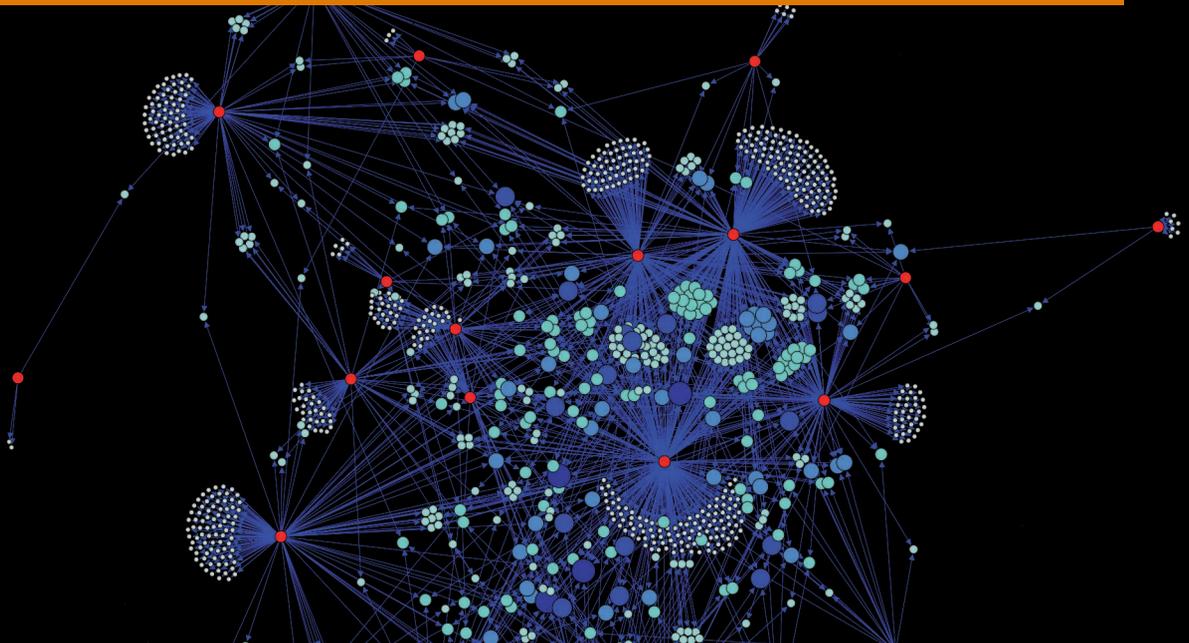


Document D-14-06



Proceedings of the RIC Project Day

Workgroups '*Locomotion & Mobility*' and
'*Simulation*'

Frank Kirchner (Editor)

Florian Cordes, Michael Rohn (Associate Editors)

09/2014

Document D-14-06

German Research Center for Artificial Intelligence (DFKI) GmbH

Bibliographic information published by the German National Library

The German National Library lists this publication in the German National Biography; detailed bibliographic data are available in the internet at <http://dnb.ddb.de>.

© German Research Center for Artificial Intelligence (DFKI) GmbH, 2014

This work may not be copied or reproduced in whole or in part for any commercial purpose. Permission to copy in whole or in part without payment of fee is granted for nonprofit educational and research purposes provided that all such whole or partial copies include the following: a notice that such copying is by permission of the German Research Center for Artificial Intelligence (DFKI) GmbH, Kaiserslautern, Federal Republic of Germany; an acknowledgement of the authors and individual contributors to the work; all applicable portions of this copyright notice. Copying, reproducing, or republishing for any other purpose shall require a licence with payment of fee to German Research Center for Artificial Intelligence (DFKI) GmbH.

Issue D-14-06 (2014)
ISSN 0946-0098

German Research Center for Artificial Intelligence
Deutsches Forschungszentrum für Künstliche Intelligenz
DFKI GmbH

Founded in 1988, DFKI today is one of the largest nonprofit contract research institutes in the field of innovative software technology based on Artificial Intelligence (AI) methods. DFKI is focusing on the complete cycle of innovation – from world-class basic research and technology development through leading-edge demonstrators and prototypes to product functions and commercialization.

Based in Kaiserslautern, Saarbrücken and Bremen, the German Research Center for Artificial Intelligence ranks among the important ‘Centers of Excellence’ worldwide. An important element of DFKI’s mission is to move innovations as quickly as possible from the lab into the marketplace. Only by maintaining research projects at the forefront of science DFKI has the strength to meet its technology transfer goals.

The key directors of DFKI are Prof. Wolfgang Wahlster (CEO) and Dr. Walter Olthoff (CFO). DFKI’s research departments are directed by internationally recognized research scientists:

- Knowledge Management (Prof. A. Dengel)
- Cyber-Physical Systems (Prof. R. Drechsler)
- Robotics Innovation Center (Prof. F. Kirchner)
- Innovative Retail Laboratory (Prof. A. Krüger)
- Institute for Information Systems (Prof. P. Loos)
- Embedded Intelligence (Prof. P. Lukowicz)
- Agents and Simulated Reality (Prof. P. Slusallek)
- Augmented Vision (Prof. D. Stricker)
- Language Technology (Prof. H. Uszkoreit)
- Intelligent User Interfaces (Prof. W. Wahlster)
- Innovative Factory Systems (Prof. D. Zühlke)

In this series, DFKI publishes research reports, technical memos, documents (eg. workshop proceedings), and final project reports. The aim is to make new results, ideas, and software available as quickly as possible.

Prof. Wolfgang Wahlster
Director

Proceedings of the RIC Project Day

Workgroups ‘Locomotion & Mobility’ and
‘Simulation’

Frank Kirchner^(1,2) (Editor)

Florian Cordes⁽¹⁾, Michael Rohn⁽¹⁾ (Associate Editors)

(1) DFKI GmbH, Robotics Innovation Center, Robert-Hooke-Straße 1, 28359 Bremen, Germany

(2) Universität Bremen, Arbeitsgruppe Robotik, Robert-Hooke-Straße 1, 28359 Bremen, Germany

09/2014

Document D-14-06 des
Deutschen Forschungszentrums für Künstliche Intelligenz (DFKI)

Abstract

This document is the current edition of a publication series which records the topics, discussions and efforts of the workgroups at the DFKI Robotics Innovation Center (RIC). Each edition contains presentation slides and posters of a project day which is organized by two workgroups.

Workgroups provide a platform for cross-project communication and knowledge transfer. They are formed by peers dedicated to a specific topic. Each workgroup has one administrator.

