPACE-System kompiliert und ausgeführt werden. Damit das Modell den Anforderungen zur Steuerung der Anlage gerecht wird, waren diverse Applikationen zu integrieren. Hierzu zählt die Integration der Inversen-Kinematik des KR60, welche synchron Daten an die Simulati-

onsumgebung Mars und an die reale Anlage senden soll, das Einbinden verschiedener Sicherheitssysteme zur Überwachung der Anlage und der Bewegungskorrektur im Fehlerfall sowie das Implementieren einer manuellen Steuerung. Desweiteren war eine Anpassung der 12D-9D-Berechnungen nötig, um die Verschiebung der TCPs (Tool Center Points) durch die räumlichen Ausdehnungen der Satelliten-Nachbildungen an den Manipulatoren zu kompensieren. Es folgt eine kurze Aufschlüsselung der Arbeitspunkte:

- Die Inverse-Kinematik des KR60 konnte aus den C++ Skripten erfolgreich in das Simulink-Modell übertragen werden. Alle für den Kuka bestimmten generierten Daten stammen nun ausnahmslos aus derselben Inverse-Kinematik, was eine Simulations-Verifikation mit Mars erst ermöglicht.
- Die Kompensation der Endeffektorlagen (TCPs) der Manipulatoren wurde erfolgreich umgesetzt. Die Satelliten-Nachbildungen an den Manipulatoren bilden nun die gleiche relative Lage- und Position ab wie jene, die sich im Orbit befinden. Eine synchrone Simulation in Mars mit einem Explorationshallen-Szenario und einem Weltraum-Szenario diente zum damaligen Zeitpunkt als optischer Nachweis.
- Um eine sichere Steuerung der Anlage zu gewährleisten, war es ebenfalls Notwendig diverse Sicherungssysteme schon auf der Softwareseite zu integrieren. Folgende Systeme wurden umgesetzt:
  - Sicherheitssystem zur Überwachung der Inversen-Kinematik
  - Raumbegrenzung mit aktiver einstellbarer Dämpfung für den CableRobot (beinhaltet die grobe Hallenstruktur: Wände, Boden, Decken, Krater)
  - Kollisionsüberwachung zwischen den Kabeln des CableRobots und des Clienten
- Die Einbindung einer manuellen Steuerung der Anlage wurde fertiggestellt und kann auch während der Laufzeit angewählt werden.

Neben den hier aufgeführten Entwicklungen wurde parallel an einer INVERITAS-Simulink-Bibliothek gearbeitet.

Nach der Erweiterung relevanter Systemsegmente, wie der Inversen- und Vorwärtskinematik für den KR60, den Sicherheitssystemen für Kollisionsprävention (Entwicklung der Kollisionserkennung zwischen den Kabeln und dem Clienten wurde erfolgreich abgeschlossen) und der manuellen Steuerung, sowie diversen Modellanpassungen, wurde das 12D - 9D-Skript um folgende Funktionalitäten erweitert:

1. Eine automatische adaptive Trajektoriengenerierung für den Manöverstart
2. Ein System zur Initialisierung der Startparameter des zu fahrenden Manövers (Geschwindigkeit, Beschleunigung,...)
3. Einen manuell bedienbaren gedämpften Sicherheitsstop

4. Eine Extra- und Interpolationsmethode zur Angleichung der verschiedenen Taktzyklen von GNC und Hardware

Die Automatische adaptive Trajektoriengenerierung für den Manöverstart

Mit der Implementierung des ersten Punktes wird die Anlage vom Startpunkt des zu simulierenden Manövers entkoppelt, das heißt, dass die Anlage von jedem beliebigen Zustand aus gestartet werden kann und anschließend automatisch die Startparameter anfährt, welche durch das Manöver und den Konfigurationen im 12D/9D-Skript determiniert sind.

Das System zur Initialisierung der Startparameter des zu fahrenden Manövers

Im Betrieb kann es vorkommen, dass zum vorher bestimmten und durch die technischen Gegebenheiten notwendigen Startpunkt bereits eine Initialbeschleunigung und -geschwindigkeit vorhanden ist. Würde man die Simulation direkt über das 12D/9D-Skript starten, entstünden somit oftmals unzulässig hohe Sprünge in der ersten und zweiten Ableitung der Bewegung, welche im schlimmsten Fall die Anlage beschädigen könnten, mit Sicherheit aber die Verifikationsqualität negativ beeinflussen. Um dies zu vermeiden musste mit Punkt zwei ein System implementiert werden, welches diese Initialparameter des Manövers unter Berücksichtigung der Anlagendynamik initialisiert. Hierfür wurde ein System geschaffen, welches mittels so genannten Anfahrtsbegrenzern die Hardware auf eine bestimmte Position zwingt und dort verharren lässt. Nach dem Start des Manövers werden sich die Manipulatoren zum einen langsam aus dieser Anfahrtsbegrenzung hinaus bewegen und zum anderen werden sich gleichzeitig die Anfahrtsbegrenzer ihrerseits in entgegengesetzter Richtung von ihnen wegbewegen. Das Ergebnis ist eine Initialisierung aller Startparameter bei Vermeidung jeglicher unsteter Trajektorienverläufe. Der Vorteil dieser Methode gegenüber den klassischen offline Trajektoriengenerierungen, welche ebenfalls alle Startparameter initialisieren und in genau den richtigen Moment dann auf das Manöver umschalten liegt darin, dass zum einen der kritische Moment des Umschaltens dabei vermieden wird. Darüber hinaus ist diese Methode adaptiv, was bedeutet, dass der selbe Mechanismus bei verschiedenen Startparametern genutzt werden kann, ohne dass eine Umkonfiguration des Systems von Nöten wäre.

Der manuell bedienbare gedämpfte Sicherheitsstop

Die einmal in Betrieb genommene Anlage soll nicht nur automatisch alle Initialparameter anfahren können und das entsprechende Manöver simulieren, sondern auch eine aktive Steuerung während der Laufzeit zulassen. Hierfür ist es notwendig, ein sicheres Stoppen der Anlage zu gewährleisten. Natürlich kann die Anlage prinzipiell immer gestoppt werden, was aber unter Umständen wieder zu den unter Punkt 2 erwähnten unschönen Sprüngen im Trajektorienverlauf führen könnte. Um dies zu vermeiden

	Extrapolation	Interpolation
abs. Positionsfehler	< 20μm	ca. 40μm
zeitliche Verzögerung	ca. 40ms	100ms
Besonderheiten	mögliche dyn. Abhängigkeit	/

Tabelle 10: *Daten der Extra- und Interpolationsmethoden zum Ausgleich der versch. Taktzeiten*

wurde das Programm mit einem Sicherheits-Stop ausgestattet, mit dem die Anlage jeder Zeit gedämpft gestoppt werden kann. Die Stärke der Dämpfung ist hierbei auch während der Laufzeit abänderbar. Ein geringer Dämpfungsweg lässt die Anlage sehr schnell stoppen, erhöht aber die Momente beim Stoppen und umgekehrt.

Die Extra- und Interpolationsmethoden zur Angleichung der verschiedenen Taktzyklen von GNC und Hardware

In ersten Tests wurde die Anlage noch durch das Auslesen eines bereits vorgerechneten Orbitalmanövers gesteuert. Für die späteren Open- und Closed-Loop Simulationen wurden die Daten für die eigentliche Simulation des Manövers allerdings nicht mehr aus einer Datei gelesen, sondern in Echtzeit vom GNC generiert, welches seinerseits die Daten an das 12D/9D-Skript übergibt. Das Skript muss dann ebenfalls in Echtzeit diese Rohdaten aus dem Weltraum in einem technisch realisierbaren Anlagenzustand umrechnen, unter Beibehaltung der relativen Position und Lage der Satelliten. Hierbei treten Probleme bei den verschiedenen Taktzeiten auf: Während die Anlage alle 4ms neue Positionsdaten benötigt, berechnet das GNC bei Rendezvousmanövern nur alle 100ms einen neuen Istzustand. Um diese Diskretisierung zu vermeiden, werden folgend zwei wählbare Systeme zwischen geschaltet: Eine Extrapolations- und eine Interpolationsmethode. Die Extrapolationsmethode liefert einen sehr gut geglätteten Verlauf und verursacht bei Rendezvousmanövern (einfache Geschwindigkeit) einen absoluten Positionsfehler von gerade einmal 20μm (Maximalwert). Allerdings besteht bei dieser Methode eine gewisse dynamische Abhängigkeit, was dazu führen könnte, das sie bei unterschiedlichen Größenordnungen in Geschwindigkeit und Beschleunigung neu parametriert werden muss. Die Interpolationsmethode hingegen ist immer absolut stabil, verursacht aber einen größeren Positionsfehler durch die zeitliche Verzögerung. Die zeitliche Verzögerung bei der Extrapolationsmethode beträgt ca. 40ms, bei der Interpolation 100ms.

Implementierung des 12D/9D-Skripts in dSPACE

In Folge der Weiterentwicklung des 12D/9D-Skripts konnte das System im vierten Quartal 2010 erstmalig auf dem dSPACE-System implementiert und getestet werden. Wurde die Anlage bis dahin nur über eine offline generierte statische Trajektorie gefahren, um die Funktionalitäten der Low-Level-Programmierung zu verifizieren, geschieht die Ansteuerung der Anlage seitdem nun online über das 12D/9D-Skript. Hierfür wurde eigens eine eigene Bedienoberfläche geschaffen, welche es ermöglicht, eine komplette Überwachung der Trajektoriengenerierung durchzuführen sowie alle 12D/9D-

Konfigurationsparameter während der Laufzeit zu beeinflussen, um z. B. einen günstigeren Missionsverlauf zu gewährleisten. Über Simulationen u. a. mit Mars und durch Testläufe mit der realen Hardware wurde der neue Anlagenzustand verifiziert, womit eine Online-Nutzung des 12D/9D-Skriptes inklusive der inversen Kinematiken auf dem dSPACE-System sichergestellt ist.

Architektur der Steuerungs- und Überwachungssoftware mit dSpace

Die Simulink-Modelle von Astrium (Avionics und GNC) und vom DFKI (12d/9d Transformation und Steuerung der Explorationshalle) wurden in einem Hauptmodell zusammengefasst. Durch den Shared Memory werden die Daten zwischen den dSpace-Kernen ausgetauscht. Die simulierte Trajektorie wird am Eingang der 12d/9d Transformation in Echtzeit bereit gestellt. Die Ziel-Positionen für die SpiderCam und den KUKA-Roboter werden berechnet und über Ethernet bzw. CAN Bus zur entsprechenden Hardware gesendet. Die CAN-Verbindung mit der SpiderCam wird mit einer dSpace CAN-Karte erstellt. Auf dem vierten dSpace Kern läuft Linux, mit dem die Ethernet-Funktionalität realisiert werden kann. Der Linux-Kern bekommt alle 12 ms die Soll-Position für den KUKA geschickt und empfängt die aktuelle Position. Alle 40ms werden Daten zu den Simulationen (Mars und Astrium) geschickt. Der DFKI-Kern und der Linux-Kern benutzen einen Teil des Shared Memory um Daten auszutauschen.

Netztopologie

Nach Integration des dSpace Systems konnten erstmalig alle INVERITAS Komponenten gleichzeitig ausgeführt werden. Da bei der offiziellen Eröffnung der Space Explorationshalle das DFKI Netzwerk durch viele Benutzer stark ausgelastet wurde und nicht die nötige Bandbreite zur Verfügung gestellt werden konnte, wurde ein separates Netzwerk für die INVERITAS Komponenten erstellt. Damit ist genügend Bandbreite für den Betrieb der Anlage in unabhängig von anderen Nutzern im DFKI gesichert.

Ein zentraler Switch im Leitstand verwaltet das DFKI Netz und das interne INVERITAS Netz. Die Netztopologie kann Abb. 56 entnommen werden. 10 GBit/s können zwischen Leitstand und SpiderCam Mount ausgetauscht werden. Dies ermöglicht das Streamen mehrerer Kameras durch mehrere Computer im Leitstand. Mehrere Netzwerkdosen in der Explorationshalle sind ebenfalls direkt mit dem INVERITAS Netz verbunden, um auch außerhalb des Leitstandes auf Daten zugreifen zu können, um beispielsweise Live Bilder zu Demonstrationszwecken in der Halle anzeigen zu können.

Erweiterte Funktionalität des Kernsystems

Taktangleichung zwischen GNC und Hardware

Es existierten bisher zwei auswählbare Taktangleichungsalgorithmen zwischen der GNC und der Hardware: Eine Interpolationsmethode sowie eine Extrapolationsmethode, welche folgend als Streckenpolation (mit extrapolierendem Verhalten) bezeichnet

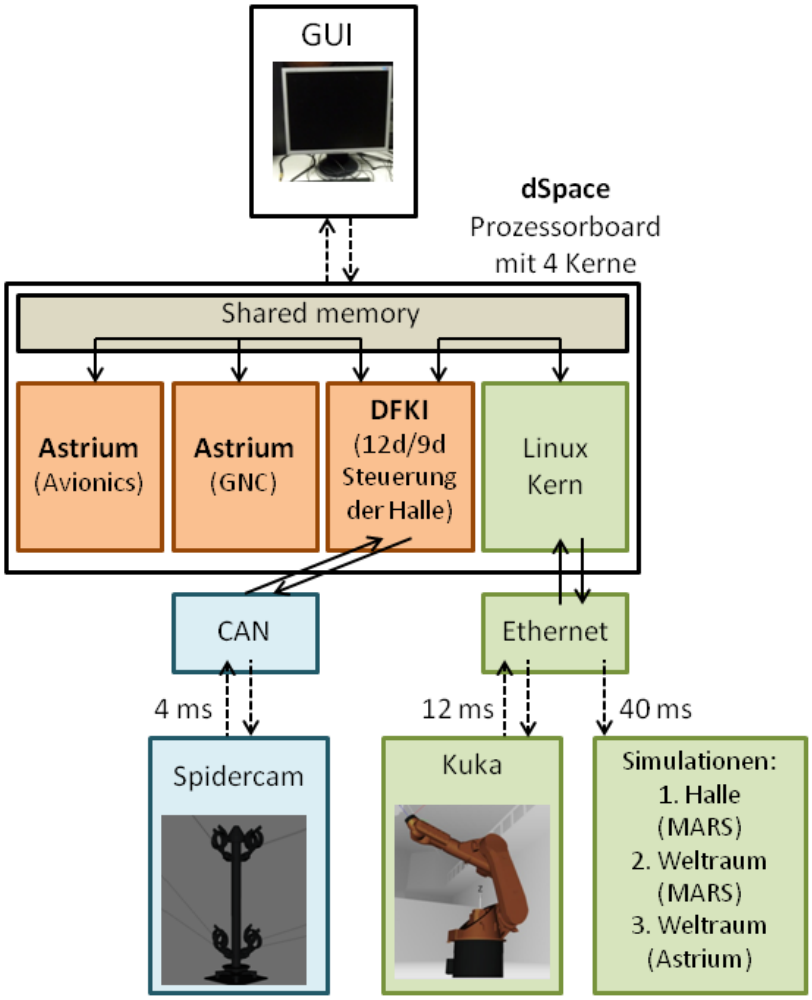


Abbildung 55: *Architektur der Software mit dSpace*

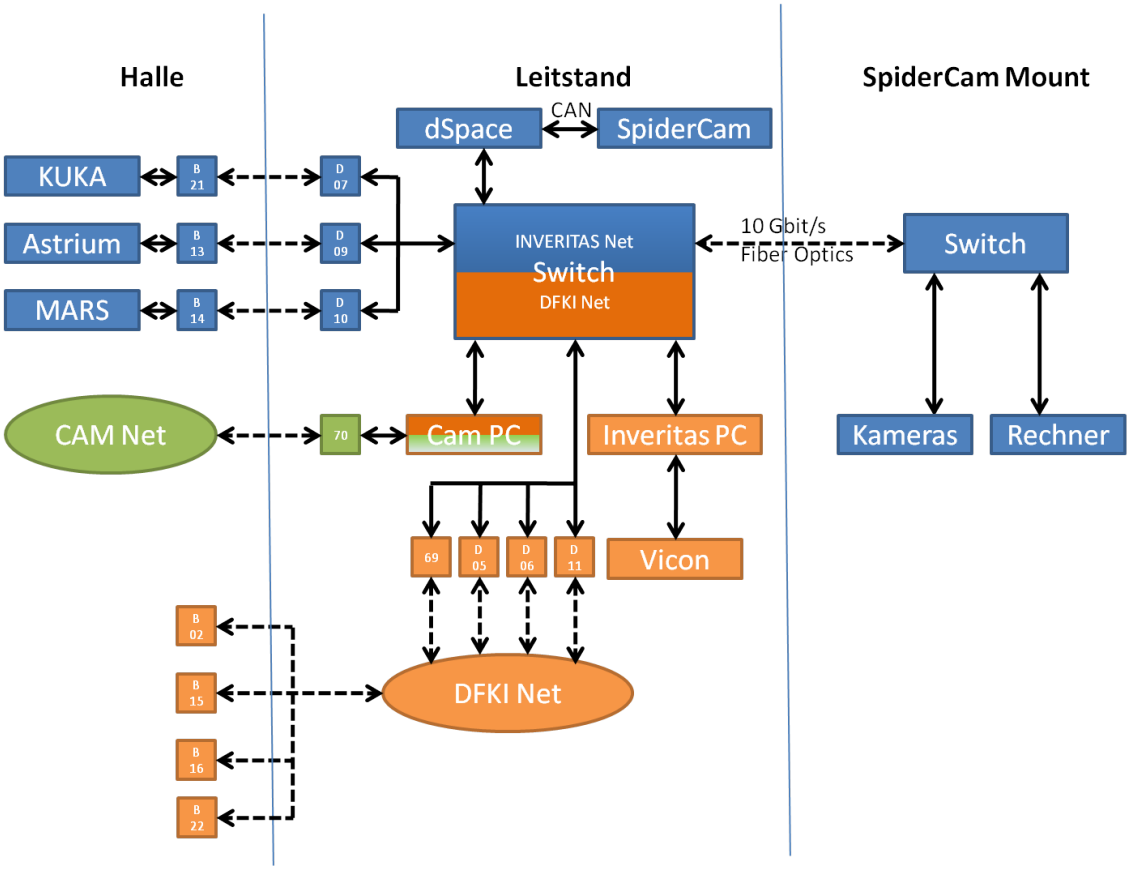


Abbildung 56: *INVERITAS Netztopologie*

wird. Um die Qualität der zukünftigen Simulationen und Verifikationen zu erhöhen sowie die Dynamik der Anlage zu stabilisieren, wurden im ersten Quartal 2011 zwei weitere (reine) Extrapolationsmethoden entwickelt. Es handelt sich hierbei um eine Extrapolation 2. Grads sowie eine Extrapolation 3. Grads. Alle Methoden stehen für jede Simulation mit der Anlage zur Verfügung. Die in Tabelle 11 angegebenen Algorithmen wurden implementiert und stehen zur Verfügung.

	abs. Pos.-Fehler (RvD)	zeitl. Verzög.	Stabilität	Rauschen
Interpolation	ca. 40μm	100ms	immer stabil	sehr klein
Streckenpolation	< 20μm	ca. 40ms	mögliche dyn. Abhängigkeit	nicht vorhanden
Extrapolation 2.$^{\circ}$	k. A. (sehr klein)	0ms	stabil bei $T_{GNC} \geq 2T_{Dspace}$	klein
Extrapolation 3.$^{\circ}$	k. A. (theoretisch Null)	0ms	stabil bei $T_{GNC} \geq 2T_{Dspace}$	mittel

Tabelle 11: *Daten der Polationsmethoden zum Ausgleich der versch. Taktzeiten*

Einbindung des CableRobots in die geschlossene Regelschleife

Die Steuerung des CableRobots erfolgte bisher in einer offenen Regelschleife (Steuerung) und wurde mittels des Kernsystems nicht geregelt. Eine Regelung erfolgte lediglich auf Low-Level-Ebene des CableRobots. Um die Genauigkeit der Simulation zu erhöhen, wurde der CableRobot im ersten Quartal 2011 in die Regelschleife integriert und wird fortan mittels eines PD2-Reglers auf der Sollposition gehalten. Da der CableRobot über Deltapositionen gesteuert wird und sich somit leicht systematische Fehler einschleichen können, ist ein großer Genauigkeitsgewinn zu erwarten. Erste Tests ergaben eine Fehlereleminierung von mehr als 10 mm. Die Regelung mittels des PD2-Reglers erfolgt sehr schnell und ohne Überschwingen. Die Abweichungen von der Sollposition betragen je nach Geschwindigkeit und Beschleunigung der Anlage 0 - 10μm. Der Regler wurde auf oberster Ebene im Subsystem SC-Control integriert. Eine dynamische Abhängigkeit des Reglers kann nicht ausgeschlossen werden.

Erweiterte Adaptivität des 12D/9D-Skripts

Um die Simulation leichter handhabbar zu machen sowie die Qualtität der Simulation zu verbessern, wurden im ersten Quartal 2011 folgende Erweiterungen vorgenommen:

- Initialposition der Anlage online auslesen
  Mit dieser Erweiterung werden die Initialposition des CableRobots sowie die Initialwinkelstellungen des Kukas ausgelesen, ins Programm übertragen und eine Trajektorie zum Startpunkt der Raumfahrt-Simulation generiert. Die Anlage kann nun von jedem beliebigen Zustand (innerhalb der gültigen Parameter) aus gestartet werden, so dass kein Anfahren einer Initialposition mehr von Nöten ist.
- Initialparameter der 12D/9D-Transformation online einstellbar
  Das 12D/9D-Skript benötigt eine Reihe von Parametern für die Transformation der Weltraumbewegungen auf die Anlage. Die Initialwerte hiervon waren bisher nach der Kompilierung fix. Eine Erweiterung dieser Funktionen erlaubt nun eine

Einstellung der Initialparameter für die 12D/9D-Transformation zwischen Anlagen- und Simulations-Start, so dass eine erneute Kompilation nicht mehr nötig wird. Die Beeinflussung der Parameter während der Simulations-Laufzeit bleibt dabei erhalten.

Erste Synthese der Simulink Modelle 12D/9D, GNC und Orbitaldynamik

Zum Erreichen des Projektziels von INVERITAS, ein Rendezvousmanöver mittels Onlinesimulation und Hardware-in-the-Loop, müssen die verschiedenen Komponenten, welche teils von Astrium und teils vom DFKI entwickelt wurden, alle in einem Simulink-Modell vereint werden, welches dann verteilt auf drei dSPACE-Kerne die Simulation ermöglicht. Hierbei handelt es sich um die Orbitalsimulation (Astrium), das GNC (Astrium) und die TestbedControl (DFKI). Die Synthese der drei Modelle wurde mithilfe eines übergeordneten Simulink-Modells realisiert, in welchem die vom Real-Time-Workshop erzeugten ausführbaren Dateien hereingeladen werden. Die Kommunikationswege können in dem übergeordneten Simulink-Modell determiniert werden.

Integration der Modelle

Erstmalig wurde im ersten Quartal 2011 eine Integration der drei Modelle zu Testzwecken durchgeführt. Astrium lieferte hierbei kompilierte Testversionen der Orbitaldynamik und des GNC. Das TestbedControl-Modell wird direkt in dem Framework integriert und vor Ort kompiliert. Die Integration verlief erfolgreich, eine Ausführung des gesamten Systems ist seitdem möglich.

Schnittstellen zwischen DFKI und Astrium

Die Schnittstellen zwischen Astrium und DFKI wurden im zweiten Quartal 2011 zusammen festgelegt. Die Schnittstellen zwischen den dSpace Kernen wurden in einem Simulinkmodell definiert. Dieses Modell schließt drei Systeme ein: Plant und Avionics von Astrium und TestbedControl vom DFKI. Die Schnittstellen werden als normale Simulink-Schnittstellen definiert. Die physikalischen Schnittstellen zwischen den dSpace Kernen werden mit Hilfe von Gigalink realisiert.

Die Ethernet-Kommunikation zwischen dSpace und den externen Systemen wurde mittels des vierten Kerns ausgeführt. Eine S-Funktion im Modell TestbedControl liest und schreibt die Daten in einen Shared-Memory. Die C++ Applikation, die auf dem Linux Kern (4. Kern) läuft, schreibt und liest diese Daten von dem Shared Memory. Jede Ethernet-Schnittstelle wird in einer eigenen C++ Klasse implementiert. Die Hauptapplikation erstellt einen Thread je Schnittstelle. Diese Architektur kann später leicht für andere Schnittstellen erweitert werden.

Die Abbildung 57 stellt die Architektur der Schnittstellen zwischen den Systemen dar. Die grünen Schnittstellen wurden definiert. Die roten waren zum damaligen Zeitpunkt noch nicht definiert.

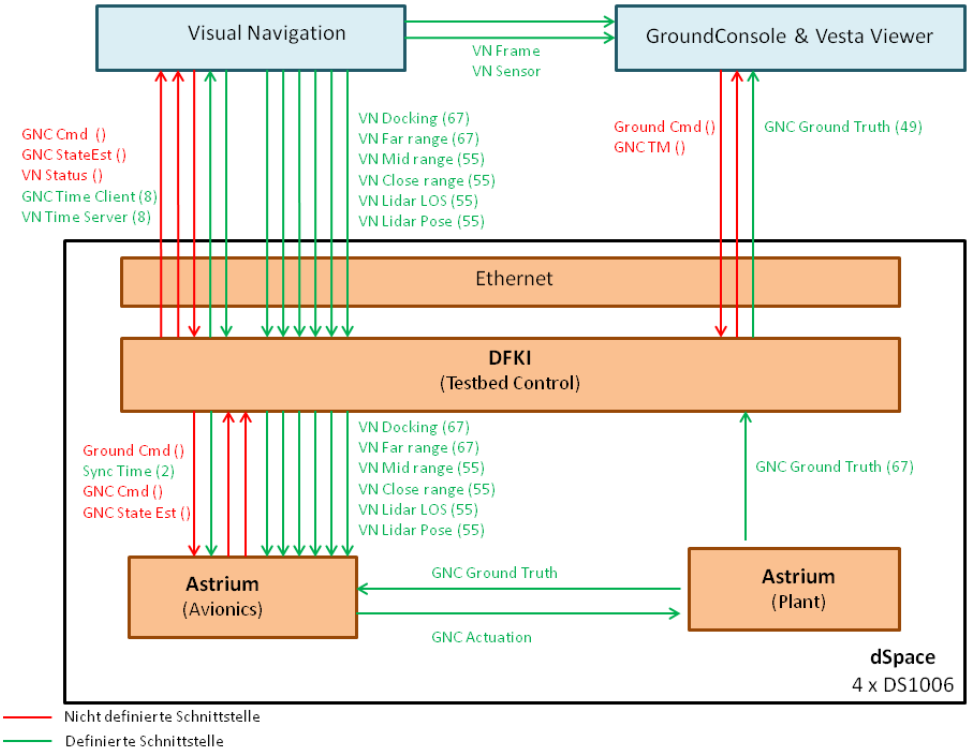


Abbildung 57: *Architektur der Schnittstellen zwischen den dSpace Kernen und zwischen dem Linux Kern und den Systemen von Astrium*

Zwei Schnittstellen wurden implementiert und auf der Hardware getestet: die Zeit-Synchronization und GNC Ground Truth.

Zeit-Synchronization

Die Applikation für die Bildverarbeitung läuft auf einem Linux PC und sendet die Positionsschätzung an das GNC-System (dSpace). Um diese Information zu verwenden wird ein Zeitstempel gebraucht. Auf einem Linux (oder Windows) PC bekommt man als Zeit die Dauer von dem 1. Januar 1970 bis jetzt. Dagegen bekommt man auf dem dSpace System die Dauer seit dem Start der Applikation. Ein Verfahren zur Zeitsynchronisation wurde erstellt, implementiert und getestet. Regelmäßig (alle 1 bis 5 Sekunden) werden Zeitstempel zwischen der Visual Navigation und dSpace ausgetauscht. Die Visual Navigation schickt ihren Zeitstempel. Wenn dSpace die Nachricht empfängt, speichert es seine eigene Zeit. Er sendet dann sofort seinen Zeitstempel zurück. Die Visual Navigation speichert einen vierten Zeitstempel und schickt alle diese Informationen an dSpace zurück. Mit diesen vier Werten werden der Offset und die Round-Trip Zeit berechnet. Diese Schleife des Daten-Austauschs muss regelmäßig ausgeführt werden da die Genauigkeit der Uhr Schwankungen unterworfen ist. Eine korrigierte Zeit wird dann berechnet und für die GNC Simulation benutzt.

GNC Ground Truth

Eine zweite Schnittstelle wurde erfolgreich implementiert und getestet: die Schnittstelle für die GNC Ground Truth Daten. Das GNC Modell von Astrium generiert die Bewegung der Satelliten im Weltraum. Diese Daten werden zuerst zum TestbedControl Kern weitergeleitet. Sie werden im Shared Memory gespeichert und an den PC für Visualization und Ground Control geschickt. Astrium hat jetzt die Möglichkeit, seine eigene Simulationssoftware Vesta Viewer zu verwenden.

Kollisionserkennungsalgorithmus für den Kuka Arm und den Cable-Robot

Während der Bewegungssimulationen sind die wahrscheinlichsten Kollisionen Eigenkollisionen zwischen Kuka und Client. Kollisionen zwischen dem Cable-Robot/Servicer und dem Kuka-Arm/Client müssen auch überwacht werden. Um diese Kollisionen zu erkennen, wird ein Algorithmus [6] verwendet, der eine feine Darstellung der Systeme benutzt und trotzdem echtzeitfähig bleibt. Dieser Algorithmus stellt die Robotersysteme als konvexe Hüllen dar, die mit einer kleinen Anzahl von Punkten definierbar sind. Eine zusätzliche Applikation berechnet und optimiert diese Volumen für jeden Körper. Die Hauptapplikation (in C++ geschrieben) nimmt als Eingangsdaten die Gelenkwinkel des Roboters, berechnet die Vorwärtskinematik und die Position der Volumen und liefert dann die geringste Distanz zwischen den Volumen. Der Algorithmus ist echtzeitfähig: Für das komplexe Modell bleibt die Berechnungszeit im Millisekundenbereich. Es gibt auch die Möglichkeit, die Berechnungskosten zu begrenzen. Wenn die Grenze erreicht ist, wird keine exakte Lösung geliefert, sondern ein annähernder Wert.

Eine Konfigurationsdatei für den Kuka-Arm und den Client wurde im dritten Quartal 2011 geschrieben und ein spezifisches Programm für das Projekt INVERITAS entwickelt. Das Programm lädt die Datei und erstellt die C++-Objekte und eine Ethernet-Verbindung mit dem dSpace-System. Das dSpace-System sendet die Gelenkwinkel des Kukas an das Programm per TCP/IP. Das Programm aktualisiert den Zustand und ruft die Kollisionsberechnungsmethode auf. Die geringste Distanz und die Id-Nr. der betroffenen Körper werden sofort zurück geschickt. Falls Pakete mit einer Verzögerung ankommen, wurde auch ein Timeout-Mechanismus entwickelt. Der dSpace-Kern sendet einen Zeitstempel gleichzeitig mit den Gelenkwinkeln. Dieser Zeitstempel wird mit den Ergebnisdaten zurück geliefert, wodurch das dSpace System die Dauer der ganzen Schleife berechnen kann. Falls die geringste Distanz oder die Roundtrip-Dauer kleiner bzw. höher als eine bestimmte Grenze ist, wird die Bewegung der Roboters sofort gestoppt.

Integration der Korrekturfunktionen für den Cable-Robot

Mehrfach durchgeführte Messungen ergaben, dass die Positionsbestimmung des CableRobots systematischen Fehlern, sowohl in der Translation, als auch in der Rotation unterliegt. Aufgrund der Systematik der Fehler konnten im vierten Quartal 2011 zwei

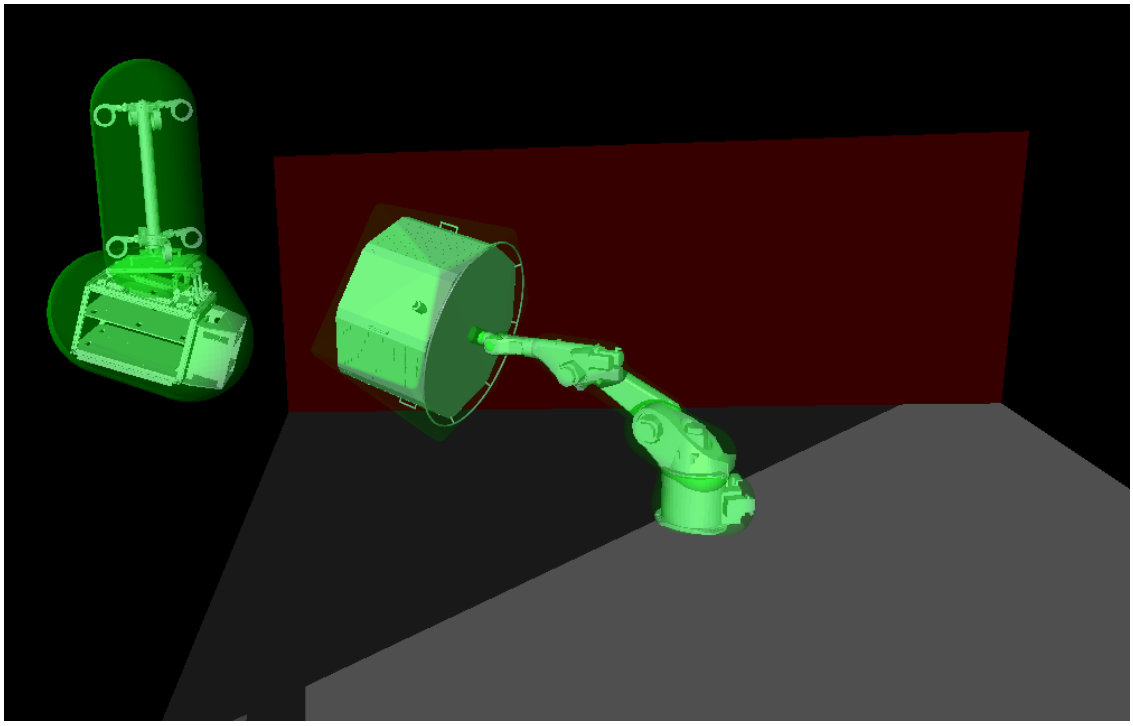


Abbildung 58: *Kuka-Arm, Client, Cable-Robot und Servicer mit entsprechenden Konvexhüllen*

Polynom-Funktionen entwickelt werden, welche fortan die Fehler herausrechnen können.

Diese Polynomfunktion wurde in den geschlossenen Regelkreis des CableRobots unter Simulink und dSpace integriert. Die systematischen translatorischen Fehler können hierbei direkt über die Position des CableRobots kompensiert werden. Da der CableRobot allerdings nur einen rotatorischen Freiheitsgrad besitzt, ist es nicht möglich, die Rotationsabweichung direkt zu kompensieren. Um dieses Problem zu umgehen, wurden im 12D-9D-Skript Eingänge vorgesehen, welche die Rotationsabweichung des CableRobots vom Korrekturpolynom einlesen. Das 12D-9D-Skript wird daraufhin eine Anlagenposition von Kuka und CableRobot berechnen, welche es ermöglicht, die Rotationsabweichung relativ zueinander zu kompensieren.

Da beide Kompensationverfahren in einer Abhängigkeit voneinander stehen, konvergiert das System jeweils in den nächstgelegenen stabilen Zustand. Dieser Vorgang funktioniert fehlerfrei und ohne Schwingungen.

Entwicklung einer Höhenstabilisierung für den CableRobot

Simulierte RvD-Manöver benötigen für eine ausreichende Verifikation einen möglichst großen Anflugweg. Bedingt durch die Topologie der Halle war es bisher nicht möglich den maximalen Anflugweg zu nutzen, da sich innerhalb der Anflugstrecke ein Teilstück der Mondlandschaft befindet. Der CableRobot kann nur in einem kleinen Höhenbereich die höchste Stelle des Kraters passieren, wodurch der CableRobot je nach Trajektorie

in seine Grenzen fährt.

Um dieses Problem zu umgehen, wird nach einer Weiterentwicklung im vierten Quartal 2011 der CableRobot auf der kritischen Strecke in einem vorher definierten Höhenbereich gehalten, so dass die Mondlandschaft passiert wird. Die geforderte Relativgenauigkeit wird eingehalten, in dem eine Kompensation der translatorischen Abweichung des CableRobot über die translatorische Stellung des Kukas durchgeführt wird. Hierbei ist zu bedenken, dass sich der Kuka nach erfolgreicher Kompensation in einer sehr ungünstigen Konfiguration befinden kann. Um die vollständige Bewegungsfreiheit des Kukas zu gewährleisten, wird eine Ausgleichsbewegung über eine PD2-Regelung initialisiert, sobald der CableRobot den kritischen Bereich verlassen hat. Der hierfür notwendige Puffer ist konfigurierbar.

Je nach Initialposition des Kukas ist es allerdings nicht immer möglich, die translatorische Abweichung des CableRobots zu kompensieren, bedingt durch den beschränkten Arbeitsbereich. Um die Kompensationsfähigkeit des Kukas zu optimieren, wird dieser in eine möglichst günstige Initialstellung und später auf eine für den Rest der Simulation günstige Endstellung gebracht, welche vom PD2-geregelten Ausgleichsvorgang erzwungen wird.

Sicherheitssysteme und neue GUI

Für die Bedienung der neuen Kontrolloberfläche, in der die drei Komponenten GNC, Orbitaldynamik und TestbedControl integriert wurden, wurde im vierten Quartal 2011 eine neue GUI entwickelt, welche sich aus insgesamt sechs Layouts zusammensetzt:

- Der Hardware-Kontrolloberfläche
- Der 12D-9D-Kontrolloberfläche
- Eine erweiterte 12D-9D-Kontrolloberfläche (für ein tieferes Eingreifen ins System)
- Ein Übersichts-Layout
- Eine Diagramm-Visualisierung der Hardware-Trajektorien
- Eine Diagramm-Visualisierung der 12D-9D-Soll-Trajektorien sowie Fehleranalyse

Ein neuer Integrationsbestandteil ist die Fehlerüberwachung der Anlage. Folgende Fehlerprüfungen wurden in die Überwachung eingebunden:

- Fehlerhafte Initialposition des Kukas (verhindert den Anlagenstart)
- Fehlerhafte Initialposition des CableRobots (verhindert den Anlagenstart)
- Aussetzen der Inversen-Kinematik (stoppt die Kuka-Bewegung)
- Kollisionsprävention zw. Client und den Kabeln des CableRobots (stoppt zeitlich gedämpft)

- Erreichen des Dämpfungsbereichs der Umgebungsgrenzen für den Kuka (räumliche Dämpfung nur auf der entsprechenden Achse, keine Unterbrechnung der Simulation)
- Erreichen des Dämpfungsbereichs der Umgebungsgrenzen für den CableRobot (räumliche Dämpfung nur auf der entsprechenden Achse, keine Unterbrechnung der Simulation)

Ebenso wird der aktuelle Status der Höhenstabilisierung über die folgenden drei Zustände dargestellt:

- Normalfahrt (keine Beeinflussung)
- Höhenstabilisierung (zwingt den CableRobot, in einem definierten Bereich zu bleiben)
- Ausgleichsbewegung (macht den Kompensationsvorgang rückgängig und verharrt bei vordefiniertem Endzustand)

Der Bewegungsraum auf Höhe der kritischen Strecke kann über die GUI konfiguriert werden, ebenso wie der Endzustand nach der PD2-geregelten Ausgleichsbewegung. Des Weiteren sind nun auch alle Umgebungsgrenzen (Halle und Mondlandschaft) sowie der Dämpfungsbereich über die GUI einstellbar. Für den Initialzustand der Anlage kann nun zwischen Hardwarezustand oder simuliertem Zustand gewählt werden (nötig für eine Simulation mit abgeschalteter Hardware). Der Initialzustand der Anlage wird nun auch vor dem Simulationsstart an die Inverse-Kinematik übergeben, was zu einer wesentlich günstigeren Anfangskonfiguration des Kukas führt. Die Geschwindigkeit der Simulation wird nun über eine PT2-Strecke geführt und besitzt den Bereich -5x bis +5x, wobei x die Originalgeschwindigkeit ist.

Ebenfalls wurde nun auch die Steuerung der zusätzlichen Z-Achse in das Simulink-Modell und dSpace integriert und kann, falls gewünscht, über das 12D-9D-Skript mit angesteuert werden.

Im Kernsystem wurden im ersten Quartal 2012 weitere Verbesserungen vorgenommen. Zum einen wurde durch erste Experimente zusammen mit den Komponenten von Astrium deutlich, dass eine Erhöhung der Genauigkeit des Bewegungssystems erforderlich war. Zum anderen wurden intelligente Verhalten implementiert, wodurch das Bewegungssystem den gegebenen Arbeitsraum automatisch optimal ausnutzt und gleichzeitig die Bedienung der Anlage erleichtert wird.

Rotationskorrektur

In den gemeinsamen Experimenten mit Astrium konnte die Lageschätzung des LIDARs mit der gewünschten Sollpose von Servicer und Client verglichen werden. Es stellte sich heraus, dass sich ein Fehler von einigen cm einstellt, der abhängig von der Position des CableRobots im Raum war. Es ließ sich auf eine Drehung des CableRobots zurückführen. Diese ungewollte Rotation musste korrigiert werden, um die Genauigkeit der Anlage entscheidend zu erhöhen.