In 2008, the workgroups started to present their results and efforts in an open presentation format called brown-bag talk.

From 2009 onwards, these presentation were held at so-called project days.

Since 2014, a project day consists of two main parts: an oral session and a poster session. Both sessions are documented in a proceedings using the DFKI Document format.

Zusammenfassung

Dieses Dokument enthält die aktuelle Ausgabe einer Tagungsbandserie, welche die Themen, Diskussionen und Bemühungen der Arbeitsgruppen am DFKI Robotics Innovation Center (RIC) protokolliert.

Jede Ausgabe enthält Vortragsfolien und Poster eines Projekttag, der von je zwei Arbeitsgruppen gestaltet wird.

Arbeitsgruppen widmen sich einem bestimmten Themengebiet und stellen eine Plattform dar, um über Projekte hinaus zu kommunizieren und Wissen zu transferieren. Jede Arbeitsgruppe wird von einem sogenannten Kümmerer administriert. Im Jahr 2008 begannen die Arbeitsgruppen ihre Ergebnisse und Arbeiten in einem offenen Vortragsformat – dem sogenannten ‘Brown Bag Talk’ – vorzustellen, welches ein Jahr später in die Form von Projekttagen überführt wurde. Seit 2014 besteht ein Projekttag nicht nur aus Vorträgen, sondern beinhaltet zudem Posterpräsentationen. Beide Formate werden seitdem in einem Tagungsband in Form eines ‘DFKI Documents’ festgehalten.

Contents

Abstract	vii
1 Editorial	3
2 ‘Locomotion & Mobility’	5
2.1 LM-T-01: ‘Introduction to Workgroup Locomotion and Mobility’ <i>Florian Cordes</i>	5
2.2 LM-T-02: ‘Coyote II - Observations on Mobility Performance in distinct Terrains’ <i>Roland U. Sonsalla</i>	9
2.3 LM-T-03: ‘SpaceBot Rover - Chassis Suspension and Mobility Tests’ <i>Marc Manz</i>	18
2.4 LM-T-04: ‘SherpaTT Motion Control System’ <i>Ajish Babu, Florian Cordes, Daniel Kuehn</i>	25
2.5 LM-P-01: ‘Europa Explorer: Development of a Hydrodynamic Casing for the Autonomous Underwater Vehicle’ <i>Philipp Kloss</i>	36
2.6 LM-P-02: ‘Introducing Particle Swarm Optimization into a Genetic Algorithm to Evolve Robot Controllers’ <i>Malte Langosz, Kai A. von Szadkowski</i>	38
2.7 LM-P-03: ‘The new Space Exploration Hall’ <i>Daniel Kuehn, Florian Cordes</i>	40
2.8 LM-P-04: ‘Autonomous Path Tracking Steering Controller for Extraterrestrial Terrain Exploration Rover’ <i>Mohammed Ahmed, Roland U. Sonsalla, Frank Kirchner</i>	42
3 ‘Simulation’	44
3.1 SM-T-01: ‘URDF, SMURF and Phobos - Better robot models for the future’ <i>Kai von Szadkowski, Malte Langosz</i>	44
3.2 SM-T-02: ‘Phobos: 3D Robot Modelling made easy’ <i>Kai von Szadkowski, Malte Langosz</i>	54
3.3 SM-T-03: ‘IMPac SOTTTL’ <i>Yong-Ho Yoo</i>	63
3.4 SM-P-01: ‘CAD2SIM’ <i>Jan Paul, Bertold Bongardt</i>	75
3.5 SM-P-02: ‘Procedural content generation’ <i>Jan Paul</i>	77
3.6 SM-P-03: ‘SMURF: A new, easily extendable robot model format based on URDF’ <i>Kai von Szadkowski, Malte Langosz</i>	79
3.7 SM-P-04: ‘Phobos: A Blender Plugin for Creating Robot Simulation Models’ <i>Kai von Szadkowski, Malte Langosz</i>	81

1 Editorial

This is the third edition of a new format to document the efforts of the DFKI-RIC thematic workgroups. Workgroups are formed by peers and provide a means for cross-project communication on a deep content level and facilitate knowledge transfer amongst the peers. In 2008 we first started forming workgroups on specific topics around robotics and AI research. Among them were topics as ‘system design & engineering’, ‘machine learning’, ‘planning & representation’ as well as ‘frameworks & architectures’ and ‘man-machine interaction’.

These workgroups were established with the intention to provide a platform for interested DFKI-RIC personnel for discussing the state of the art, recent achievements, and future developments in the respective fields. Over time the workgroups gathered a collection of material in form of presentations, short papers, and posters which were worthwhile to be presented also to the rest of the institute.

Due to this development, in 2009, we started to have a project day once every quarter. Each project day provided a platform for two of the workgroups to present their material and to discuss it with the colleagues of the institute.

Nowadays, the project day is organized as a one-day workshop with oral presentations, poster sessions, and a free pizza lunch for everybody who attends. Until now, the talks and posters have only been collected on our servers but were not assembled in a citable document.

The format presented with this document is the next evolutionary step and it aims at eliminating this deficit by compiling the material of the workgroups presented during a project day into a single, citable document of unified format. We will see which steps can be taken in the future to enhance the presentation quality of this material.

Frank Kirchner

At this year’s fourth project day (date: September, 18th 2014) the material of the workgroups *Locomotion & Mobility* and *Simulation* was presented.

In the past years the workgroup *Locomotion & Mobility* used to be focussed on the mechanics aspects of the locomotion apparatus of mobile robots. In the year 2013 there was a time of only reduced activities in the workgroup. With the year 2014 the workgroup was re-initialized and now has the following five topics that are regularly discussed:

- (i) State of the Art. What are new robots with innovative locomotion capabilities?
- (ii) System Description and Comparison. How can we describe locomotive capabilities of systems such that a comparability is provided?
- (iii) Test Facilities. What test facilities are needed, how can we make use of existing facilities in order to provide data for system comparison and abilities?
- (iv) Interfaces between Locomotion Control and High Level Control. What information is needed to be exchanged between the locomotion control layer and higher levels such as navigation and planning?
- (v) Tasks for Mobile Robots. What are typical tasks and what are the required locomotive abilities for mobile robots?

‘Traditionally’ the workgroup *Locomotion & Mobility* has a focus on surface robots, especially walking robots. However, underwater robots are an important part at the DFKI institute and provide a completely different range of locomotive capabilities. With the discussions on the Langdon underwater vehicle (see also LM-P-01), the first underwater system showed up in the workgroup *Locomotion & Mobility*.