	Rauschen der Position	Rauschen der Rotation
An Arbeitsraumgrenze	+/-5.00 mm	+/-0.150°
Mitten im Arbeitsraum	+/-0.05 mm	+/-0.002°

Tabelle 12: *Rauschen des CableRobot-Objekts im Motion Tracking System*

Mit der Anlagenkalibrierung wurde deshalb auch ein Korrekturpolynom für die Rotation des CableRobots erstellt. Diese und die online mit dem MTS gemessene Rotation bilden den Eingang für einen Kalman-Filter der beide Daten fusioniert und eine stetige Rotationsschätzung berechnet auch wenn keine MTS Daten verfügbar sind. Diese Rotation wird jetzt als zusätzlicher Eingang dem 12D/9D- oder 12D/6D-Kernsystem zugeführt.

Mit den neu ermittelten Orientierungswerten des Servicers kann durch eine Umkonfiguarion des Kukas und des CableRobots die gewünschte Relativposition beider Objekte zueinander wiederhergestellt werden. Da dieses Vorgehen einer mathematisch exakten Kompensation entspricht, ist die Genauigkeit der Fehlerkompensation einzig allein von der Genauigkeit des ermittelten Rotationsfehlers abhängig. Der eliminierte Fehler pro Orientierungsfehlereinheit steigt mit der Entfernung des Servicers vom Clienten. Deshalb wird über einen PID Regler die Rotationskorrektur langsam erhöht, um Schwingungen zu minimieren.

Kalman-Filter

Die Experimente im Closed-Loop, die mit Astrium durchgeführt wurden, haben gezeigt, dass die Annäherungspolynome für die Position und die Rotation des CableRobots teilweise nicht ausreichend genau sind. Außerdem wurde durch den Vergleich verschiedener Testkampagnen ersichtlich, dass die Güte der Polynome stetig abnimmt. Ursachen hierfür könnten Temperaturunterschiede oder Fehler in der Kabellängenmessung des CableRobots selbst sein.

Aus diesem Grund muss das Motion Tracking System (MTS) als ständige Referenzmessung integriert werden. Das MTS hat den Nachteil, dass es nicht den gesamten Arbeitsbereich abdeckt und an den Randbereichen das Rauschen der Daten stark zunimmt und gleichzeitig die Genauigkeit nachlässt, da weniger Marker von weniger Kameras gesehen werden (Tab. 12). Um mit den Nachteilen des MTS umzugehen und trotzdem dessen Vorteile zu nutzen, wurde ein Kalman-Filter implementiert, der die Pose aus der Polynom-Funktion mit den MTS Messdaten fusioniert.

Der Kalman Filter hat zwei Schritte, die sich nacheinander abwechseln (Abb. 59):

- In der **Prädiktion** wird anhand eines definierten Evolutionsmodells für die Position und Rotation nach jedem Zeitstempel die nächste Pose geschätzt. Die Kovarianz-Matrix wird ebenfalls aktualisiert, die das Vertrauen in die Prädiktion darstellt.
- Eine **Korrektur** der Schätzung wird jedes Mal vorgenommen sobald neue Messwerte von den Sensoren kommen. Der Kalman-Gain und die Kovarianz-Matrix der Messdaten werden zunächst aktualisiert. Je größer der Kalman-Gain, desto mehr

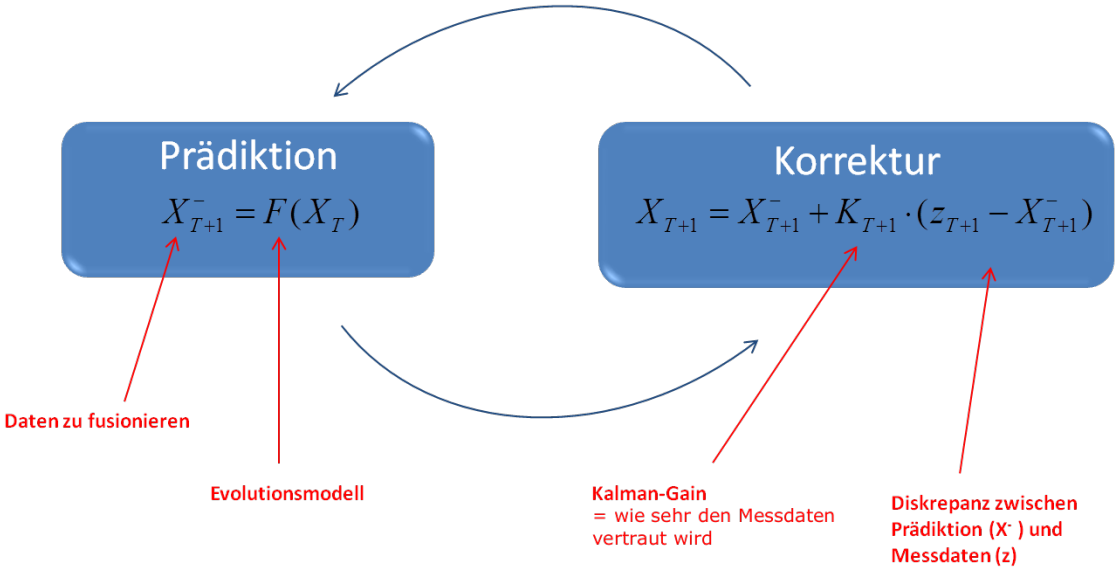


Abbildung 59: *Grobes Prinzip des Kalman-Filters*

wird den jeweiligen Messdaten vertraut. Die Diskrepanz zwischen Messdaten und Prädiktion wird mit dem Kalman Gain multipliziert und auf die Prädiktion addiert. Das Ergebnis ist der fusionierte Messwert. Gegebenenfalls wird die Kovarianz-Matrix aktualisiert.

In unserem Fall werden sowohl die MTS Daten als auch die Daten aus den Korrekturpolynomen als Messwert angesehen. Das Verhalten des Kalman-Filters wird durch die Einstellung der Kovarianz-Matrizen festgestellt. Wenn man mehr der Pose des Polynoms vertraut, orientiert sich der Output des Kalman-Filters näher am Polynom. Wenn dem MTS und dem Polynom gleich vertraut wird, sind die gefilterten Daten ungefähr in der Mitte. Wenn dem MTS sehr stark vertraut wird, rauschen die Ergebnisse fast so stark wie die MTS Messdaten selbst. Für den ersten Versuch wurde die Absolutgenauigkeit des Cable-Robots und des MTS genommen (siehe Tab. 13). Das Problem mit diesem Ansatz ist jedoch, dass in der Mitte des MTS Arbeitsbereichs den MTS Daten zu wenig und an den Randbereichen zu viel vertraut wird und somit durch das Rauschen der Messwerte große Schwingungen entstehen (Abb. 60). Vor allem bei großen Relativabständen zwischen Servicer und Client ergeben sich große Schwingungen, da durch die Rotationskorrektur Client und vor allem der Servicer große Positionsänderungen vornehmen müssen. Außerdem gibt es ein Problem in den Randbereichen des MTS. Hier kann es passieren, dass der CableRobot seine Position aufgrund der neuen MTS Daten ändert, damit aber wieder aus dem Arbeitsbereich der MTS rausfährt. Ein Aufschwingen wird somit unausweichlich.

Zur Verbesserung wurde ein Puffer eingefügt, der mit MTS Daten gefüllt wird und erst bei Überlauf die MTS Daten zum Kalman-Filter hinzuschaltet. Außerdem wird über die gesamte Länge des Puffers die Varianz berechnet und anhand dieser Größe das Vertrauen in die Daten angepasst. Somit vertraut man dem MTS an dessen Grenzbereich wo ein hohes Rauschen der Werte auftritt weniger als im Zentrum.

Der Vergleich mit den Sensordaten des Servicers (LIDAR) zeigte jedoch, dass trotz des hohen Rauschens die Genauigkeit des MTS am Randbereich höher ist, als die des Polynoms. Aus diesem Grund wurde ein Median Filter vor dem Kalman-Filter eingesetzt, der vor allem Spitzen in den MTS Daten herausfiltert. Dadurch sind die Eingangsdaten und somit auch die Ausgangsdaten weniger verrauscht und durch die damit verbundene kleinere Varianz wird auch den MTS Daten mehr vertraut. Der Nachteil ist, dass der gesamte Filter langsamer auf Änderungen reagiert. Die aktuelle Implementierung zeigt Abb. 60, doch könnte das Resultat durch feineres Einstellen der verschiedenen Parameter weiter verbessert werden.

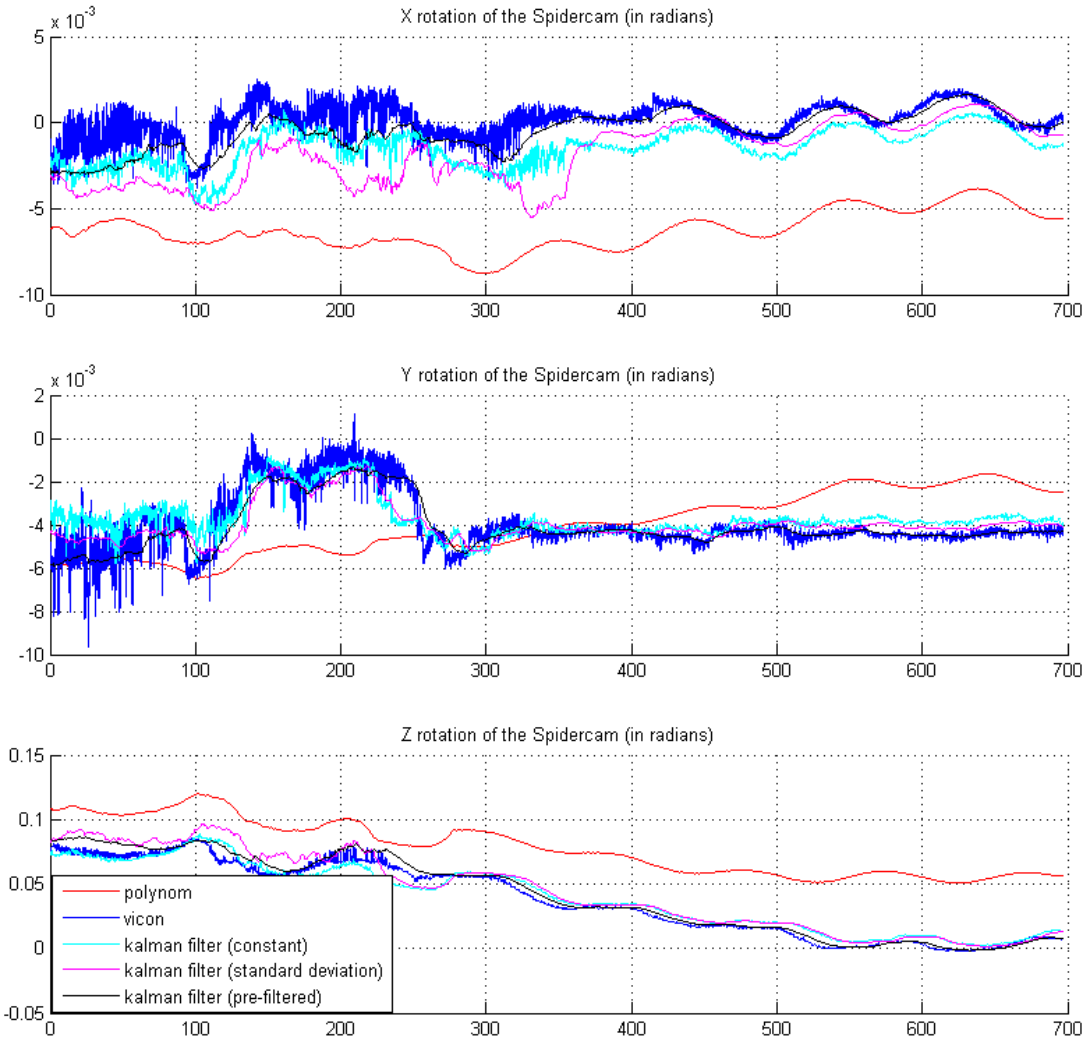


Abbildung 60: *Vergleich der Kalman-Filter Iterationen*

	Absolutgenauigkeit	Wiederholgenauigkeit
KUKA (ohne Korrektur)	1.6 mm,	0.01 mm
CableRobot ohne Korrektur	267.4 mm	1.1 mm
CableRobot mit Korrektur	6.6 mm	1.1 mm
MTS ohne Korrektur	8.9 mm	0.6 mm
MTS mit Korrektur	5.3 mm	0.6 mm

Tabelle 13: *Absolut- und Wiederholgenauigkeit des Cable-Robots*

Automatische Steuerung der z-Achse des CableRobots

Die Anbringung der Z-Achse am Cable-Robot dient der Vergrößerung des Arbeitsraumes durch das Hinzufügen eines an sich redundanten Freiheitsgrades. Die Z-Achse kann Rotationen des Systems um die globale (nicht relative) Z-Achse ausführen.

Hintergrund: Rotationen um die globale Z-Achse werden (bei geringen Anflugwinkel) immer dann nötig, wenn der Servicer relativ zum Clienten um die Weltraum-Z-Achse (zeigt Richtung Erdmittelpunkt) rotiert. Vor Anbringnung der zusätzlichen Z-Achse an dem CableRobot musste dieser systembedingt immer große Ausgleichsbewegungen radial zum Drehpunkt tätigen, wobei vor allem die Platzverhältnisse in Y-Richtung sehr begrenzt waren. Mit der zusätzlichen Z-Achse kann eine solche Rotation (beinahe) direkt ausgeführt werden, die Ausgleichsbewegung ist gering, der Arbeitsraum vergrößert.

Die Ansteuerung der Z-Achse basiert auf einem PD-Kompensationsregler, welcher einen einstellbaren Anflugwinkel des Servicers um die globale Z-Achse beibehält. Die Notwendigen Ausgleichsbewegungen werden von der Z-Achse übernommen.

Automatische Steuerung der Kuka Achsen A2 und A3

Die A2/A3-Automatik dient der Vergrößerung bzw. Optimierung des Arbeitsbereiches der Simulationsanlage und wurde für den Betrieb mit Kuka-3D konzipiert. Die Kuka-3D-Lösung arbeitet lediglich mit drei der sechs Kuka-Achsen, den Achsen A1, A3 und A6. Die Achsen A4 und A5 werden fixiert, um eine höhere Nutzlast am Kuka-TCP zu ermöglichen. Achse A2 ist hierbei redundant und wird nun für eine Erweiterung des Arbeitsbereiches eingesetzt. Besonders die geringen vertikalen Freiräume des Cable-Robots auf großer Distanz vom Kuka schränken die Simulationsfähigkeit enorm ein. Mit dem Einbinden der Achse A2 kann dieser Arbeitsbereich vergrößert werden. Dazu wird ein Offset auf den eigentlich konstanten Winkel der Achse A2 gegeben, wodurch sich die Höhe des Clients ändert. Auf die Achse A3 wird der entgegengesetzte Offset gegeben, damit am TCP kein Rotationsfehler entsteht. Die realtive Gesamtbeweglichkeit in der Vertikalen wird somit vergrößert (Abb. 61). Neben der Höhenänderung kommt es auch zu einer Positionsänderung in der xy-Ebene. Es existieren zwei Methoden zur Steuerung der Achse A2:

- Methode 1 hält die Höhe des CableRobots konstant und steuert entsprechend die Achse A2 an, um eine benötigte relative Höhenänderung durchzuführen (Abb. 62).

- Methode 2 lässt den CableRobot die benötigte Höhenänderung solange selbst durchführen, bis zum Erreichen einer Grenze. Von da ab übernimmt die Achse A2 die entsprechende Höhenänderung.

(a) Hohe Lage des Clienten (b) Tiefere Lage des Clienten

Abbildung 61: *A2/A3 Automatik ändert die Position des Clienten ohne die Orientierung zu beeinflussen*

Automatische Steuerung der Kuka Achsen A4 und A6

Die A4/A6-Automatik dient der Vergrößerung des Arbeitsbereiches der Simulationsanlage im Kuka-3D-Modus. Hierbei wird das relative Rotationsvermögen der beiden Objekte (Client und Servicer) zueinander erhöht. Die Rotationsachse entspricht der Verbindungslinie beider Objekte (im folgenden als X-Achse bezeichnet). Bei der Simulation von Objekten, welche zueinander um ihre X-Achse rotieren, ist eine physische Begrenzung des Arbeitsraums durch die Kuka-Achse A6 gegeben, welche als einzige diese Rotation abbilden kann (Kuka-3D). Durch die gesonderte fixe Stellung der Achse A5 befindet sich die Achse A4 auf der gleichen Rotationsachse wie Achse A6 (Singularität). Diese Besonderheit lässt sich ausnutzen, indem beide Achsen für eine Rotation um X verwendet werden. Das Rotationsvermögen kann somit verdoppelt werden. Somit wird ein manuelles Umkonfigurieren und damit das Starten und Stoppen des gesamten Bewegungssystems hinausgezögtert oder vermieden.

Die Stellung der Achse A4 wurde im Kuka-3D-Modus so gewählt, dass die Achse A5 niemals dem Knickmoment des Clienten ausgesetzt ist (da dieses zu hoch werden kann). Mit der zusätzlichen Nutzung der Achse A4 kann dies nicht mehr gewährleistet werden. Um die Belastung auf der Achse A5 gering zu halten wird ein so genannter

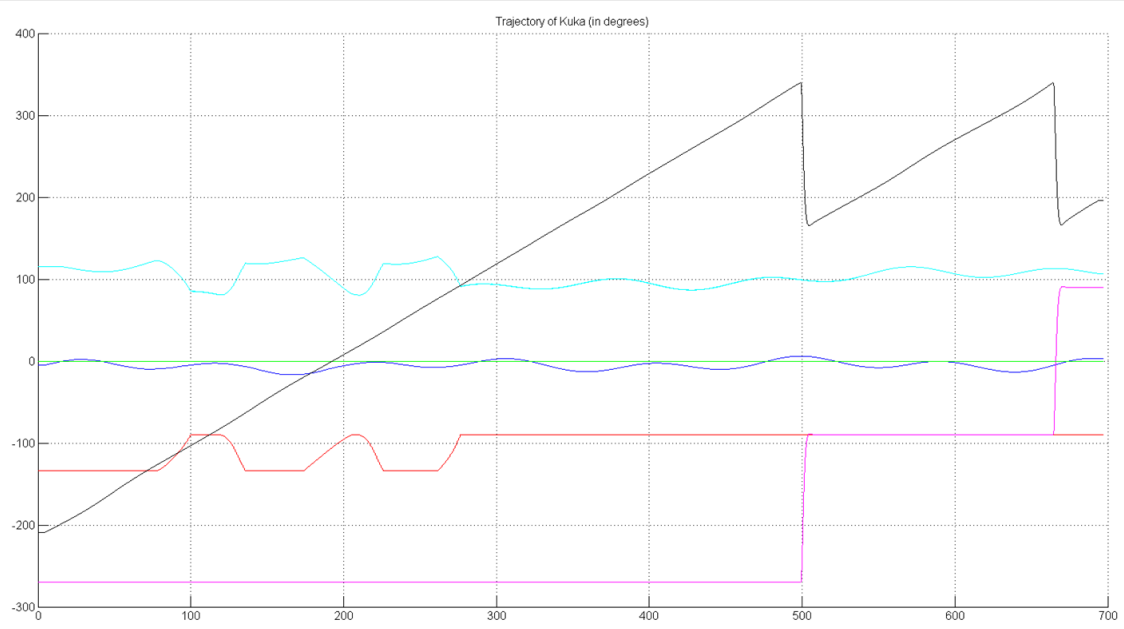


Abbildung 62: *Gelenkwinkel des Kuka während der Mission 108 - Die rote Linie zeigt das Einsetzen der A2/A3 Automatik um den Servicer im Arbeitsbereich zu halten. Die schwarze und magentafarbende Line zeigen die automatische Umkonfiguration von Achse A4 und A6, die ein Stoppen der Anlage für eine manuelle Umkonfiguration verhindert*

Umschaltmoment initiiert. Das heißt, wenn A6 an ihre Grenze gerät, drehen sich die Achsen A4 und A6 gegenläufig um jeweils 180 Grad, so dass A6 wieder Spielraum besitzt (Abb. 62). Die Umschaltung erfolgt über eine PT2-Strecke. Da die Achse A5 bei zu langer Überbelastung ausschaltet, ist die Zeit eine wichtige Komponente. Die PT2-Strecke wurde entsprechend ausgelegt.

3.1.14 AP42100: Sensorsystemsimulation

Mit dem DFKI-Simulationssystem MARS werden verschiedene Sensoren simuliert. Insbesondere ist die Simulation von Laserscannern möglich. Die Auflösung, der Öffnungswinkel oder die maximale Reichweite des Sensors können gewählt werden. Nachdem endgültige Spezifikationen und Scandaten des in INVERITAS eingesetzten LIDAR-Laserscanners vorliegen, kann die Simulation dieses LIDARs im Weltraum und in der Halle weiterentwickelt werden. Zu diesem Zweck wurde ebenfalls ein Scan der Kraterlandschaft mit dem Leica Laserscanner vorgenommen. Mit Hilfe der daraus entstandenen Punktwolke kann eine Objekt in Mars erstellt werden, mit dem der simulierte Lidar getestet werden kann.

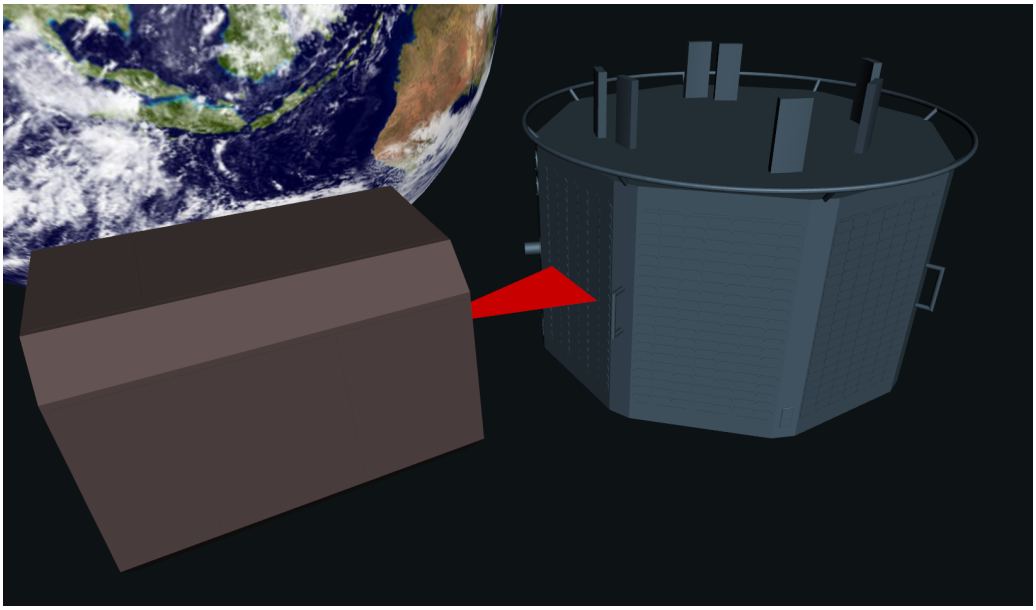


Abbildung 63: *Simulation eines Laserscanners im Weltraum*

Aufgrund der engen Verknüpfung mit den Systemen von Astrium wurde zur Sensorsimulation letztendlich eigenentwickelte Software von Astrium herangezogen, insbesondere die Simulation von Kamera- und Laser-Sensoren (wie z.B. dem Lidar) ist jedoch auch in zukünftigen Projekten alternativ mit den hier erläuterten Fähigkeiten der stetig weiterentwichekten DFKI-Simulationssoftware MARS möglich.

3.1.15 AP42200: RCS-Simulation

Das in AP41100 skizzierte Kernsystem bietet durch seinen modularen Aufbau die Möglichkeit, beliebige Steuerungs- und Simulationssysteme für die zu simulierenden Objekte zu integrieren. Dies gilt auch für die RCS-Simulationskomponente und die GNC-Steuerungskomponenten. Dabei müssen die Kommandos der GNC-Komponente einerseits in die RCS-Simulation einfließen, in der Gegenrichtung müssen jedoch auch Ergebnisse der Simulation und des Demonstrationssystems an die GNC-Komponente zurückgeführt werden, beispielsweise die Daten realer oder simulierter Sensoren.

Um die Schnittstellen zwischen dem DFKI-Kernsystem und den Steuerungs- und Simulationskomponenten von ASTRIUM festzulegen, fand am 16.04.2010 eine Besprechung mit Astrium statt. Der wichtigste neue Punkt hierbei war die Einigung auf den Einsatz eines dSPACE-Echtzeit-Simulationssystems, in das sowohl GNC, RCS-Simulation und die Orbitdynamiksimulation als auch das am DFKI entwickelte Kernsystem der Simulation integriert wurden. So können alle Komponenten reibungslos ich Echtzeit interagieren. Die Simulationen von GNC und RCS/Orbitaldynamik laufen auf jeweils einem Prozessorkern des dSPACE-Systems, auf einem dritten Prozessorkern läuft das DFKI-Kernsystem. Mit dem beschafften dSPACE-System ist dies möglich, da es über 3 Prozessorkerne für Echtzeitsimulation und einen vierten Kern für Ethernet-Kommunikation über Linux verfügt. Der vierte Kern kann so zur Steuerung von Hardware genutzt werden, beispielsweise des KUKA-Roboters über Ethernet und RSI.

Die RCS-Simulation läuft seit dem dritten Quartal 2010 bei ASTRIUM in Simulink und in deren dSPACE-System, womit eine wichtige Voraussetzung zur Portierung dieser Simulationskomponente auf das dSPACE-System des DFKI geschaffen war. Dieser Schritt war erforderlich, um die RCS-Simulation in das auf dSPACE basierende Simulations-Kernsystem des DFKI zu integrieren. Diese Integration war im dritten Quartal 2010 als Konzept bereits vorbereitet und wurde zusammen mit der Integration der Orbitdynamiksimulation und der GNC-Komponenten in das Kernsystem vorgenommen.

Bei der Darstellung von Missionen mit dem Bewegungssystem sind viele Interaktionen zwischen den Komponenten der Anlage selbst und den autonomen Systemkomponenten vorhanden. Diese galt es im Rahmen von Experimenten aufeinander abzustimmen, so dass eine Systemstabilität und eine möglichst hohe stationäre Genauigkeit erreicht wird.

Die Daten des Motion Tracking Systems (MTS) haben beispielsweise einen großen Einfluss auf das System, da sie die Positionen des CableRobots und des Clients beeinflussen. Durch die Rotationskorrektur wird die ungewollte Rotation des CableRobots, die durch das MTS gemessen wird, ausgeregelt, in dem der Client seine Rotation und damit zwangsläufig auch seine Position ändern muss sowie auch der CableRobot seine Position verändert. Diese Veränderung ruft jedoch eine Rotationsänderung hervor, die wiederum eine Positionsänderung erzeugt. Da die Änderungen der Rotation aber deutlich kleiner als die Änderungen der Position sind, ist das System an sich stabil. Damit große Schwingungen verhindert werden, wurde ein geeigneter PID Regler eingefügt. Je mehr Filterwirkung der Regler jedoch besitzt, desto langsamer reagiert das

Gesamtsystem auf gewollte Positionsänderungen.

Im Closed-Loop Betrieb werden die Korrekturen zusätzlich durch die gewollten Positionsänderungen der Guidance Navigation Control (GNC) überlagert. Durch Abweichungen der LIDAR Lageschätzung sendet die GNC Befehle zu den simulierten Triebwerken, die anhand der Orbitaldynamik neue Korrekturen erzeugen. Dieser Regelkreis plus weitere Regelkreise wie z.B. die der Aktuatoren oder Arbeitsraumverbesserungen können schnell für Schwingungen sorgen, die zum einen die Genauigkeit verschlechtern und zum anderen ein instabiles Verhalten verursachen können. Anhand der Closed-Loop Testergebnisse wurde im ersten Quartal 2012 gezeigt, dass ein stabiles Verhalten erzeugt werden konnte. Da sich die Systeme (Servicer/Client) jedoch nicht berühren, war keine Rückkopplung aus dem Bewegungssystem zurück in die Orbitaldynamik erforderlich.

	abs. Pos.-Fehler (RvD)	zeitliche Verzögerung	Stabilität	Rauschen
Interpolation	ca. 40 μm	100 ms	immer stabil	sehr klein
Streckenpolation	< 20 μm	ca. 40 ms	mögliche dyn. Abhängigkeit	nicht vorhanden
Extrapolation 2. Grad	k. A. (sehr klein)	0 ms	stabil bei $T_{GNC} \geq 2T_{Dspace}$	klein
Extrapolation 3. Grad	k. A. (theoretisch Null)	0 ms	stabil bei $T_{GNC} \geq 2T_{Dspace}$	mittel

Tabelle 14: *Daten der Polationsmethoden zum Ausgleich der verschiedenen Taktzeiten*

3.1.16 AP42300-D: Orbitdynamiksimulation

Zum Erreichen des Projektziels von INVERITAS, ein Rendezvousmanöver mittels Onlinesimulation und Hardware-in-the-Loop, müssen die verschiedenen Komponenten, welche teils von Astrium und teils vom DFKI entwickelt wurden, alle in einem Simulink-Modell vereint werden, welches dann verteilt auf drei dSPACE-Kerne die Simulation ermöglicht. Hierbei handelt es sich um die Orbitalsimulation (Astrium), das GNC (Astrium) und dem TestbedControl (DFKI). Die Synthese der drei Modelle wurde mithilfe eines übergeordneten Simulink-Modells realisiert, in welchem die vom Real-Time-Workshop erzeugten ausführbaren Dateien hereingeladen werden. Die Kommunikationswege können in dem übergeordneten Simulink-Modell determiniert werden.

Die von den Systemblöcken GNC und Orbitaldynamik gelieferten Daten besitzen eine von der Anlage abweichende Frequenz (siehe oben), welche stets geringer ist als die Anlagenfrequenz (f=250GradHz). Um diese Inkompatibilität zu beseitigen, wurden insgesamt vier Blöcke entwickelt, welche vor der weiteren Nutzung der Daten in 12D/xD verwendet werden.

Es handelt sich hierbei um eine lineare Interpolation, eine Extrapolation 2. Grads, eine Extrapolation 3. Grads sowie eine Streckenpolation (mit extrapolierendem Verhalten)(Tab. 14)

Die verschiedenen Methoden können je Bedarf und Simulationsschwerpunkt ausgewählt werden. Als besonders stabil und ausreichend genau hat sich die lineare Interpolation erwiesen, welche als einzige fix in das System integriert worden ist. Die anderen Methoden sind fertig entwickelt und könnten bei Bedarf schnell in das System integriert werden. Alle Methoden wurden bereits in Soft- und Hardware verifiziert.

Die Orbitaldynamik ist in der Lage die errechneten Daten in zwei verschiedenen Koordinatensystemen zu liefern: Dem LVLH und dem ECI. Das LVLH-System besitzt seinen Mittelpunkt in einem der beiden Objekte (Servicer oder Client) und liefert uns orbitale Relativdaten. Das ECI-System besitzt seinen Mittelpunkt im Erdkern und liefert uns somit Absolut-Daten in Bezug auf die Erde. Ebenso ist es (unabhängig vom Koordinatensystem) möglich die Daten in den mathematischen Formaten namens Dreh-Cosinus-Matrix oder Quaternionen auszugeben. Im Verlauf der Entwicklungen wurden alle vier Möglichkeiten programmiert und erfolgreich getestet. Aus Gründen der Einheitlichkeit hat man sich am Ende für das Koordinatensystem ECI und das Format Dreh-Cosinus-Matrix entschieden, welches nun als einziges im Einsatz ist. Alle anderen Methoden sind allerdings ebenfalls bei Bedarf mit relativ geringem Aufwand einsetzbar.

3.1.17 AP43100: Evaluierung des aktuellen Standes der Forschung

Es existiert mittlerweile eine Vielzahl von technischen Greifsystemen in den verschiedensten Branchen. Die größte Verbreitung finden Greifsysteme bisher in der Industrierobotik, Unterwasserrobotik sowie an wissenschaftlichen Instituten. Die Nutzung von Greifsystemen in der Service-, Medizin- sowie Weltraum-Robotik steht hingegen noch an ihrem Anfang.

Grundsätzlich kann bei Greifsystem zwischen industriellen Greifern und den in der Entwicklung noch ziemlich neuen künstlichen Händen unterschieden werden. Industrielle Greifer zeichnen sich besonders durch ihre hohe Robustheit, geringe Komplexität, hohe Spezialisierung auf Objekte sowie vergleichsweise geringe Kosten aus. Darüber hinaus sind sie über Jahrzehnte erprobt und besitzen somit einen hohen Zuverlässigkeitsgrad. Typische industrielle Greifer sind z. B.:

- Zwei-Finger-Parallelgreifer
- Zwei-Finger-Winkelgreifer
- Drei-Finger-Zentrischgreifer
- Flachsauger
- Faltenbalgsauger

Nischenanwendungen finden darüber hinaus auch folgende Greifersysteme:

- Magnetgreifer
- Elektrostatische Greifer
- Kapillargreifer
- Gefriergreifer
- Klebstoffgreifer

Für einen Gebrauch dieser Greifsysteme in der Raumfahrt, zum Einfangen von Objekten, sind diese Systeme vor allem wegen ihres hohen Spezialisierungsgrades und der daraus folgenden geringen Flexibilität eher ungeeignet. Anwendungen dürften sich damit nur unter speziell kooperierenden Raumfahrtobjekten bieten.

Künstliche Hände hingegen bieten einen sehr viel höheren Grad an Flexibilität und sind durchaus in der Lage, sich an nicht genau bekannte Objekte anzupassen. Abbildung 64 zeigt einige Beispiele solcher bis jetzt nur in der Forschung existierenden Hände.

Bei den oberen drei Händen in Abb. 64 handelt es sich um modulare Bauweisen (Aktoren sind in der Hand integriert), bei den unteren zwei um integrierte Bauweisen (Aktoren sitzen außerhalb der Hand). Beide, integrierte sowie modulare Bauweisen, eignen sich durch ihre adaptiven Fähigkeiten prinzipiell für Fangsysteme in der Raumfahrt. Jedoch sind auch diesen Systemen Grenzen gesetzt. Sie besitzen zwar einen hohen

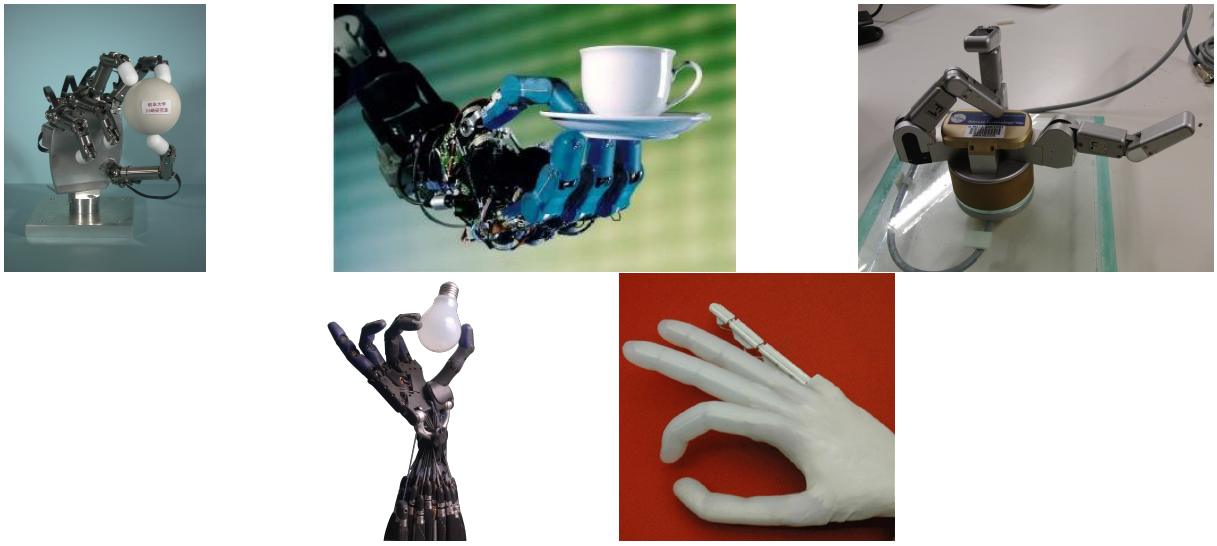


Abbildung 64: *künstliche Hände (von links oben: Kawasaki & Mouri Laboratory, DLR,TU-Berlin, Shadow Robot Company, Laboratory of Automation and Robotics*

Grad an Beweglichkeit und Manipulationsmöglichkeiten, haben aber einen relativ kleinen Arbeitsbereich (Einfangbereich). Besitzt ein Raumfahrtobjekt greifbare mechanische Teile, können sich diese Hände (mit der entsprechenden Bildverarbeitung, Navigation, etc.) wahrscheinlich an solche Objekte klammern und sie manipulieren. Weist hingegen ein solches Objekt nur ebene Flächen ohne relevante geometrische Abweichungen auf, verlieren sie, ähnlich wie eine menschliche Hand, ihr Greifvermögen. Dies könnte zum Beispiel beim Einfangen von Trümmerteilen oder natürlichen Weltraummüll der Fall sein. Zu erwähnen sei hier noch, das es unter den künstlichen Händen noch eine dritte Gruppe, die der unteraktuierten Hände gibt, welche durch flexible Elemente mit derselben Anzahl an Aktuatoren eine größere Anzahl an Freiheitsgraden besitzen können. Die Probleme für das Einfangen von Objekten im Weltraum sind aber auch hier grundlegend dieselben.

Über Industrie und Wissenschaft hinaus befinden sich bereits einige wenige Robotersysteme im Betrieb der Raumfahrt. Unter ihnen

- Rokviss (RObotik-Komponenten-Verifikation auf der ISS) - der deutsche Roboterarm auf der ISS
- Remote Manipulator System (RMS) oder Canadarm - ein kanadischer Roboterarm an Bord der Space Shuttles
- Space Station Remote Manipulator System (SSRMS) oder Canadarm2 - ein kanadischer Roboterarm an der Außenwand der ISS
- Special Purpose Dexterous Manipulator (SPDM), auch Canada Hand oder Dextre - ein kanadisches Robotersystem gekoppelt am SSRMS

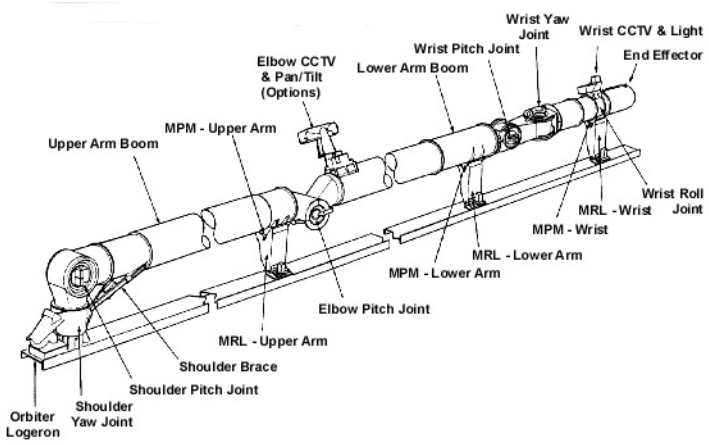


Abbildung 65: *Der RMS "Canadarm" (IEEE Canada)*

Rokviss

Der deutsche Roboterarm, entwickelt am Institut für Robotik und Mechatronik des DLR, wurde zum Testen von hoch integrierten Roboterkomponenten im Weltraum sowie neuer Kontrollverfahren (automatisch als auch telepräsent) gebaut. Er besteht aus einem Roboterarm mit zwei Freiheitsgraden, einen Metallfinger an der Spitze des Arms sowie einer Stereo- und Monokamera. Für das Einfangen oder Greifen von Objekten ist er ungeeignet.

RMS

Der kanadische RMS dient vorrangig zum Aus- und Einladen von Nutz- oder Abfalllasten zum oder vom Space-Shuttle. Seine Länge beträgt über 15 Meter und er besitzt sechs Freiheitsgrade. Mit ihm ist man bereits in der Lage, über einen speziellen Greifer Satelliten einzufangen und wieder auszusetzen. Somit ist er für das Einfangen und Greifen von Objekten durchaus geeignet. Allerdings ist dies mit dem Canadarm nur mit bemannten Operationen möglich und auch mur mit kooperierenden Satelliten (kontrollierte Bewegung, greifbar). Abbildung 65 zeigt den Canadarm schematisch.