In this year’s presentations of the workgroup *Locomotion & Mobility* a focus was on two systems with field experience: The Coyote II rover as a scouting rover for larger exploration rovers as well as the Artemis rover that took part in the DLR SpaceBot Cup.

Coyote II is a micro rover that is based on the design of the successful Asguard family. For the cooperation project FASTER, the rear legged wheels were replaced by cylindrical wheels such that more smooth turning motions in soft soil became possible. The talk presented at this year's project day focussed on video footage from the Airbus DS Mars Yard in Stevenage and experiments in DFKI's own test facilities.

The Artemis talk concentrated on the suspension system design and experiments conducted to improve the design for the SpaceBot Cup. The experiments also include different flexible wheel designs. Artemis' wheels are manufactured by water jet cutting from rubber plates.

The workgroup *Simulation* focused in the past year on tools and formats for the MARS simulation developed at DFKI. The aim is to generate a faster and easier workflow to generate robot models for the simulation. There is for example the SMURF file format which is based on the well known URDF format. The talk "SMURF: A new, easily extendable robot model format based on URDF" gives an overview over the format itself and the ongoing work. An other tool to make the workflow and the editing of a scene file for the simulation more convenience is Phobos. Phobos is an extension of the 3D tool blender. It is able to load old mars scene files and save these as SMURF files. The talk "Phobos: 3D Robot Modelling made easy" All the tools introduced on this project day are work in progress. There are other talks and posters dealing with the conversion of a 3D model from a CAD file to a model which is simplified to be used in an real time simulation environment. This tool is called CAD2SIM. Last but not least is a poster about the procedural content generation. This is really useful for large scaled environment. All these tool will be enhanced in the future. The goal is to integrate these tools in on workflow from an CAD model to an scenario in the (MARS) simulation.

We would like to thank the authors of the fourth project day 2014 for their contributions and for the effort to provide their material in a standardized format.

Florian Cordes, Michael Rohn

2 ‘Locomotion & Mobility’

2.1 ‘Introduction to Workgroup Locomotion and Mobility’ (LM-T-01)

Florian Cordes⁽¹⁾

(1) DFKI GmbH, Robotics Innovation Center, Robert-Hooke-Straße 1, 28359 Bremen, Germany

(2) Universität Bremen, Arbeitsgruppe Robotik, Robert-Hooke-Straße 1, 28359 Bremen, Germany

Contact: `florian.cordes@dfki.de`

Abstract

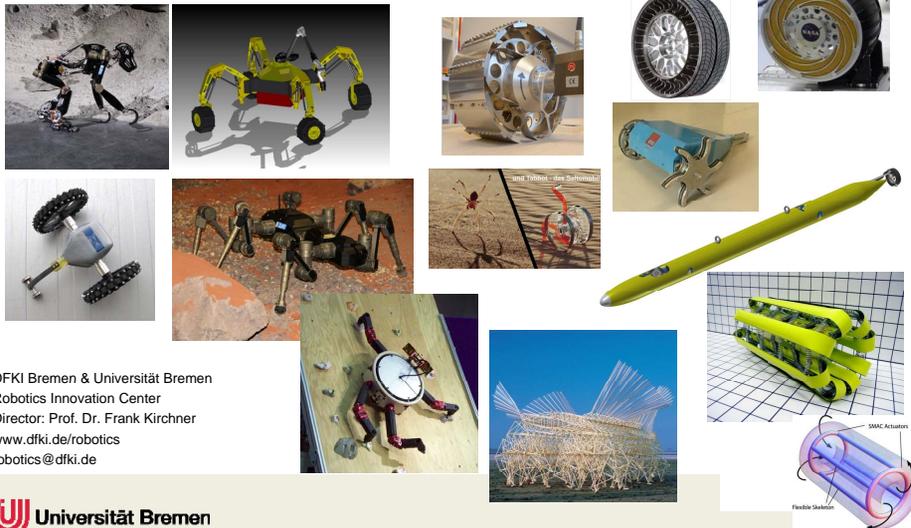
The workgroup Locomotion and Mobility used to be focussed on the mechanics aspects of the locomotion apparatus of mobile robots. In the year 2013 there was a time of only reduced activities in the workgroup. With the year 2014 the workgroup was re-initialized and now has the following five topics that are regularly discussed:

- State of the Art. What are new robots with innovative locomotion capabilities?
- System Description and Comparison. How can we describe locomotive capabilities of systems such that a comparability is provided?
- Test Facilities. What test facilities are needed, how can we make use of existing facilities in order to provide data for system comparison and abilities?
- Interfaces Locomotion Control and High Level Control. What information is needed to be exchanged between the locomotion control layer and higher levels such as navigation and planning?
- Tasks for Mobile Robots. What are typical tasks and what are the required locomotive abilities for mobile robots?

Traditionally the workgroup has a focus on surface robots, especially walking robots. However, underwater robots are an important part at the DFKI institute and provide a completely different range of locomotive capabilities. With the discussions on the Langdon underwater vehicle (see also LM-P-01), the first underwater system showed up in the workgroup Locomotion & Mobility. The discussions concerning the test facilities are currently centred around the new space exploration hall which is currently in the planning phase. However, facilities like the outdoor test track and the current space hall are taken into account as well.



Workgroup Locomotion & Mobility



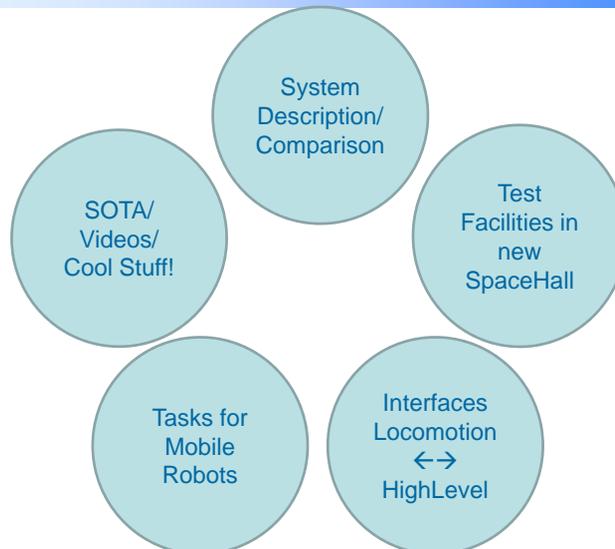
(Regular) Participants



„Applications are Welcome!“



Topics in the Workgroup (since June)