SSRMS und SPDM

Der SSRMS an der Außenwand der ISS wurde vor allem für Wartungs- und Reparaturarbeiten, sowie für den Zusammenbau der ISS konzipiert. Darüber hinaus dient er experimentellen Versuchen außerhalb der Raumstation. Am Ende des sogenannten Canadarm2 wurde die Canada Hand installiert. Diese verfügt über zwei komplexe Roboterarme mit je sieben Gelenken an deren Enden sich eine Auswahl an verschiedenen Werkzeugen und Kameras befindet. Es ist ebenso möglich, weitere Geräte bei Bedarf aufzunehmen. Obwohl das System vorrangig zu Reparaturzwecken sowie zur Unterstützung wissenschaftlicher Experimente im leeren Raum konzipiert wurde, ver-

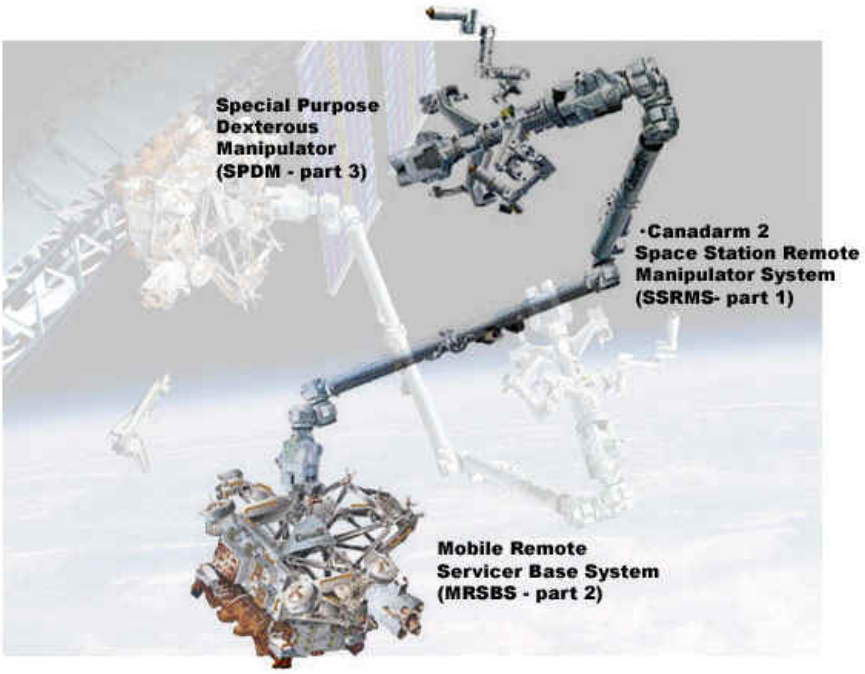


Abbildung 66: *Das Gesamtsystem der ISS (Canada Connects)*

fügt es jedoch über eine sehr große Flexibilität, mit der es theoretisch ebenfalls möglich sein dürfte andere Objekte einzufangen. Die Bedienung muss jedoch auch hier vor Ort erfolgen. Abbildung 66 und 67 zeigen den Canadarm2 und die Canada Hand.[4]

Industrie, Wissenschaft und Raumfahrttechnologien bietet sich somit schon eine Fülle an möglichen Greif- und Fangsystemen, von denen die meisten einen sehr hohen Spezialisierungsgrad aufweisen (Industrie, Raumfahrt), hingegen andere eine sehr hohe Generalisierung anstreben (Wissenschaft). Werkzeuge aus Industrie und Wissenschaft sind nur sehr bedingt für das Einfangen und/oder Greifen von Objekten im Weltraum geeignet, wobei gerade die wissenschaftlichen Entwicklungen hohes Potential bieten. Die bis jetzt geeignetsten Systeme befinden sich bereits, wie nicht anders zu erwarten, in der Raumfahrt, sind aber auch auf ihre Aufgaben relativ spezialisiert und weniger zum Einfangen frei beweglicher Objekte konzipiert. Auch in der Unterwasser- und Medizinrobotik findet man eine Reihe von Manipulatoren, deren Aufgaben aber meist der stärksten Spezialisierung unterliegen. Für das Ergreifen frei schwebender Objekte sind auch sie nicht geeignet, wobei sich das Medium Wasser hervorragend zum Testen und Evaluieren solcher Systeme eignen würde.

Neue Ansätze für das Abfangen vor allem unkontrollierter und nicht kooperierender Objekte müssen geschaffen werden, um eine Service-Dienstleistung für alte oder defekte Satelliten oder auch das Beseitigen von Meteoriten oder Weltraumschrott bewerkstelligen zu können. Erste Ansätze bieten z. B. die Missionen DEOS und SmartOLEV. DEOS (DEutsche Orbitale Servicing Mission) ist eine deutsche Technologiedemons-

[4] DLR, NASA

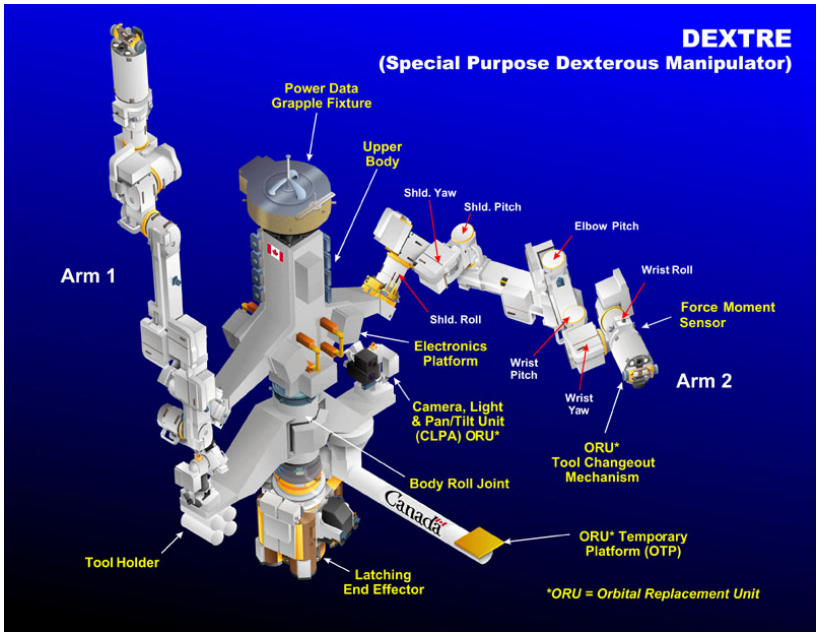


Abbildung 67: *Die Canad Hand (Gizmodo)*

tration zum automatischen (unbemannten) Einfangen von defekten, unkontrollierbaren Satelliten im LEO-Orbit, wobei das Andockverfahren über einen speziellen Roboterarm erfolgen soll. Auch Wiedereintrittsversuche sollen erprobt werden. SmartOLEV (OLEV-Orbital Lifetime Extension Vehicle) ist der kommerzielle Versuch einer Lebenszeitverlängerung von geostationären Satelliten. Hierbei soll ein Servicer an den Zielsatelliten andocken und dessen Lage- und Positionskontrolle übernehmen. Der Andockvorgang soll hierbei über die Antriebsdüse des Zielsatelliten erfolgen, da dies das einzige Bauteil aller Satelliten ist, das einer Normierung unterliegt.

Abbildung 68: *DEOS (OHB-Technology)*

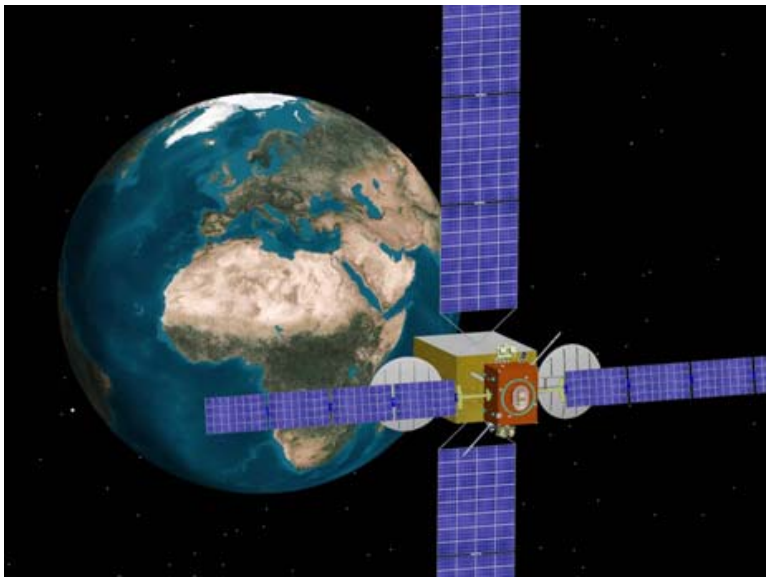


Abbildung 69: *SmartOLEV (DLR)*

3.1.18 AP43200: Konzepte für neue Capture-Systeme

Die evolutionäre Robotik beschäftigt sich mit Algorithmen zum Entwerfen von neuartigen Hardware- und Software-Lösungen für robotische Systeme [1, 4, 5, 9, 10, 18]. Der wesentliche Vorteil dieser Ansätze ist, dass sie unvoreingenommen einen Suchraum von Lösungen erforschen und nicht durch bestehende Lösungen und Methoden eingeschränkt sind. Menschliche Entwickler haben im Gegensatz selten die Möglichkeit Lösungen zu betrachten die auf den ersten Blick völlig abwegig erscheinen. Dennoch können die Ansätze der evolutionären Robotik nicht die Erfahrungen und das Fachwissen von Experten ersetzen. Daher empfielt sich beim Entwerfen von robotischen Systemen für neue Einsatzgebiete eine Co-Entwicklung bei der die Systeme von Experten entwickelt werden die einen Input von neuen Ideen und Konzepten aus evolutionär entworfenen Lösungen gewinnen können. Der Zugewinn durch die evolutionären Verfahren hängt dabei stark mit der Formulierung des Problems, der Performanzbewertung und der generischen Beschreibung der Lösungen ab. Dabei muss ein Kompromiss aus einem möglichst kleinem Suchraum und dennoch vielen möglichen Variationen der Lösungen gefunden werden.
Zur generischen Beschreibung von möglichen Lösungen wurde ein Konzept entworfen, welches das Entwickeln von mechanischen Strukturen auf Basis von Verbindungen und Gelenken erlaubt. Dabei können die Verbindungen in ihrer Dimension variieren und wahlweise linear Aktuatoren zur Längenänderung beinhalten. Die Gelenke können über ein bis drei passive oder angetriebene Achsen verfügen.

Um höchstmöglichen Ertrag aus den evolutionär entwickelten Lösungen gewinnen zu können soll eine “Insel” basierte Evolution durchgeführt werden, bei der mehrere Evolutionen parallel nach Lösungen suchen und unter bestimmten Bedingungen Informationen zwischen den einzelnen Evolutionen ausgetauscht werden können. Dieser Ansatz erlaubt eine höchstmögliche Skalierbarkeit der Parallelisierung und somit den Einsatz von beliebig vielen Prozessen (Rechnern) zum Finden neuartiger Konzepte.
Die Durchführung des evolutionsbasiertem Ansatzes wurde in AP 43300 behandelt.

Neben der rein simulationsgestützten Entwicklung wurden im Vorhinein neue Konzepte anhand von bestehenden Technologien betrachtet. Im folgenden sind diese zusammengefasst.

Für das Design neuer Capture Systeme wurden zunächst folgende Gegebenheiten festgelegt:

- Die Position des Capture Systems zum Satelliten ist fest.
- Das Capture System ist in der Welt fixiert.
- Der Satellit stellt eine im Raum schwebende Einheit ohne initiale Bewegung dar.

Die Anforderungen an das Capture System sind gegeben durch:

- Das Capture System soll Satelliten fixieren können.
- Bei dem Capture Vorgang soll der Satellit unversehrt bleiben.

Ideen aus Brainstorming:

- Den Satellit über eine Art "Harpune" einfangen.
  Da die Satelliten möglichst unbeschädigt bleiben sollen, stellt diese Lösung keine Option für ein Capture Verfahren dar.
- Den Satellit über lange Arme mit vielen Gelenken "eindrehen" (Oktopus ähnlich).
  Dieser Ansatz wird näher erläutert unter Option (A).
- Ein textiles Netz kontrolliert über Satellit werfen.
  Diese Ansatz ist nur schwer zu beurteilen. Das Verhalten von Textilien oder anderen Netzmaterialien unter den jeweiligen Umgebungsbedingungen wäre zuvor zu testen.
- Ein Lasso oder ein Gürtel um den Satelliten werfen.
  Könnte in einer mechanischen Umsetzung dem Ansatz mit langen Armen und vielen Gelenken entsprechen.
- Kleine Steuersatelliten die ausschwärmen, an dem Satelliten andocken und ihn dann über ihre Düsen kontrollieren können.
  Dieser Ansatz wird näher erläutert unter Option (B). item Große Greifer die im Inneren elastisch sind. Dies kann über geschickt verknüpfte Metallbleche erreicht werden (siehe flexible DLR-Räder [20]).
  Dieser Ansatz wird näher unter Option (C) erläutert.
- Finray (FinGripper) [7]: Umschließt über/durch Struktur beim Zugreifen den Satelliten oder Teile von ihm.
  Opiton (D).
- Kontaktlos den Satelliten beeinflussen:
  - Satelliten umschließen und über Düsen, Magnetismus oder Photonenenergie (Prinzip einer Lichtmühle) den Satelliten in Position halten.
- Über eine Art des Klebens den Satelliten greifen:
  - Stickybot Füße [13]
    Sehr interessanter Ansatz über Van-der-Waals Kräfte. Wird unter Option (E) näher erläutert.
  - Haftflüssigkeit
    Schwer zu diskutieren und auf Seiten einer Simulation, ähnlich oder (je nach Abstraktionslevel) gleich zu Option (E).

Konzept Option (A): Oktopus Arme

Die Idee dieses Konzeptes ist es, mit mehreren Fangarmen Objekte unbekannter Form und Größe umschließen zu können. Dabei sollen die Objekte nicht zu großem Druck ausgesetzt werden müssen. Da die Umsetzung der Fangarme den Umweltbedingungen des Orbits ausgesetzt werden würde, soll eine Lösung auf Basis von Elektromotoren und Mechanischen Verbindungen erstellt werden. Aus diesem Grund ähnelt das Konzept der Arme einem klassischem Manipulatorarm mit wahrscheinlich

bis zu 30 Gelenken. Um den Druck der Arme auf das Objekt minimal zu halten kann einerseits eine Impedanz-Kontrolle auf Seiten der Gelenksteuerung verwendet werden und andererseits können die Soll-Kontaktpunkte der Arme mit metallischen Federn und Dämpfern ausgestattet sein. Abbildung 70 ist eine erste grafische Darstellung des Konzeptes mit drei Fangarmen und zwei Gelenken pro Kugel.
Nachteile des Konzeptes sind die große Anzahl an benötigten aktuierten Gelenken und der damit verbundene Energie- und Steuerungsaufwand.

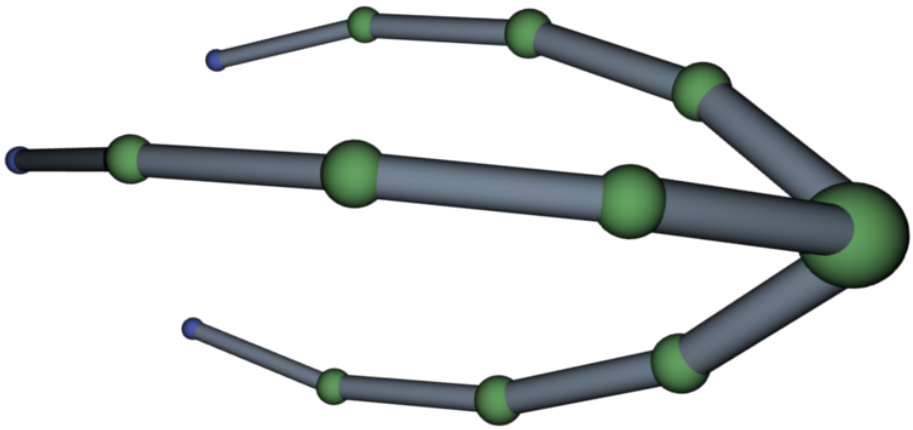


Abbildung 70: *Konzept Option (A).*

Konzept Option (B): Kleine Service-Raketen

Bei diesem Konzept handelt es sich um eine Lösung aus dem Bereich der Schwarmrobotik. Mehrere kleine Service-Raketen sollen kooperativ an dem Satelliten andocken um über eine Schubdüse auf den Satelliten Einfluss nehmen zu können. Das Konzept ähnelt den Schleppern die große Schiffe in einen Hafen geleiten. Problem bei dem Konzept ist, dass jede Service-Rakete eine eigene relative Geschwindigkeit zu dem Satelliten hat und die Rotationsachse der Basis verlassen muss. Dadurch müssen die Service-Raketen zum kontrollierten Andocken sehr schnell sein. Außerdem ist die Steuerung und Koordination der Service-Raketen sehr kompliziert.

Konzept Option (C): Großer Greifer mit elastischem Innerem

Abbildung 72 stellt das Konzept eines großen Greifers dar. Nicht in der Abbildung enthalten sind metallische Strukturen die eine anpassbare Innenfläche der Greifer

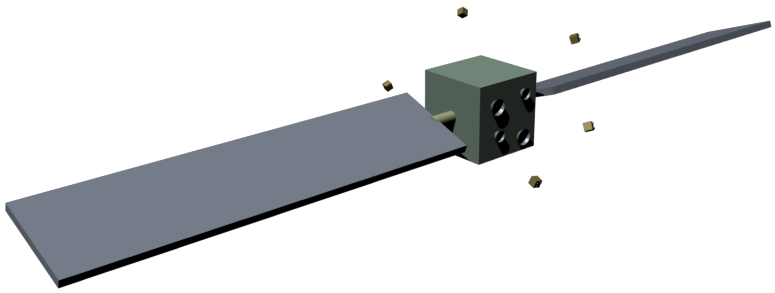


Abbildung 71: *Konzept Option (B).*

realisieren und somit den Satelliten in sich einschließen und fixieren. Eine mögliche elastische Struktur könnte den Konzepten der flexiblen Rover Räder der DLR ähnlich sein [20].

Konzept Option (D): FinGripper

Bei dem Konzept der Finray Struktur [7] schließen sich die Finger durch Druck auf der Innenseited des Greifers. Dieses Prinzip könnte für einen großen Greifer verwendet werden und würde eine Mischung aus der Option (A) und Option (C)darstellen.

Konzept Option (E): Stickybot Füße

Die Füße des Stickybot Roboters [13] ermöglichen Halt auf glatten Oberflächen auf Basis der Van-der-Waals Kräfte. Dieses Prinizip ist aus der Biologie inspiriert (Vorbild sind die Füße von Echsen). Dabei besteht die Oberfläche der Füße aus vielen mikroskopisch kleinen Haaren die sich in Zugrichtung mit glatten Oberflächen auf molekularer Ebene verhaken können. Bei dem Stickybot ist das Material der Füße aus Kunststoff, daher müsste für eine Anwendung im Orbit zunächst ein geeignetes Material, welches die gleichen Eigenschaften hat und den Weltraum-Umgebungsbedingungen trotzen kann, entwickelt werden. Grundsätzlich wäre der Greifvorgang bei diesem Ansatz mit dem fixieren eines Satelliten über einen Magneten am Endeffektor vergleichbar. Da Satelliten im Allgemeinen keine magnetischen Eigenschaften aufweisen scheiden auf Magneten basierende Greifsysteme als Konzept aus.

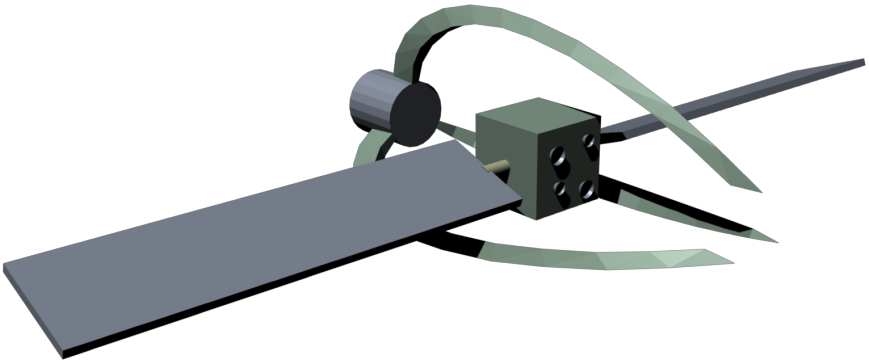


Abbildung 72: *Konzept Option (C).*

Abbildung 73: *Konzept metallische Elastizitäten am Beispiel der flexiblen DLR Rover Räder.*

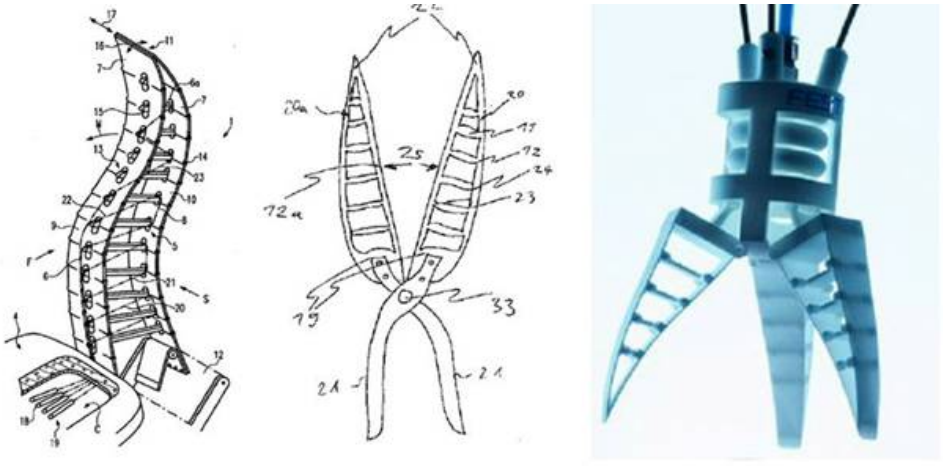


Abbildung 74: *Konzept Option (D).*

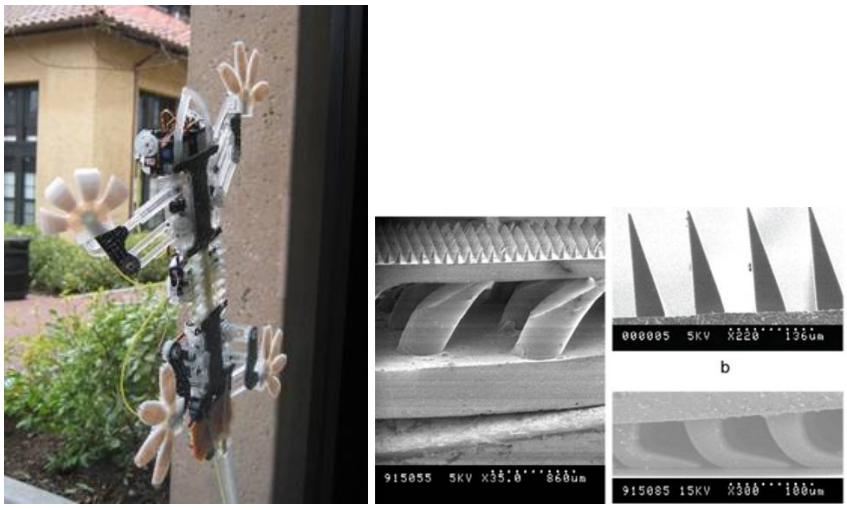


Abbildung 75: *Konzept Option (E).*

3.1.19 AP43300: Simulation ausgewählter Capture Systeme

Für dieses Arbeitspaket wurde zunächst die inselbasierte Co-Evolution implementiert. Dazu wurde über das hauseigene Netzwerkframework “ADRF” eine serverbasierte Architektur entwickelt. Dabei wird eine Anwendung als Evolutionsserver gestartet und beliebig viele Simulationen können als Clients verteilt auf mehreren Rechnern gestartet werden.

Der Evolutionsserver kann mehrere Evolutionen [17] (Insel-Evolutionen) gleichzeitig verwalten und die zu bewertenden Individuen über das Netzwerk verteilen. Bei der Verteilung werden alle Individuen allen Clients zur Verfügung gestellt. Die Clients wählen zufallsbasiert ein Individuum aus welches getestet wird und teilen die Auswahl den anderen Simulationen mit. Individuen können von mehreren Simulationen bewertet werden, wodurch eine Redundanz erzeugt wird und ein Rechnerabsturz nicht die gesamte Evolution blockieren kann. Individuen die noch nicht an einen Client vergeben sind werden bei der Auswahl allerdings immer bevorzugt. Die gesamte Kommunikation wird von “ADRF” organisiert und erlaubt ein dynamisches Entfernen und Hinzufügen der Simulationen. Das gesamte Framework wurde über 24 Stunden mit drei Insel Evolutionen und ca. 30 Clients getestet. Dabei wurden knappe 6 Millionen Individuen einer Testanwendung bewertet.

Mit Hilfe der Festkörpersimulation MARS wurde der Entwurf eines Oktopus ähnlichen Greifarms simuliert und über die Insel basierte Co-Evolution optimiert. Dafür wurde eine parametrisierte Beschreibung des Armes und der Steuerung entworfen. Dabei bestimmen 14 Parameter die Morphologie und das Verhalten des Armes. Die Parameter sind in der folgenden Aufzählung aufgeslistet:

1. num-elements: Definiert die Anzahl der Elemente (Kapseln) aus denen der Greifarm besteht.
2. initial_length: Definiert die Länge des ersten Elements des Greifarms.
3. initial_radius: Definiert den Radius des ersten Elements des Greifarms.
4. initial_speed1: Definiert die Geschwindigkeit der ersten Achse des ersten Elements des Greifarms.
5. initial_speed2: Definiert die Geschwindigkeit der zweiten Achse des ersten Elements des Greifarms.
6. initial_torque1: Definiert die Drehmomentgrenze der ersten Achse des ersten Elements des Greifarms.
7. initial_torque2: Definiert die Drehmomentgrenze der zweiten Achse des ersten Elements des Greifarms.
8. length_factor: Der Faktor um den die Länge nach jedem Element verändert wird.
9. radius_factor: Der Faktor um den der Radius nach jedem Element verändert wird.
10. speed_factor1: Der Faktor um den die Gelenkgeschwindigkeit der ersten Achse nach jedem Element verändert wird.

11. speed_factor2: Der Faktor um den die Gelenkgeschwindigkeit der zweiten Achse nach jedem Element verändert wird.
12. torque_factor1: Der Faktor um den die Drehmomentgrenze der ersten Achse nach jedem Element verändert wird.
13. torque_factor2: Der Faktor um den die Drehmomentgrenze der zweiten Achse nach jedem Element verändert wird.
14. num_inactive_motors: Bestimmt wie viele Motoren des Greifarm nicht angesteuert werden sollen. Dabei wird mit dem ersten Element angefangen zu zählen.

Abbildung 76 zeigt eine Skizze des Greifarms. Das erste Objekt (Servicer) stellt den Träger des Greifarms dar. Die Objekte o_1 - o_n sind die einzelenen Elemente des Greifarms die um einen konstanten Faktor kleiner werden. Die eingezeichnete Box (Satellite) stellt das zu fixierende Objekt dar. Die Elemete des Greifarms sind jeweils mit zwei angetriebenen Freiheitsgraden verbunden. Dabei liegt der erste Freiheitsgrad immer auf der x Achse und der zweite Freiheitsgrad erlaubt eine Rotation um die y Achse.

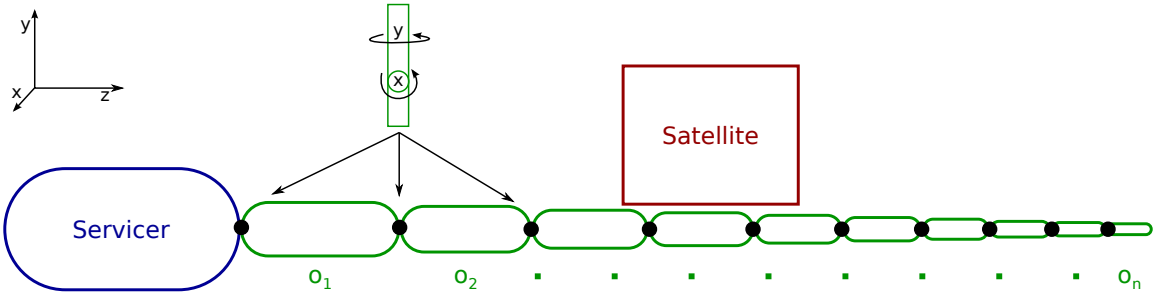


Abbildung 76: *Skizze des für die Simulation parametrisierten Greifarms.*

Für die Bewertung der bei der Evolution entwickelten Parameterkombinationen wird zunächst der durch einen Parametersatz bestimmte Arm in der Simulation erzeugt. Das Testen des Parametersatzes wird beendet und die Bewertung fällt schlecht aus wenn der erzeugte Greifarm nicht lang genug ist den Satelliten zu erreichen. Wenn der Greifarm eine minimale Länge überschreitet wird die Simulation gestartet und die Motoren des Greifarms werden den Parametern entsprechend angesteuert. Der Greifprozess wird nach einer definierten Zeitspanne von 480 Sekunden (8 Minuten) beendet. Danach wird eine weitere Zeitspanne von 480 Sekunden eine Kraft auf den Satelliten gegeben. Die Entfernung die der Satellit durch die einwirkende Kraft zurücklegt wird als Bewertungskriterium für den Greifvorgang verwendet. Bei einem erfolgreichem Greifprozess sollte diese Entfernung gegen Null gehen. Zudem werden durchgehend die Kontaktkräfte zwischen dem Satelliten und dem Greifarm ermittelt. Diese Kontaktkräfte dienen als zusätzliches Bewertungskriterium und sollten bei erfolgreichen Greifprozessen minimiert werden. Durch die Schwerelosigkeit und die großen Massen (der Satellit ist mit einer Masse von einer Tonne angenommen) sind geringe Geschwindigkeiten notwendig. Eine zu impulsive oder unkontrollierte Kollision führt schnell zu einem entdriften des Satelliten. Diese große Zeitspanne stellt einen wesentlich größeren Simulationsaufwand dar im Vergleich zu den bisher im DFKI durchgeführten simulationsbasierten Optimierungen. Durch die benutzerfreundliche Verteilung der Evolutionen

auf mehrere Rechner konnten ca. 70 unabhängige Optimierungen durchgeführt werden, mit insgesamt ca. 448.000 vorgenommen Bewertungen. Die durchgeführten Optimierungen des Oktopus-Greifarmkonzeptes wurden unter Zuhilfenahme von “Growing Cell Structures” [8] klassifiziert. Dabei wurden die insgesamt 70 Evolutionsergebnisse in 12 Klassen unterteilt. Die drei Klassen mit den meisten Lösungen repräsentieren alle ein Greifkonzept wie es in Abbildung 77 dargestellt ist. Zudem sind die Lösungen mit den besten Bewertungen ebenfalls in diesen drei Klassen zu finden. Weitere Beispiele für Greifkonzepte die entwickelt wurden sind in Abbildung 78 zu sehen.

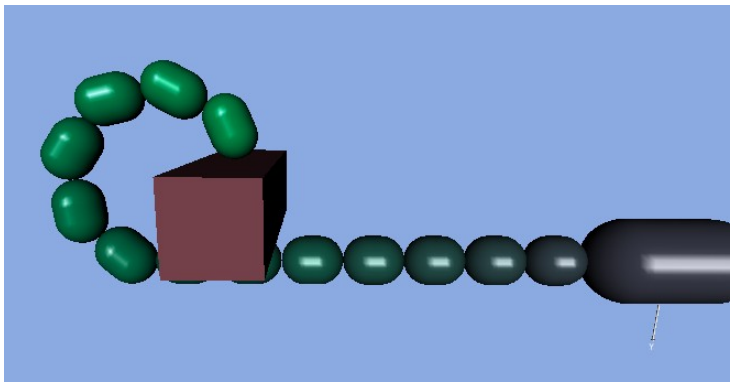


Abbildung 77: *Konzept der besten Greifarmlösungen.*

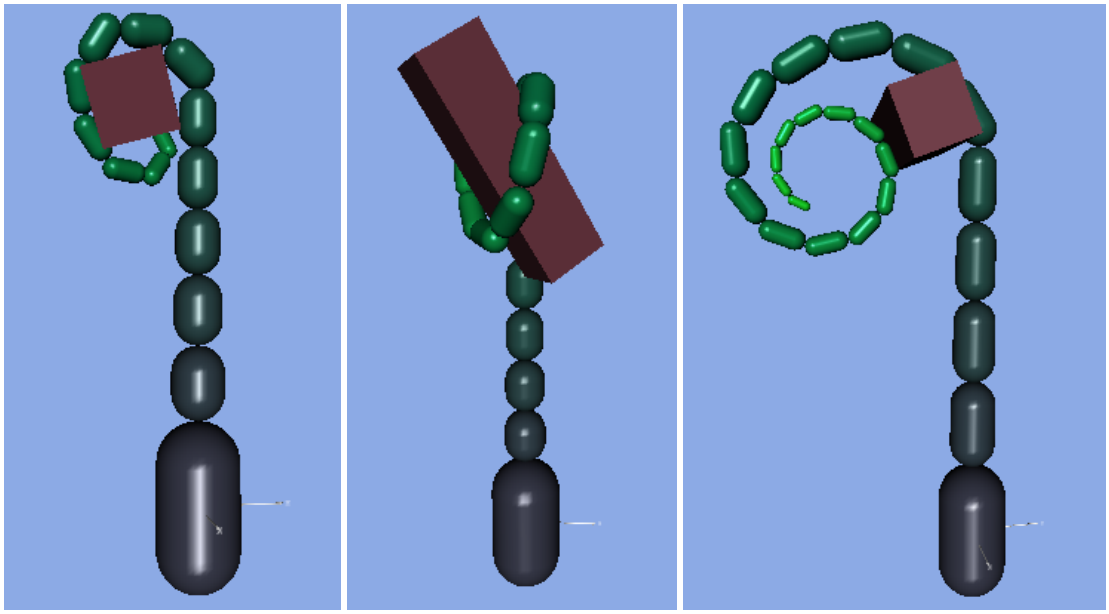


Abbildung 78: *Beispiele anderer Greifarmkonzepte mit schlechterer Bewertung.*

In einem zweiten Ansatz wurde eine Strukturoptimierung entwickelt auf Basis von Genetischen Algorithmen [21]. Im Gegensatz zu den Evolutionären Strategien wird ein Individuum nicht durch einen Parametervektor dargestellt sondern durch einen Graphen dessen Kanten und Knoten Parameter enthalten. Dieser Graph wird als Genotype bezeichnet und erst bei der Interpretation des Graphen und seiner Parameter wird der

Phenotype erzeugt. Bei der umgesetzten generalisierten Implementierung kann der Phenotype eine mechanische Struktur mit Größenangaben, Massen und Ausrichtung darstellen oder beispielsweise ein neuronales Netz wobei die Parameter des Graphen Gewichte der neuronalen Verbindungen und die Übertragungsfunktionen der Neuronen definieren. Im Gegensatz zu den Evolutionären Strategien ist neben der Mutation auch die Rekombination ein wesentlicher Bestandteil zum Erzeugen neuer Generationen. Bei der Rekombination werden die Kanten und Knoten zweier Graphen neu zusammengesetzt und erzeugen so ein Individuum der neuen Generation. Abbildung 79 stellt zwei Beispiele für die Rekombination dar.

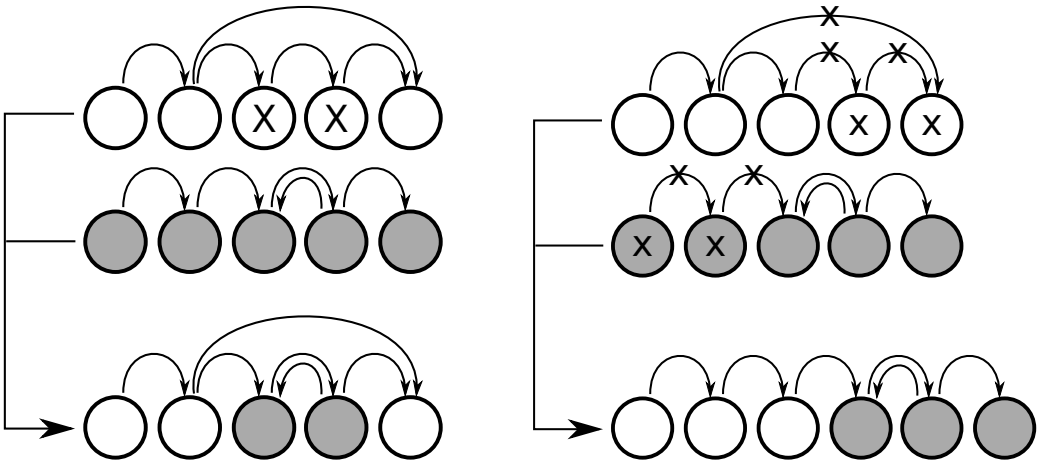


Abbildung 79: *Beispiele der Rekombinationsmöglichkeiten zum Erzeugen neuer Individuen aus zwei Eltern.*

Abbildung 80 zeigt ein vereinfachtes Beispiel wie aus dem Genotype ein Phenotype erzeugt wird welches eine Festkörperstruktur darstellt. Mit Hilfe dieses Ansatzes wurden Strukturen entwickelt die in der Lage waren den Satelliten (immer noch durch eine Box repräsentiert) derart zu umschließen, dass der Satellit in der Struktur fixiert war und somit manipuliert werden konnte. Abbildung 81 zeigt eine Lösung der Strukturentwicklung.

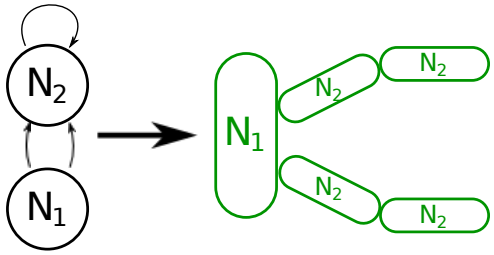


Abbildung 80: *Abbildung des Genotypes in eine Festkörperstruktur (Phenotype).*

Die Arbeiten an der evolutionören Entwicklung neuer Capturestrategien haben gezeigt, dass die simulationsgestützte Optimierung ein wertvolles Werkzeug darstellen kann. So stellt die Lösung der Fixierung des Satelliten durch einen langen einrollbaren Greifarm durchaus ein plausibles Konzept dar, welches nicht Bestandteil des Brainstormings für mögliche neue Konzepte war. Zudem ist zu erwarten, dass die Entwicklung

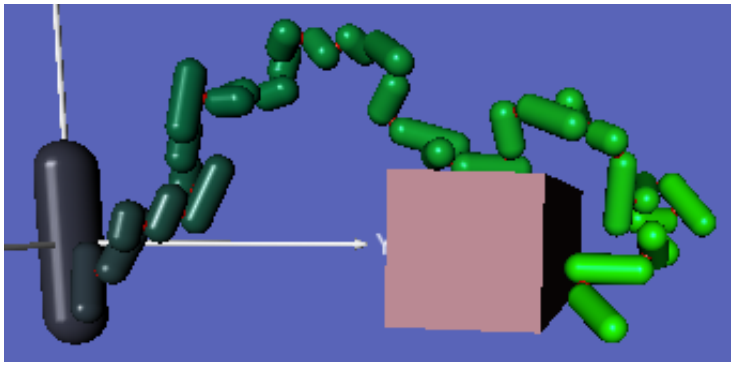


Abbildung 81: *Beispiel einer Lösung aus der Strukturoptimierung.*

von Strukturen in Kombination mit Aktuatoren, Sensoren und Verhalten ebenfalls neue Ideen hervorbringen kann. Allerdings stellt die Anwendung eine komplexe Aufgabe dar, die nicht innerhalb des Arbeitspaketes umgesetzt werden konnte. Die hier entwickelten Algorithmen haben vielversprechende Lösungen hervorgebracht bergen aber auch viel Raum für Erweiterungen und Verbesserungen.

3.1.20 AP44000-D: Gesamttest Simulation und Abschlussanalyse

Im Rahmen des Projekts INVERITAS wurde ein Simulink Modell erstellt, welches das Gesamtsystem von INVERITAS steuert. Dieses beinhaltet:

- Die Modelle von Astrium integriert als Binaries
- Alle Schnittstellen sowohl zu externen Komponenten via Ethernet (MTS, KUKA, MMS-PC, Mars Simulation, Vesta Viever) und CAN (CableRobot und dessen Z-Achse) als auch innerhalb des dSPACE Systems
- Verschiedene Inter- und Extrapolationsmethoden zum Anpassen der Astrium Kerne an den Steuerungskern des DFKI
- Eine Auswahlmöglichkeit der zu fliegenden Mission
- Die Steuerung der Anlage manuell oder durch das GNC
- Drei Kernsysteme zur Berechnung der KUKA und CableRobot Posen basierend auf der gegebenen Relativlage zwischen Servicer und Client
- Korrekturfunktionen für CableRobot und MTS zur Genauigkeitserhöhung der Anlage
- Eine GUI zur Steuerung aller Funktionen und zum Einstellen von Regelgrößen sowie missionspezifischen Parametern

Große Teile des Gesamtsystems wurden in diesem Arbeitspaket zusammen getestet. So wurde die Anlage bereits des öfteren dazu verwendet, verschiedene Missionsszenarien in der Explorationshalle mit Hardware darzustellen, um reale Sensordaten für das MMS aufzunehmen und darauf basierende Algorithmen zu testen.

Nachdem erfolgreich getestet werden konnte, dass alle Komponenten des Gesamtsystems zusammen funktionieren, konnten auch die abschließenden Open- und Closed-Loop Tests zur Verifikation der Visual Navigation (VN) und Guidance Navigation Control (GNC) durchgeführt werden.