Agenda for Today



- 09:35 – 09:55
Coyote II - Observations on Mobility Performance in distinct Terrains
Roland Sonsalla
- 09:55 – 10:15
SpaceBot Rover: Chassis Suspension and Mobility Tests
Marc Manz
- 10:15 – 10:30
SherpaTT Motion Control System
Ajish Babu

Postersession



Hull Development in Eurex

Philipp Kloss

Introducing Particle Swarm Optimization Into a Genetic Algorithm to Evolve Robot Controllers

Malte Langosz

Autonomous Path Tracking Steering Controller for Extraterrestrial Terrain Exploration Rover

Roland Sonsalla

The new Space Exploration Hall

Florian Cordes, Daniel Kuehn

2.2 ‘Coyote II - Observations on Mobility Performance in distinct Terrains’ (LM-T-02)

Roland U. Sonsalla⁽¹⁾

(1) DFKI GmbH, Robotics Innovation Center, Robert-Hooke-Straße 1, 28359 Bremen, Germany

Contact: roland.sonsalla@dfki.de

Abstract

Coyote II is a micro rover with high mobility performance in various terrains. Equipped with its own power source, on-board sensor suite and computer it is able to perform autonomous exploration tasks. The communication subsystem allows to cooperate with other systems and provides a link for remote control. Due to the robust structural design and powerful actuators, Coyote II is able to carry several kilograms ($\geq 6\text{ kg}$) of payload.

A special characteristic of Coyote II is its novel locomotion concept. It combines the high mobility performance of hybrid legged-wheels (in the front) with the smooth wheel movement of spherical helical wheels (in the rear). Therefore, the scout rover is able to move on soft soil as well as on unstructured terrain and can perform side-to-side steering movements.

The focus of this talk is on observations made during different test drives performed at the DFKI Space Exploration Hall in Bremen and the Airbus DS Mars Yard in Stevenage. During these tests the Space Exploration Hall was equipped with basalt split ($\sim 0 - 1\text{ mm}$) while the Mars Yard was covered with 2EW ($\sim 500, \mu\text{m}$) quartz-based sand. Major differences in the driving performance of the rover could be observed in particular with respect to side-to-side steering movements. The rovers kinematics are introduced, providing the theoretical background for different rover manoeuvres. A comparison of the side-to-side steering movements with different parameter sets is presented by video footage. Furthermore, the results of odometry calibration for the distinct soil within the Mars Yard are presented as well as a several meter long autonomous traverse performed by Coyote II.

FASTER

Forward Acquisition of Soil and Terrain data for Exploration Rover

Coyote II – Observations on Mobility Performance in distinct Terrains

Roland Sonsalla
DFKI RIC
2014, September 18





Deutsches
Forschungszentrum
für Künstliche
Intelligenz GmbH



UNIVERSITY OF
SURREY



AIRBUS
DEFENCE & SPACE



spaceapplications



LIQUIFER
SYSTEMS GROUP

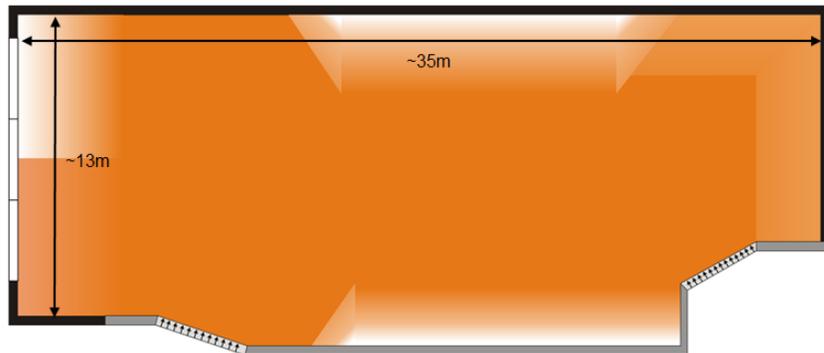


ASTRI POLSKA



Mars Yard @ Airbus DS Stevenage

FASTER
Forward Acquisition of Soil and Terrain data for Exploration Rover



*Figure 4-2: Graphic representation of the ASU Mars Yard.
Tall walls are represented in black; low walls are represented in grey; control room windows are on the left-hand side of the drawing; and entryways are marked by arrows. Sloped sections of the yard are shown by gradient fill areas where the slope rises toward the lighter fill colour.*

The floor of the yard is filled with 2EW soil simulant, which is a coarse (~500µm) quartz-based sand. The simulant is evenly distributed about the yard with a nominal density of about 1.80g/cc (roughly 65% of maximum density for the soil).



Deutsches
Forschungszentrum
für Künstliche
Intelligenz GmbH



UNIVERSITY OF
SURREY



AIRBUS
DEFENCE & SPACE



spaceapplications



LIQUIFER
SYSTEMS GROUP



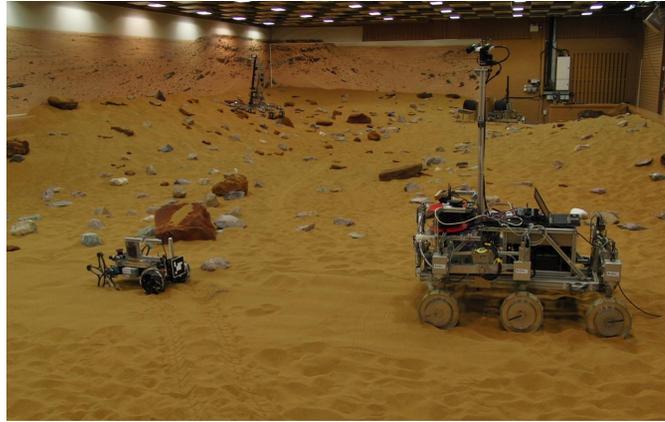
ASTRI POLSKA

2



Mars Yard Impressions

FASTER
Forward Acquisition of Soil and Terrain data for Exploration Rover



3



Kinematics

FASTER
Forward Acquisition of Soil and Terrain data for Exploration Rover

- Wheel circumferences

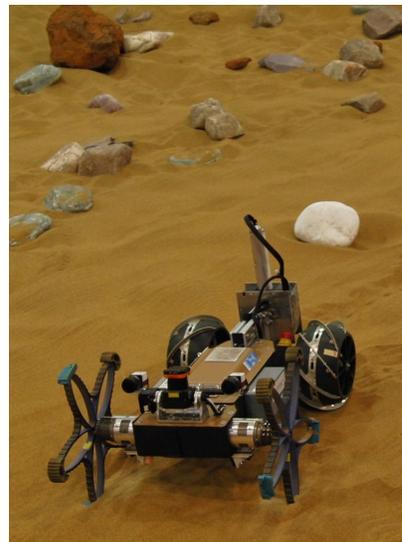
$$D_f = 5 \frac{10 \cdot r_f}{\sqrt{50 + 10\sqrt{5}}} \quad (1)$$

$$D_r = 2\pi r_r \quad (2)$$

- Speed ratio driving straight

$$F = \frac{D_f}{D_r} = \frac{\omega_r}{\omega_f} \approx 1.5 \quad (3)$$

with $r_f = 185$ mm and $r_r = 117.5$ mm.