Die Ergebnisse waren plausibel und entsprachen in vielerlei Hinsicht den erwarteten Genauigkeiten. Es gibt verschiedene Fehlerquellen, die sich konstruktiv und destruktiv überlagern. Abb. 82 veranschaulicht, wo Fehler auftreten können. Der Delay zwischen Sensordatenaufnahme und Verarbeitung der Sensorinformationen im GNC beträgt ca. 2 s, wird aber vom System detektiert, da Zeitstempel einer synchronisierten Systemzeit in allen Sensorinformationen mitgeliefert werden. Somit können jeweils die richtigen Informationen miteinander verglichen werden.

Im Pre-Processing kann eine Ungenauigkeit entstehen, da die Systemzeit des GNC von 10 Hz langsamer läuft als die interne Anlagensteuerungsfrequenz von 250 Hz. Es müssen Zwischendaten erzeugt werden, die durch geeignete Inter- und Extrapolationsmethoden so gering wie möglich gehalten werden.

Auch bei der Transformation von Relativposen vom Orbit in die Explorationshalle (hier bezeichnet als 12D/9D) entstehen Fehler. Diese werden nicht durch die mathematischen Umformungen erzeugt, sondern durch Rundungsfehler und vor allem durch

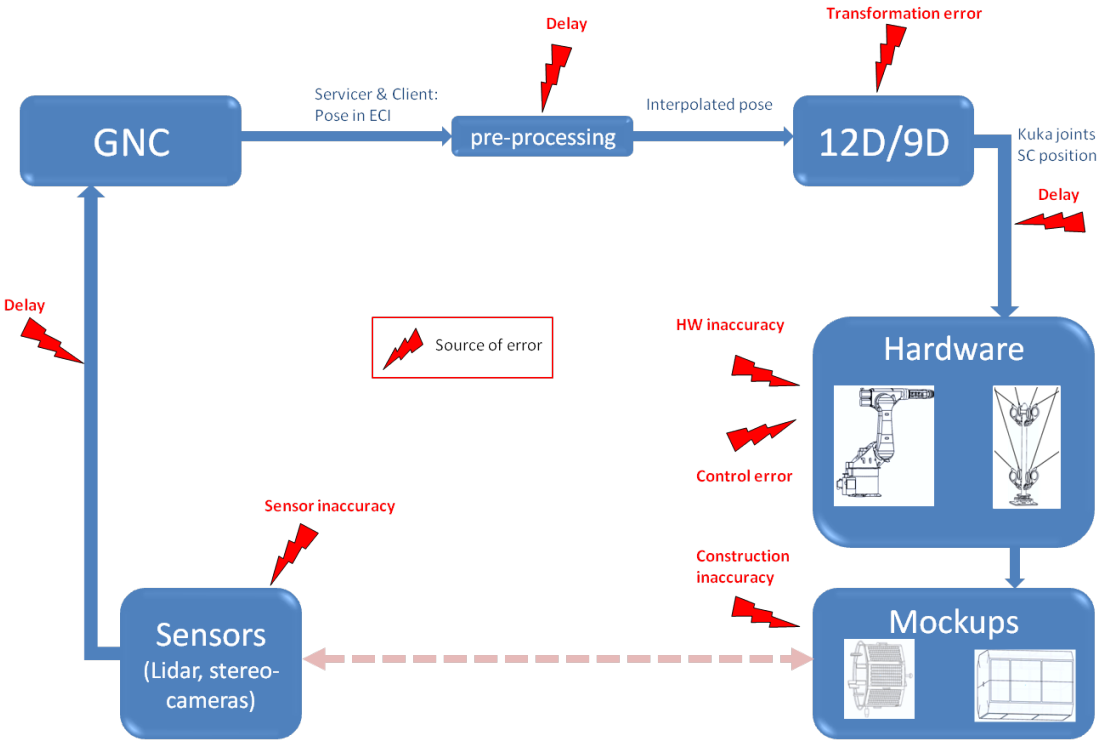


Abbildung 82: *Zusammenfassung möglicher Fehlerquellen*

Einschwingvorgänge von Reglern, die für ein weiches Verschieben von Servicer und Client zur Arbeitsraumoptimierung (A2/A3-Automatik, A4/A6-Automatik, Rotationskorrektur, z-Achsen Automatik) genutzt werden.

Abb. 83 zeigt, dass der größte Anteil für den Positionierungsfehler durch die Systeme selbst verursacht wird. Zum einen entsteht ein Delay zwischen Senden einer Sollposition und Erreichen dieser. Während der CableRobot die gewünschte Position in 12 ms erreicht sofern sie in einem Zeitschritt erreichbar ist, benötigt der KUKA zwischen 3 und 10 Takte, was einem Delay von bis zu 120 ms entspricht. Durch die maximale Verfahrgeschwindigkeit von ca. 50 mm/s, entsteht ein Fehler von maximal 5 mm. Dieser dynamische Aspekt überlagert den stationären Fehler der Einzelkomponenten, der durch die Testkampagnen mit einem Laser Tracker festgestellt wurde und durch Korrekturpolynome stark verbessert werden konnte.

Eine weitere Fehlerquelle sind die Mockups selbst, da sie Open-Loop gesteuert werden, d.h. es wird sich ausschließlich auf die Genauigkeit des Kuka verlassen und von einer starren Verbindung ausgegangen. Verformungen können nicht gemessen werden, da Marker für das MTS sich nicht auf dem Client befinden dürfen. Für den Servicer gilt ähnliches. Hier wird nur der CableRobot vom MTS getrackt. Ob die Positionierung des Servicers am CableRobot von den CAD Daten abweicht, wurde nicht exakt bestimmt. Insgesamt wird ein Fehler von ca. 1 cm angenommen.

Der reine Positionsfehler ergibt sich aus der Summe der Einzelkomponenten (Abb. 84(a)). Hinzu kommt ein Rotationsfehler der durch die Stellgenauigkeit der Kuka Achsen, der z-Achse des CableRobots und der ungewollten Rotationsbewegung

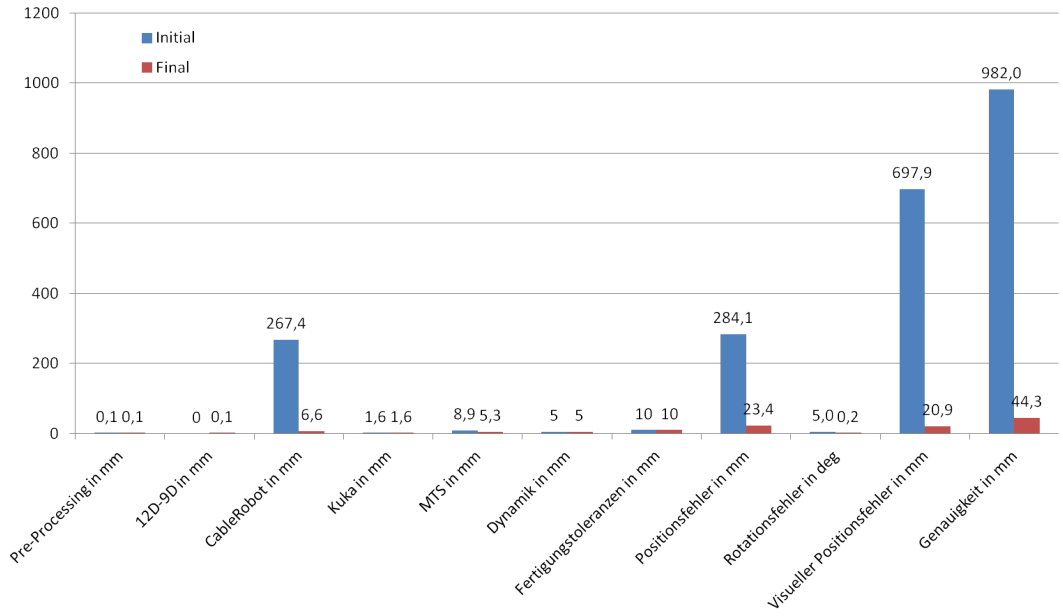
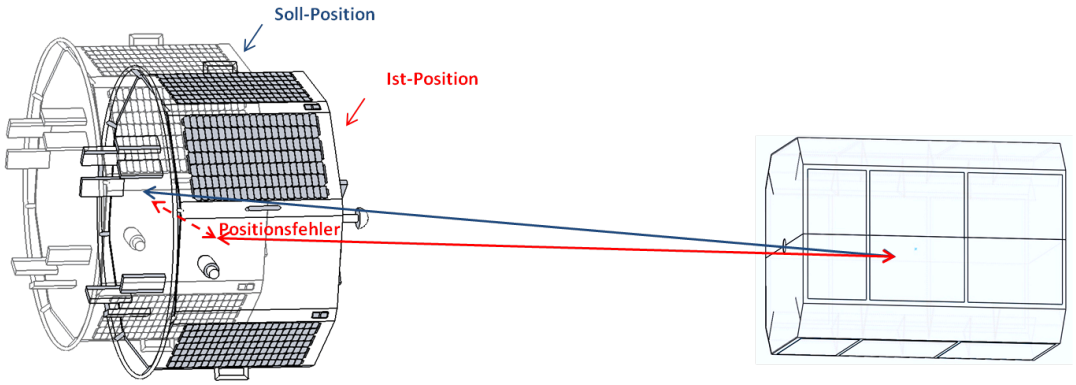


Abbildung 83: *Größenordnung von Fehlerquellen in mm*

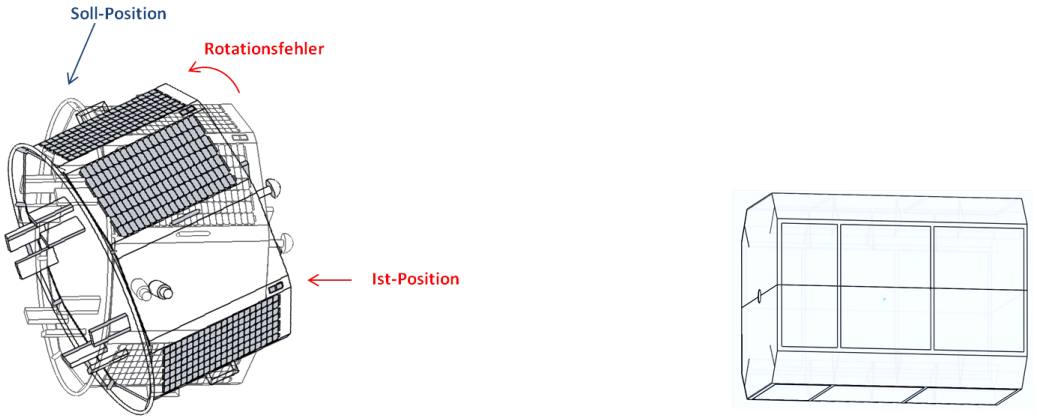
des CableRobots selbst entsteht (Abb.84(b)). Dieser Rotationsfehler erzeugt aber auch einen visuellen Positionsfehler (Abb.84(c)), der aufgrund des großen Relativabstands zwischen Servicer und Client von maximal 8 m einen großen Anteil zur Ungenauigkeit der Anlage beiträgt. Ein Fehler in der Rotationsmessung von 0,15° bei dieser Entfernung resultiert nach Gl(37) bereits in einem visuellen Positionsfehler von 2,1 cm.

$$e_{Vis} = \sqrt{e1^2 + e2^2} = \sqrt{(d \cdot sin(\alpha))^2 + (d - d \cdot cos(\alpha))^2} = d \cdot \sqrt{2 \cdot (1 - cos(\alpha))} \quad (37)$$

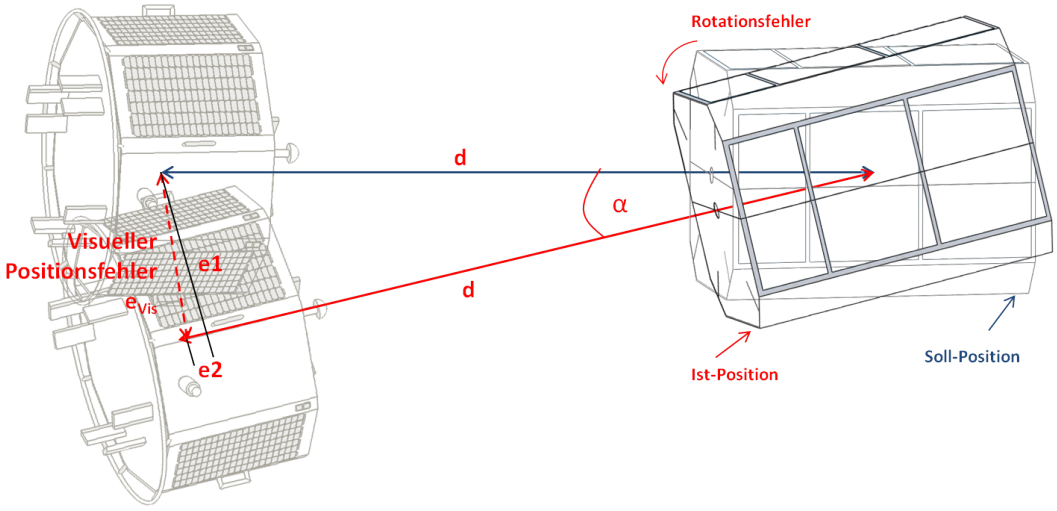
Minimale Verbesserungen der Anlagengenauigkeit könnten durch Tunen verschiedenster Parameter (Filter, Regler, ...) erreicht werden. Eine Verbesserung der Genauigkeit in der Größenordnung von einer 10er Potenz erfordert ein besseres Messsystem und genauere Hardwarekomponenten. Der KUKA besitzt den Nachteil einer Ethernet Ansteuerung und relativ geringen Taktzeit. Der CableRobot tendiert eher zu Schwingungen. Das MTS könnte mit mehreren Kameras erweitert werden. Dadurch würde zum einen die Abdeckung im Arbeitsbereich und damit die Genauigkeit erhöht werden. Zum anderen könnte auch der Arbeitsbereich vergrößert werden, so dass keine Positionen existieren, die schlecht oder gar nicht abgedeckt werden und somit schlechte bis gar keine Messdaten liefern würden. In Zukunft müssten die Posen von Servicer und Client direkt gemessen werden. Dazu müsste bei der Konstruktion der Mockups darauf geachtet werden, wo und wie Marker eines Positionsmessgerätes angebracht werden können. Gegebenenfalls muss ein Ersatz für das MTS gefunden werden, welches genauer und mit aktiven Markern arbeitet, so dass die LIDAR Messungen nicht beeinflusst werden würden.



(a) Positionsfehler



(b) Rotationsfehler



(c) Visueller Positionsfehler

Abbildung 84: *Mögliche Fehlerarten*

3.2 Die wichtigsten Positionen des zahlenmäßigen Nachweises

Kostenart	Nr.	VK	VN
Material	0813	193.463,00 €	136.677,18 €
FE-Fremdleistungen	0823	400.000,00 €	409.283,84 €
Summe Personalkosten	0837	780.857,88 €	907.369,43 €
Reisekosten	0838	0,00 €	1.517,76 €
Abschreibungen auf vorhabenspezifische Anlagen	0847	0,00€	7.145,39 €
Abschreibungen auf sonstige Anlagen	0848	0,00 €	0,00 €
Sonstige unmittelbare Vorhabenkosten	0850	1.000,00 €	672,38 €
Innerbetriebliche Leistungen	0856	0,00 €	0,00 €
Verwaltungskosten	0860	472.419,02 €	425.675,25 €
gesamte Selbskosten des Vorhabens	0881	1.847.739,90 €	1.888.341,22 €

Tabelle 15: *Die wichtigsten Positionen des zahlenmäßigen Nachweises*

3.3 Nutzen und Verwertungsplan

Grade im Bereich der Raumfahrttechnologie ergeben sich für Weiterentwicklungen der INVERITAS-Simulationsanlage zahlreiche wirtschaftlich interessante Anwendungen.

Nach erforderlichen Verbesserungen der Genauigkeit der Anlage sowie der Tragfähigkeit der Bewegungssysteme für höhere Lasten kann die Anlage für die Hardware-in-the-Loop-Simulation einer Vielzahl von Rendezvous-Manövern eingesetzt werden. Im Bereich Satelliten-Servicing aber auch bei orbitalen Andock-Manövern z.B. von Versorgungsfahrzeugen an Raumstationen ergeben sich Anwendungen, konkret z.B. im Bereich der DEOS-Mission. Doch auch im zunehmend auch wirtschaftlich relevanten Bereich der Space Debris Entsorgung könnten Capture-Vorgänge mit Weiterentwicklungen der INVERITAS-Anlage simuliert werden. Auch Landemanöver z.B. bei einer Mondmission können durch die großen translatorischen Freiheiten des Kabelroboters nach entsprechenden Anpassungen betrachtet werden.

Nach einer Erhöhung der Präzision des Systems sind auch Tests von Sensorsystemen wie z.B. verschiedenen LIDAR-Varianten für Lande- und Dockingmanöver möglich. Dies ist in einer möglichen Beauftragung des DFKI durch Astrium im Rahmen von DEOS Phase B2 geplant.

Auch bei erdgebundenen Anwendungen könnten nach entsprechenden Anpassungen der Anlage autonome Rendezvous oder Docking-Vorgänge betrachtet werden, beispielsweise autonomes Docking von Elektrofahrzeugen zur Bildung von Road-Trains oder Erntegutübernahme zwischen autonomen Erntemaschinen.

Die INVERITAS-Simulationsanlage ist ein in der konkreten Umsetzung einzigartiges Werkzeug zur Hardware-in-the-Loop-Simulation (=HIL-Simulation), das durch seinen auch softwareseitig modularen Aufbau vielseitig einsetzbar und erweiterbar ist.

Im Bereich von Weltraummissionen sind auch interaktive Anwendungen denkbar, wie z.B. ein interaktives manuelles Steuern eines orbitalen Andockmanövers mit Hilfe von echten Sensordaten aus der INVERITAS-Anlage, ggf. auch mit realen Signalverzöge-

rungen über eine echte Satelliten-Datenstrecke. Die Technische Universität München verfügt z.B. über eine solche Datenstrecke und es wird derzeit geplant, diese in Gemeinschaftsprojekten zusammen mit der INVERITAS-Anlage einzusetzen.

In den Projekten RTES-TA/V (beantragt) und RTES-TA (geplant) soll die INVERITAS-Anlage weiterentwickelt werden, um sie auch im Bereich Space-Debris-Removal einsetzen zu können bzw. sie auch in diesem Rahmen einzusetzen. Dabei werden in RTES-TA/V die möglichen Weiterentwicklungen hauptsächlich geplant und teilweise auch umgesetzt, erst in RTES-TA sollen weitreichendere Systemänderungen vorgenommen und das so verbesserte Simulationssystem auch im Projekt eingesetzt werden, zur Verifikation der im Projekt neu entwickelten Space-Debris-Removal-Technologien.

Jegliche Entwicklung autonomer Hardware sollte in einer HIL-Simulation getestet werden, bevor direkt nach Softwaresimulationen gleich ein Einsatz unter Realbedingungen folgt, der das System und seine Umgebung gefährden könnte. Jegliche Forschung die autonome Docking- oder Captureverfahren zum Ziel hat kann von Weiterentwicklungen des INVERITAS-Simulationssystems profitieren. Dies kann die wissenschaftlichen Forschungen z.B. auch in den Bereichen autonomes Docking von AUVs, autonomes Bilden von Road-Trains aus Fahrzeugen oder autonome Teams von Erntemaschinen unterstützen.

In Folgeprojekten wie z.B. dem geplanten Projekt "RTES: Robotische Technologien zur Entsorgung von Weltraumschrott" kann die Präzision der Anlage durch den Einsatz verbesserter Tracking-Methoden wie z.B. einem Indoor-GPS beträchtlich erhöht werden, was die Einsetzbarkeit der Anlage z.B. für Sensorevaluationen beträchtlich erhöht. Eine weitere Erhöhung des möglichen Einsatzsprektrums der Anlage kann durch die Erhöhung der möglichen Nutzlasten des Bewegungssimulationssystems erreicht werden. In dem Beantragung befindlichen Vorhaben "RTES-TA/V: Robotische Technologien zur Entsorgung von Weltraumschrott - Phase 1: Technologie Assessment - Verifikation" sollen solche Weiterenwicklungen geplant werden, um diese dann in Folgeprojekten wie "RTES" umzusetzen.

Das oben genannte präzise Tracking von Bewegungen ist dabei eine Technologie, die wissenschaftlich in vielen weiteren Anwendungsgebieten nutzbar ist. Beispielsweise ist eine präzise Erfassung von Roboterbewegungen erforderlich, um deren Verhalten mit Simulationen abzugleichen, was für die Entwicklung und Verifikation von Simulationswerkzeugen sehr wichtig ist. Auch Interaktionen des Menschen mit einem Simulations- oder Robotersystem können durch eine präzise Erfassung der Nutzerbewegungen weiterentwickelt werden. Auch bei der Steuerung von Robotersystemen in künstlichen kooperativen Umgebungen wie Fabrikhallen, Raumstationen oder Laboratorien können präzise Tracking-Systeme eingesetzt werden.

Im Folgenden werden tabellarisch Verwertungsmöglichkeiten des Projekts “INVERITAS - Innovative Technologien zur Relativnavigation (-bewegung) und Capture mobiler autonomer Systeme” dargestellt.

lfd.	Bezeichnung	Zeithorizont
1	DEOS Phase B2, mögliche Beauftragung durch Astrium, um Sensorik mit der INVERITAS-HIL-Simulationsanlage zu verifizieren	kurzfristig
2	RTES-TA/V, beantragtes Projekt mit Astrium als Projektpartner und Konsortialführer, um unter Anderem mögliche Weiterentwicklungen der INVERITAS-HIL-Simulationsanlage zu analysieren, zu planen und teilweise umzusetzen	kurzfristig
3	RTES-TA, geplantes Projekt mit Astrium als Projektpartner und Konsortialführer, um Möglichkeiten zur Space-Debris Beseitigung zu entwickeln, was entsprechende Weiterentwicklungen an der INVERITAS-HIL-Simulationsanlage beinhaltet	mittelfristig
4	Einsatz der INVERITAS-Anlage zur Systemverifikation in Projekten, die autonome Docking-Vorgänge zum Thema haben	langfristig

3.4 Forschungsergebnisse Dritter

Es sind in Deutschland mehrere Bewegungssimulationssysteme (z.B. INVERITAS, Lama, EPOS, TRON) verfügbar, die aber alle für unterschiedliche Anforderungen ausgelegt sind. Wie im Projekt INVERITAS gezeigt wurde, eignet sich die INVERITAS-Anlage besonders für Entwicklungsaktivitäten für Satellite Servicing Missionen, EPOS hat dagegen durch die wesentlich größeren Genauigkeiten Vorteile bei Verifikationsaufgaben.

Zum Vergleich folgen zunächst die Daten des Bewegungssimulationssystems aus dem Projekt **INVERITAS**:

Ort: Weltraumexplorationshalle DFKI GmbH, Robotic Innovation Center Bremen

Manipulatoren:

- KR60 (Kuka, matt schwarz)
- Kabelgeführter Roboter (CableRobot) mit zusätzlicher Achse in Z-Richtung (SpiderCam, 4 Motorwinden mit jeweils zwei Kabeln ausgerichtet als Parallelogramm, Traglast 150kg)

Mock-ups:

- Client-Mock-up am KR (optische Nachbildung eines Satelliten im Maßstab 1:1 ohne Funktion)
- Servicer Mock-up am Cable-Robot (optische Nachbildung eines Satelliten im Maßstab 1:4 inkl. Rechnerarchitektur zur Steuerung und Bildverarbeitung, einem Lidar und mehreren Kameras)

Lichtverhältnisse:

- Matt schwarzer Anstrich des gesamten Explorationsraumes, inkl. der Manipulatoren und Geräte
- Sechs 600W Warp-Scheinwerfer mit einstellbarer Schärfe und Fokus (verfahrbar in vertikaler Richtung mittels Zügen)

Simulationseigenschaften:

- Simulation zweier Objekte im Abstand von 0 - 18m
- Rotationsfreiheit: Client >360 Grad, Rotationsfreiheit Servicer abhängig von der Entfernung der Objekte ca. 10 - 360 Grad
- Nur leichte Kontaktsimulation möglich (Taktfrequenz 83,3Hz (Kuka) und 250Hz (CableRobot), schwingungsanfällig)

Steuerung:

- Echtzeitfähig mittels dSpace
- Modellierung u. a. mittels Matlab, Simulink, C

Ortung:

- Vicon Motion-Tracking-System (6 Kameras mit jeweils 1M Pixel und 1 Kamera mit jeweils 16M Pixel, decken ca. 80 Prozent des Arbeitsraumes ab)
- Absolutgenauigkeit 5,3 mm
- zusätzlich: Positionsbestimmung über Gelenkstellung (Kuka) und Kabellängen (CableRobot)
- Offline Verifikation mittels Laser-Tracker

Es gibt aktuell verschiedene Bewegungssimulationssysteme, die mit der INVERITAS-Anlage vergleichbar sind, von denen auf einige repräsentative Beispiele im Folgenden eingegangen wird:

EPOS 2:

Ort: Deutsches Raumfahrtkontrollzentrum GSOC, DLR e.V. Oberpfaffenhofen

Manipulatoren:

- KR100 (Kuka, orange, verfahrbar auf einer Linearschiene montiert)
- KR240 (Kuka, orange)

Mockups:

- Jeweils eine Satelliten-Nachbildung mit Solarpanele ohne Funktion

Lichtverhältnisse:

- Standardverhältnisse einer Fabrikhalle, kein Reflexionsschutz
- bodengebundener manuell verfahrbarer Scheinwerfer für eine direkte Beleuchtung

Simulationseigenschaften:

- Simulation zweier Objekte im Abstand von 0 - 22m
- Rotationsfreiheit pro Objekt und Achse >360 Grad
- Kontaktsimulation möglich (Taktfrequenz 250Hz, kaum schwingungsanfällig)

Steuerung:

- Echtzeitebene mittels vxWorks
- Modellierung u. a. mittels Matlab, Simulink, C

Ortung:

- Positionsbestimmung über Gelenkstellung (Kuka) und Linearmotor
- Offline Verifikation mittels Leica-Laser-Tracker

LAMA:
Ort: Institut für Raumfahrtsysteme RY, DLR e.V. Bremen
Manipulatoren:

- KR500 (Kuka, orange, verfahrbar auf einer Linearschiene montiert)

Lichtverhältnisse:

- Standartverhältnisse einer Fabrikhalle, kein Reflexionsschutz

Simulationseigenschaften:

- Simulation von Aufsetz- und Fahrdynamik unter reduzierter Schwerkraft

TRON:
Ort: Institut für Raumfahrtsysteme RY, DLR e.V. Bremen
Manipulatoren:

- KR16 (Kuka, schwarz, verfahrbar auf einer Linearschiene montiert)

Lichtverhältnisse:

- Matt schwarze Verkleidung des gesamten Explorationsraumes, inkl. der Manipulatoren und Geräte
- Auf einer Krananlage befestigter Scheinwerfer, 4 Freiheitsgrade

Simulationseigenschaften:

- realistische Anflugssimulation auf ein natürliches Objekt (z. B. Mond)
- skalierte Oberfächenstrukturen

Ortung:

- Positionsbestimmung über Gelenkstellung (Kuka) und Linearmotor

Fazit:

INVERITAS und Epos sind zwei robotische Systeme, welche dazu konstruiert wurden, relative Bewegungsmuster zweier Objekte zu simulieren und zwar unabhängig von ihrer eigentlichen Bewegungsmechanik. Solche Simulationen dienen in erster Linie der Verifikation von Sensoren und Steuerungsalgorithmen.

Das INVERITAS-System basiert auf zwei sehr unterschiedlichen Manipulatoren, welche ein völlig gegensätzliches Verhalten aufweisen. Der Kuka besitzt geringe Freiheiten in der Translatorik und große Freiheiten in der Rotatorik. Der CableRobot besitzt hingegen große translatorische Freiheiten, dafür nur eine rotatorische, nämlich die in seiner Vertikalen. In der Ansteuerung der Manipulatoren müssen diese unterschiedlichen Bewegungsmuster beachtet werden, um eine relative Positions- und Lagebestimmung vorzunehmen. Nachteilig an diesem System sind die sehr eingeschränkten Rotationsmöglichkeiten des Servicers bei großem Abstand beider Objekte. Würde man den Kabelroboter noch zwei Rotationsfreiheitsgrade dazugeben, könnte man allerdings den gesamten Raum der Halle als redundanten Freiheitsgrad verwenden, um z. B. einen realistischen Beleuchtungszustand zu erhalten, oder ein anderes drittes Objekt in die Simulation mit einzubeziehen.

Epos ging den direkten Weg der Simulation über eine Linearschiene. Beide Roboter besitzen hohe Rotationsfreiheiten und kaum translatorische Freiheiten. Letztere sind für diese Art der Simulation aber nicht nötig, da eine direkte Nachahmung der relativen Position und Lage unter Nutzung der eindimensionalen translatorischen Freiheit der Linearschiene auf der Verbindungsgeraden zwischen beiden Objekten möglich ist.

Neben den Manipulatoren spielen besonders auch die Umgebungsbedingungen der Simulationsanlage eine wichtige Rolle, vor allem für eine realistische Verifikation der Sensoren. Da sich die Simulationen allesamt im Weltraum abspielen, wurde die gesamte INVERITAS-Anlage mit einer matt-schwarzen lichtabsorbierenden Farbe versehen, welche fast keine Lichtreflexionen zulässt. Desweiteren wurden vertikal verfahrbare Lichtstrahler an geeigneten Punkten der Anlage angebracht, welche ein Lichtspektrum aufweisen, dass dem der Sonne sehr ähnelt. Das Ergebnis ist eine dunkle

Simulationsumgebung, in welcher die Simulationsobjekte realistisch scharfe Schattenkanten werfen.

Ein weiteres wichtiges Merkmal stellt die Genauigkeit der Anlage bezüglich der relativen Positionierbarkeit von Servicer und Client dar. INVERITAS verwendet diesbezüglich ein Motion-Tracking-System und die transformierte Lagebestimmung der Manipulatortelemetriedaten. Die resultierende Genauigkeit des Systems wurde mit einem Laser-Tracker verifiziert und als ausreichend befunden. Eine Positionsbestimmung mittels der Kinematik scheitert vor allem am Kabelroboter, dessen Kabelverhalten nur unzureichend physikalisch beschrieben ist. Die Vicon-Kameras besitzen hingegen für die räumlichen Ausmaße eine relativ geringe Auflösung. Eine Erhöhung der Genauigkeit besitzt für die Zukunft somit hohe Priorität.

Epos besitzt Vermessungssysteme basierend auf der Roboterkinematik und einer Leica-Offline-Vermessung. Durch die sehr präzise gefertigten Bauteile und die hohe Steifigkeit der Materialien lassen sind die Ergebnisse durchaus belastbar. Eine zusätzliche Laser-Online-Vermessung der beiden Simulationsobjekte ist bereits geplant und wird die Genauigkeit weiter stark erhöhen.

Im Gegensatz zu INVERITAS und Epos wurde die Anlage LAMA zur Simulation von Aufsetz- und Fahrdynamik entwickelt. Ziel ist die Untersuchung der Roboterdynamik unter reduzierter Schwerkraft mittels Kontaktdynamik. Eine Simulation zweier Objekte ist nicht möglich.

Die Roboteranlage TRON dient zur Untersuchung des Anflugverhaltens eines künstlichen Objektes auf ein natürliches Objekt (z. B. Anflug auf den Mond). In dieser Anlage wird eine skalierte Simulation durchgeführt (1:50000 bis 1:100). Eine unskalierte Simulation ist ebenso wie eine Simulation zwischen zwei künstlichen Objekten nicht vorgesehen bzw. nicht möglich.

3.5 Veröffentlichungen

Da die wichtigste Innovation der DFKI-Arbeiten im Projekt INVERITAS das Zusammenwirken aller Einzelkomponenten im Langstrecken-Bewegungssimulationssystem ist, das dadurch zu einem leistungsfähigen HIL-Simulator für allgemeine Rendezvous und Docking Vorgänge wird, ist eine zusammenfassende Veröffentlichung über das Gesamtsystem geplant. Zuvor mussten insbesondere die abschließenden Tests zur Präzision der Anlage durchgeführt werden, um Teil einer entsprechenden Veröffentlichung werden zu können. Da diese nun vorliegen, wird aktuell an einer entsprechenden Veröffentlichung gearbeitet. Die Form der Veröffentlichung wird noch entschieden.

Im Projektverlauf wurde die INVERITAS-Anlage mehrfach Besuchern vorgeführt und wurde auch von den öffentlichen Medien positiv aufgenommen und war Gegenstand mehrerer Reportagen. Im Folgenden werden die wichtigsten Reportagen und Besuchsveranstaltungen in den Tabellen 16 und 17 aufgeführt:

Ferner wurde das Verbundvorhaben INVERITAS am 06.03.2012 auf der zweiten nationalen Konferenz zur RaumfahrtRobotik in Berlin durch den Konsaortialführer Astrium präsentiert.

Datum	Titel	Medium
07.05.2012	Ein Hauch von Raumfahrt	Weser Kurier
26.10.2011	Vom Weltraum-Schrott zum Weltraum-Werkstoff	Handelsblatt
26.10.2011	Darpa will Satelliten wiederverwerten	Golem.de
12.09.2011	In Deutschland wurde ein Testgelände erschaffen, das die Mondlandschaft nachbildet (Originaltext in Russisch)	Cybersecurity.ru
26.05.2011	Pioniere für fremde Planeten	Technology Review
26.03.2011	Künstliche Intelligenz auf dem Vormarsch	ORF
19.03.2011	Wissenschaftliches Spiel zeigt große Wirkung	morgenweb.de
10.03.2011	"Keine Mittel vom Militär"	TAZ
08.03.2011	Liegt Bremen hinter dem Mond?	SWR2
27.12.2010	Spielwiese für Roboter	Deutschlandradio
26.11.2010	Training vor dem Einsatz	Deutschlandradio
23.11.2010	Der Mond liegt in Bremen	Golem.de
23.11.2010	Neues Testgelände für Weltraum-Roboter	astronews.com
23.11.2010	Krabbelstube für Weltraum-Roboter	Kurier.at
23.11.2010	Universum aus der Retorte	Nordsee-Zeitung.de
22.11.2010	Härtetest am künstlichen Mondkrater	Handelsblatt.com
22.11.2010	Roboter üben auf künstlichem Mondkrater (dpa-Bericht)	ZEIT online, Focus online, Frankfurter Rundschau, sueddeutsche.de, Welt online, Handelsblatt.com, Berliner Zeitung, stern.de, Wirtschaftswoche.de, br-online, n24.de, heise.dernz.de, wetterauer-zeitung.de, augsburger-allgemeine.de, rhein-zeitung.de, suedkurier.de, maerkischeallgemeine.de, general-anzeiger-bonn.de, echo-online.de, usinger-anzeiger.de, mainpost.de, borkenerzeitung.de, Elbe Jeetzel-Zeitung.de, lz-online.de
22.11.2010	Roboter testen Einsatz im All in Bremen	BILD.de
22.11.2010	Roboter testen in Bremen den Einsatz im Weltall (dapd-Meldung)	Nordwest Zeitung online, nordkurier.de, ovb-online.de, pr-inside.com
22.11.2010	Künstliche Mondlandschaft für Roboter in Bremen	radiobremen.de
22.11.2010	Trainingsgelände für Roboter in Bremen eröffnet	Weser Kurier

Tabelle 16: *Pressespiegel Projekt INVERITAS, 05/2009 - 05/2012*

Datum	Veranstaltung
22.11.2010	Eröffnung der Weltraum-Explorationshalle
29.06.2011	Tag der offenen Tür
05.07.2012	Tag der offenen Tür

Tabelle 17: *Vorführungen des Projekts INVERITAS für Besucher, 05/2009 - 05/2012*

Literaturverzeichnis

[1] BEER, R. D. ; GALLAGHER, J. C.: Evolving Dynamical Neural Networks for Adaptive Behavior. In: *Adaptive Behavior* 1 (1992), Juni, Nr. 1, 91–122. http://dx.doi.org/10.1177/105971239200100105. – DOI 10.1177/105971239200100105. – ISSN 1059–7123

[2] BELL, R. ; MORPHOPOULOS, T. ; POLLACK, J. ; COLLINS, J. ; WERTZ, J. ; ALLEN, R.: Hardware-in-the-loop tests of an autonomous gn&c system for on-orbit servicing. In: *AIAA-LA Section/SSTC Responsive Space Conference*, 2003

[3] BOGE, T. ; WIMMER, T. ; MA, O. ; ZEBENAY, M.: EPOS - A Robotics-Based Hardware-in-the-Loop Simulator for Simulating Satellite RvD Operations.

[4] BONGARD, J.C. ; PFEIFER, Rolf: A method for isolating morphological effects on evolved behaviour. In: *appear in Proceedings of the Seventh International Conference on the Simulation of Adaptive Behavior*, Citeseer, 2002

[5] CHEMOVA, S. ; VELOSO, M.: An evolutionary approach to gait learning for four-legged robots. In: *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*. http://dx.doi.org/10.1109/IROS.2004.1389794. – DOI 10.1109/IROS.2004.1389794. ISBN 0–7803–8463–6

[6] DIETRICH, A. ; WIMBOCK, T. ; TAUBIG, H. ; ALBU-SCHAFFER, A. ; HIRZINGER, G.: Extensions to reactive self-collision avoidance for torque and position controlled humanoids. In: *Robotics and Automation (ICRA), 2011 IEEE International Conference on* IEEE, 2011, S. 3455–3462

[7] FESTO: *Adaptiver Greifer*. http://www.festo.com/cms/de_de/15922.htm,

[8] FRITZKE, Bernd: Growing Cell Structures - A Self-organizing Network for Unsupervised and Supervised Learning. In: *Neural Networks* 7 (1993), S. 1441–1460

[9] GRUAU, Frederic ; QUATRAMARAN, Kameel: Cellular encoding for interactive evolutionary robotics. Amsterdam, The Netherlands : CWI (Centre for Mathematics and Computer Science), 1996. – Forschungsbericht

[10] HORNBY, G S. ; FUJITA, M ; TAKAMURA, S ; YAMAMOTO, T ; HANAGATA, O: Autonomous Evolution of Gaits with the Sony Quadruped Robot. In: *Proceedings of the Genetic and Evolutionary Computation Conference* Bd. 2. Orlando, Florida, USA : Morgan Kaufmann, 1999. – ISBN 1–55860–611–4, 1297–1304

[11] HUYNH, D.Q.: Metrics for 3D rotations: Comparison and analysis. In: *Journal of Mathematical Imaging and Vision* 35 (2009), Nr. 2, S. 155–164

[12] ISO, ISO: *9283: 1998, manipulating industrial robots-performance criteria and related test methods*. 1998

[13] KIM, Sangbae ; SPENKO, M. ; TRUJILLO, S. ; HEYNEMAN, B. ; MATTOLI, V. ; CUTKOSKY, M.R.: Whole body adhesion: hierarchical, directional and distributed control of adhesive forces for a climbing robot. In: *Robotics and Automation, 2007 IEEE International Conference on*, 2007. – ISSN 1050–4729, S. 1268 –1273

[14] LEONHARD, W.: *Regelung elektrischer Antriebe*. Springer, 2000

[15] LEY, W. ; WITTMANN, K. ; HALLMANN, W.: *Handbook of space technology*. Bd. 22. Wiley, 2009

[16] PARK, K.T. ; PARK, C.H. ; SHIN, Y.J.: Performance test of industrial dual arm robot. In: *Industrial Informatics, 2008. INDIN 2008. 6th IEEE International Conference on* IEEE, 2008, S. 425–429

[17] RECHENBERG, I.: *Evolutionsstrategie; Optimierung technischer Systeme nach Prinzipien der biologischen Evolution.* [Stuttgart-Bad Cannstatt] Frommann-Holzboog [c1973], 1973. – 170 S.

[18] REIL, Torsten ; HUSBANDS, Phil: Evolution of central pattern generators for bipedal walking in a real-time physics environment. In: *IEEE Trans. Evolutionary Computation* 6 (2002), S. 159–168

[19] REINTSEMA, D. ; SOMMER, B. ; WOLF, T. ; THEATER, J. ; RADTHKE, A. ; SOMMER, J. ; NAUMANN, W. ; RANK, P.: DEOS-THE IN-FLIGHT TECHNOLOGY DEMONSTRATION OF GERMAN'S ROBOTICS APPROACH TO DISPOSE MALFUNCTIONED SATELLITES.

[20] RICHTER, L. ; EPPLER, S. ; GRZESIK, A. ; WEISS, S.: Flexible Wheels for Planetary Rovers? - A Comparative Investigation of Planetary Rover Traction Performance. In: *Joint Asia-Pacific and North American Conference of the International Society for Terrain-Vehicle Systems (ISTVS)*, 2007

[21] SIMS, Karl: Evolving virtual creatures. In: *Proceedings of the 21st annual conference on Computer graphics and interactive techniques - SIGGRAPH '94* (1994), Nr. July, 15–22. http://dx.doi.org/10.1145/192161.192167. – DOI 10.1145/192161.192167. ISBN 0897916670

[22] TAUBIG, H. ; BAUML, B. ; FRESE, U.: Real-time swept volume and distance computation for self collision detection. In: *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on* IEEE, 2011, S. 1585–1592