4



Driving Straight @ Mars Yard

FASTER
Forward Acquisition of Soil and Terrain data for Exploration Rover



- Additional Payload: ~ 5kg



Side-to-Side Steering

FASTER
Forward Acquisition of Soil and Terrain data for Exploration Rover

- Rear wheel helix lead per rotation

$$h_r = \frac{2\pi r_r}{\tan \alpha_g} \quad (4)$$

- Front and rear wheel turn rates

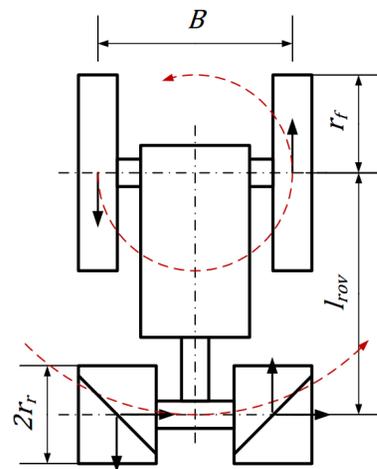
$$\omega_{f,sts} = \frac{\pi B}{D_f} \quad (5)$$

$$\omega_{r,sts} = \frac{2\pi l_{rov}}{h_r} \quad (6)$$

- Speed ratio point turn

$$F_{sts} = \frac{\omega_{r,sts}}{\omega_{f,sts}} \approx 3.1 \quad (7)$$

with $\alpha_g = 45^\circ$, $B = 489$ mm, $l_{rov} = 522$ mm.



Side-To-Side Steering @ DFKI

FASTER
Forward Acquisition of Soil and Terrain data for Exploration Rover

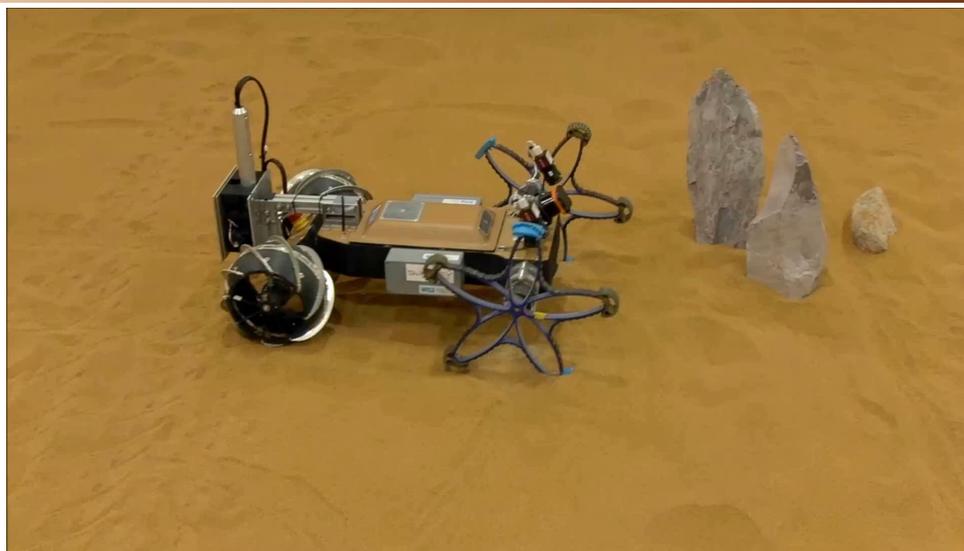


- Wheel Speed Ratio: $F_{sts} = 3.1$



Side-To-Side Steering @ Mars Yard

FASTER
Forward Acquisition of Soil and Terrain data for Exploration Rover



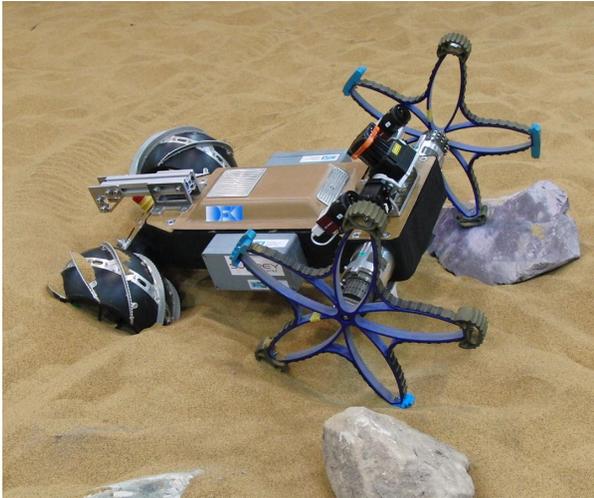
- Wheel Speed Ratio: $F_{sts} = 3.1$



Steering Maneuver

FASTER
Forward Acquisition of Soil and Terrain data for Exploration Rover

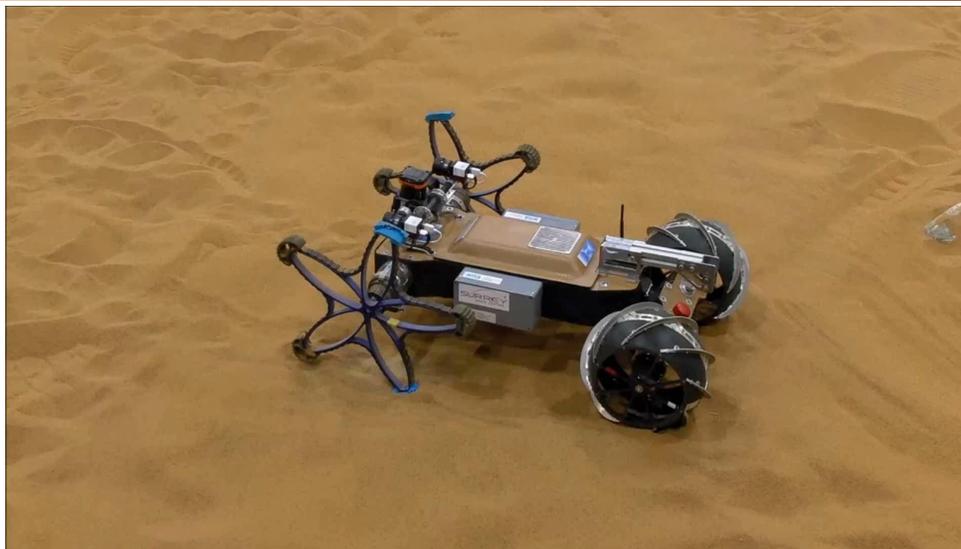
- $F_{sts} = 3.1$



9

Side-To-Side Steering @ Mars Yard

FASTER
Forward Acquisition of Soil and Terrain data for Exploration Rover

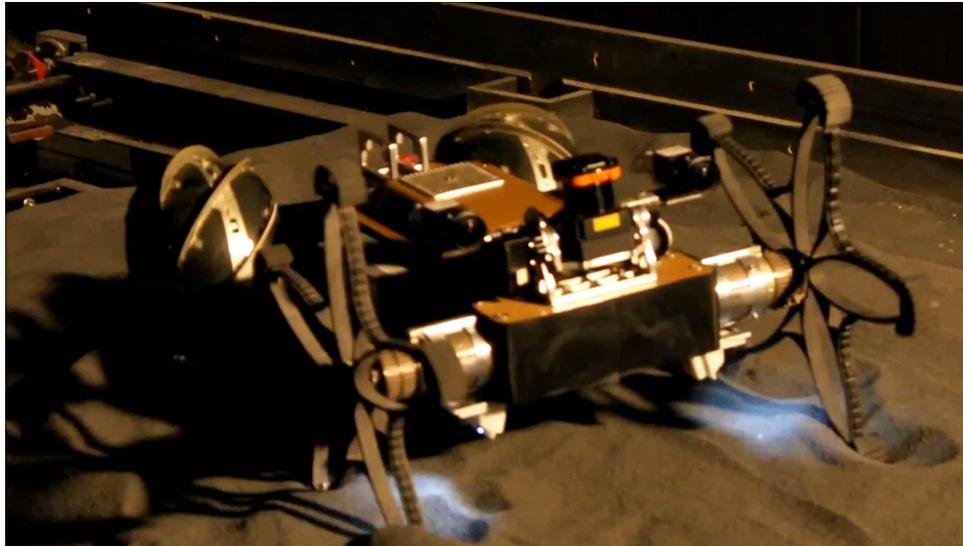


- Wheel Speed Ratio: $F_{sts} = 3.1 * 2$

10

Skid Steering @ DFKI

FASTER
Forward Acquisition of Soil and Terrain data for Exploration Rover



- Wheel Speed Ratio: $F_{sts} = 1.5$



Skid Steering @ Mars Yard

FASTER
Forward Acquisition of Soil and Terrain data for Exploration Rover

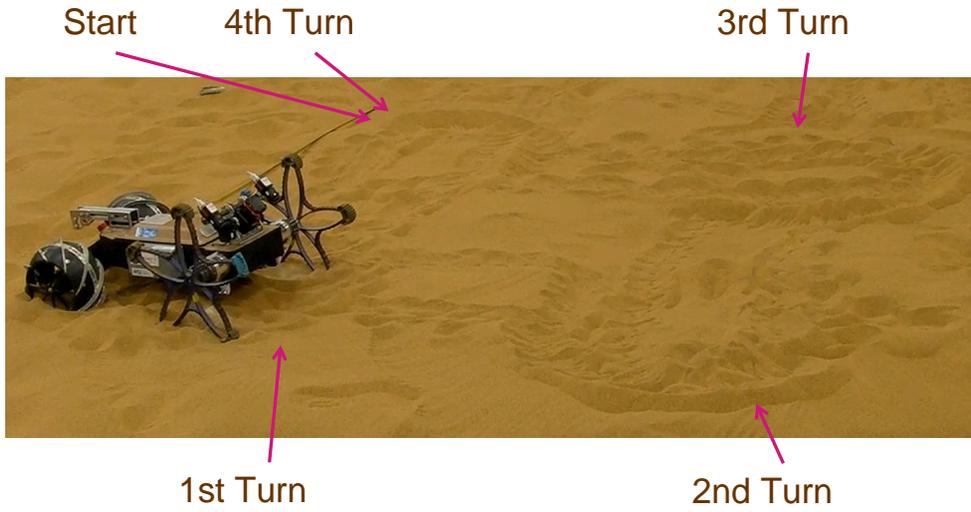


- Wheel Speed Ratio: $F_{sts} = 1.5$



Odometry Calibration

FASTER
Forward Acquisition of Soil and Terrain data for Exploration Rover



Logos for DFK (Deutsches Forschungszentrum für Künstliche Intelligenz GmbH), UNIVERSITY OF SURREY, AIRBUS DEFENCE & SPACE, spaceapplications, LIQUIFER SYSTEMS GROUP, and BCTI POLSKA. The number 13 is displayed on the right, along with the European Union flag and the Horizon 7 logo.

Autonomous Traverse @ Mars Yard

FASTER
Forward Acquisition of Soil and Terrain data for Exploration Rover



Logos for DFK (Deutsches Forschungszentrum für Künstliche Intelligenz GmbH), UNIVERSITY OF SURREY, AIRBUS DEFENCE & SPACE, spaceapplications, LIQUIFER SYSTEMS GROUP, and BCTI POLSKA. The number 14 is displayed on the right, along with the European Union flag and the Horizon 7 logo.



FASTER
Forward Acquisition of Soil and Terrain data for Exploration Rover

Thank you!

<https://www.faster-fp7-space.eu/>

Contact
Thomas Vögele (Project Manager),
Roland Sonsalla (Systems Engineer)
German Research Center for Artificial Intelligence
email: thomas.voegele@dfki.de
roland.sonsalla@dfki.de

 Deutsches Forschungszentrum für Künstliche Intelligenz GmbH
 UNIVERSITY OF SURREY
 AIRBUS DEFENCE & SPACE
 spaceapplications
 LIQUIFER SYSTEMS GROUP
 BSCFI POLSKA

2.3 'SpaceBot Rover - Chassis Suspension and Mobility Tests' (LM-T-03)

Marc Manz⁽¹⁾

(1) DFKI GmbH, Robotics Innovation Center, Robert-Hooke-Straße 1, 28359 Bremen, Germany

Contact: marc.manz@dfki.de

Abstract

This presentation describes the development and test of the mobile manipulation platform Artemis intended for a terrestrial robotic competition. While current space missions are planned to minimize complex manipulation tasks, plans for future space missions go beyond these restrictions. Infrastructure deployment, human-robot cooperative missions and complex sample collection require increasingly complex manipulation capabilities. To meet this need the Spacebot Cup consists of several complex manipulation tasks in unstructured terrain. These requirements were the main design driver for the presented system. The presented rover consists of a 3-Boogie-Chassis designed to increase the maximum stepping size, flexible rubber wheels to increase the maximal climbing inclination on loose surfaces and a small six degree of freedom manipulator to handle objects within the competition. The iterative simulation and experiment process used to develop the flexible rubber wheels is presented. Furthermore experiments are presented which allow a performance comparison between flexible and rigid wheels on loose surfaces.



SpaceBot Rover Chassis Suspension and Mobility Tests

DFKI Bremen & Universität Bremen
Robotics Innovation Center
Director: Prof. Dr. Frank Kirchner
www.dfki.de/robotics
robotics@dfki.de



Agenda



1. Introduction
2. Mechanical Design
 1. Concept
 2. Flexible Wheels
3. Experiments
 1. Step Climbing
 2. Wheel Test
4. Outlook

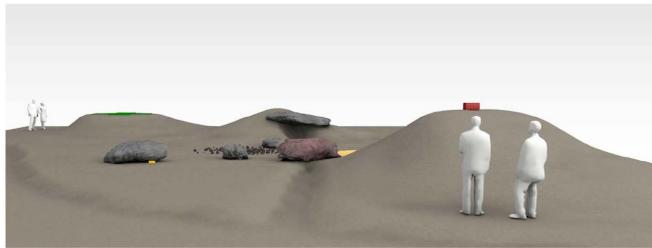


Introduction



SpaceBot Cup:

- Mobile manipulation
 - Battery and cup filled with water
 - Insert the battery into the base station and weighing the cup with water
- High mobility in uneven terrain
 - Inclinations up to 30 degree
 - Stone fields and loose surfaces with different kinds of sand



Universität Bremen

Introduction

Mechatronic Design

Experiments

Outlook

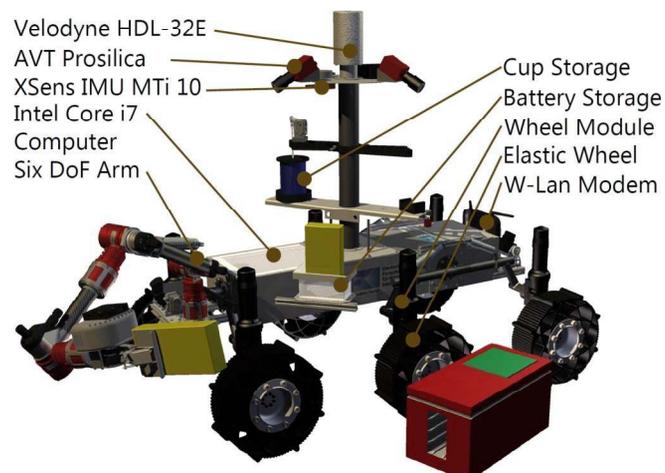
Contest site (Source: DLR)

3

Concept



Overview:



Universität Bremen

Introduction

Mechatronic Design

Experiments

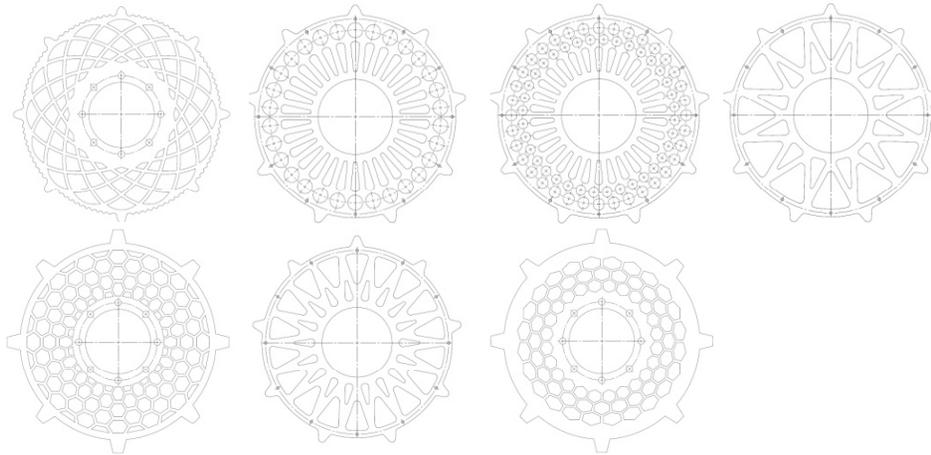
Outlook

4

Components



Flexible wheels:



Universität Bremen

Introduction

Mechatronic Design

Experiments

Outlook

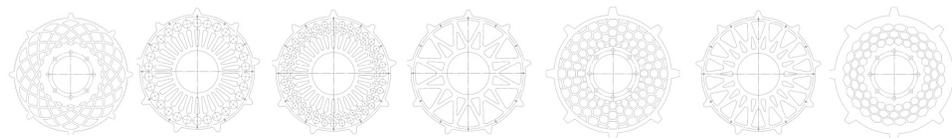
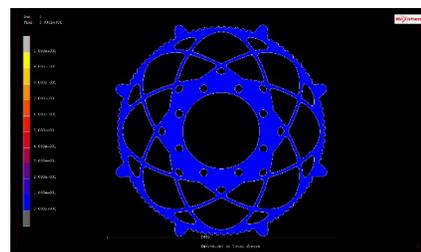
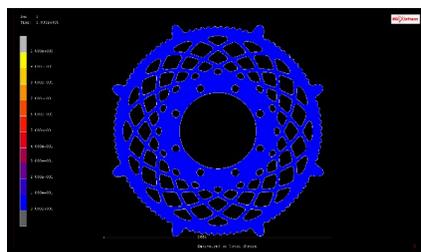
5

Components



Flexible wheels:

- Contact Body FEM Simulation



Universität Bremen

Introduction

Mechatronic Design

Experiments

Outlook

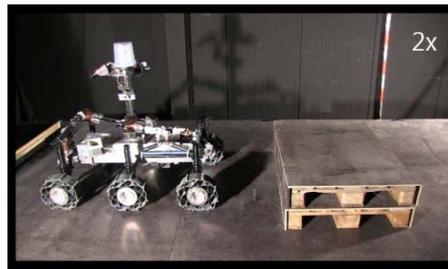
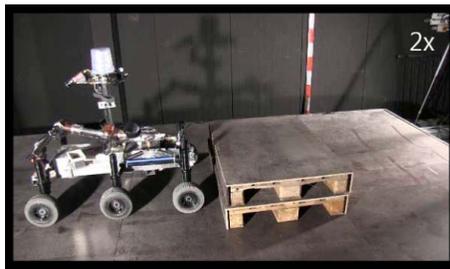
6

Experiments



Step Climbing:

- Obstacles of 1,5 of the Wheel diameter



Introduction

Mechatronic Design

Experiments

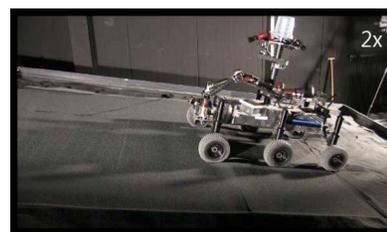
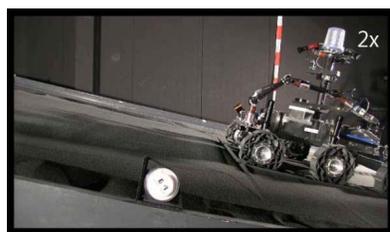
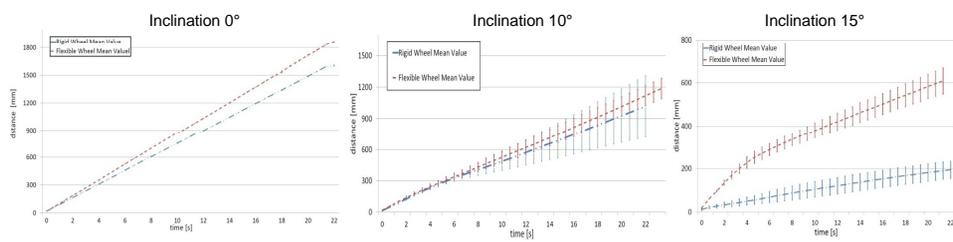
Outlook

7

Experiments



Wheel test and comparison:



Introduction

Mechatronic Design

Experiments

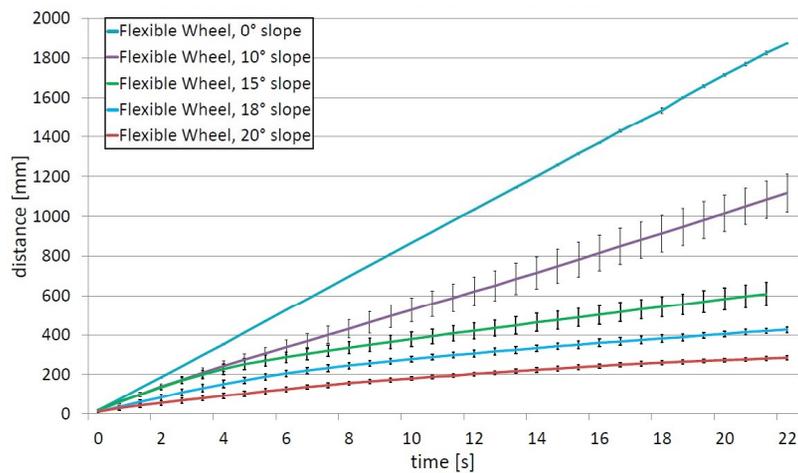
Outlook

8

Experiments



Wheel Test:



Conclusion



Lessens learned:

- For terrestrial robots are the flexible rubber wheels beneficial and will be continuously developed over the next years (TransTerra, SpaceBot)

Applying for the next SpaceBot Cup in 2015:

- Slight mechanical improvements of the flexible wheels will be tested and compared with the current version

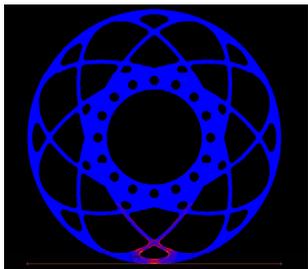
Outlook



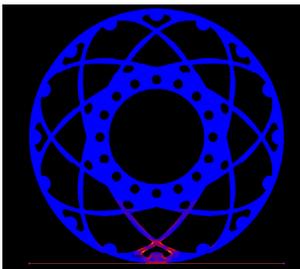
Flexible wheels:

- **New Design derived by Christian and Leon**

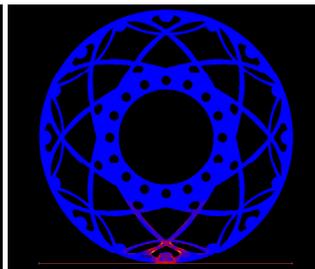
Tweels 2013



Tweels 2015 V1



Tweels 2015 V2



2.4 ‘SherpaTT Motion Control System’ (LM-T-04)

Ajish Babu⁽¹⁾, Florian Cordes⁽¹⁾, Daniel Kuehn⁽¹⁾

(1) DFKI GmbH, Robotics Innovation Center, Robert-Hooke-Straße 1, 28359 Bremen, Germany

Contact: ajish.babu@dfki.de, florian.cordes@dfki.de, daniel.kuehn@dfki.de

Abstract

The presentation describes the design of an active suspension system named SherpaTT, which is an improved design from the older version named Sherpa. It also describes the initial design of the Motion Control System (MCS) being developed for controlling the suspension system. MCS controls the posture of the robot body, the foot print of the legs, executes the motion commands, and implements variety of different compensators and safety checks. Preliminary results from the tests with the simulation are also presented.

SherpaTT Design

Objective

Conceptual Design

MCS in ROCK

Videos



SherpaTT Motion Control System (MCS)

Ajish Babu

DFKI Bremen & Universität Bremen
 Robotics Innovation Center
 Director: Prof. Dr. Frank Kirchner
www.dfki.de/robotics
robotics@dfki.de



SherpaTT Design

Objective

Conceptual Design

MCS in ROCK

Videos

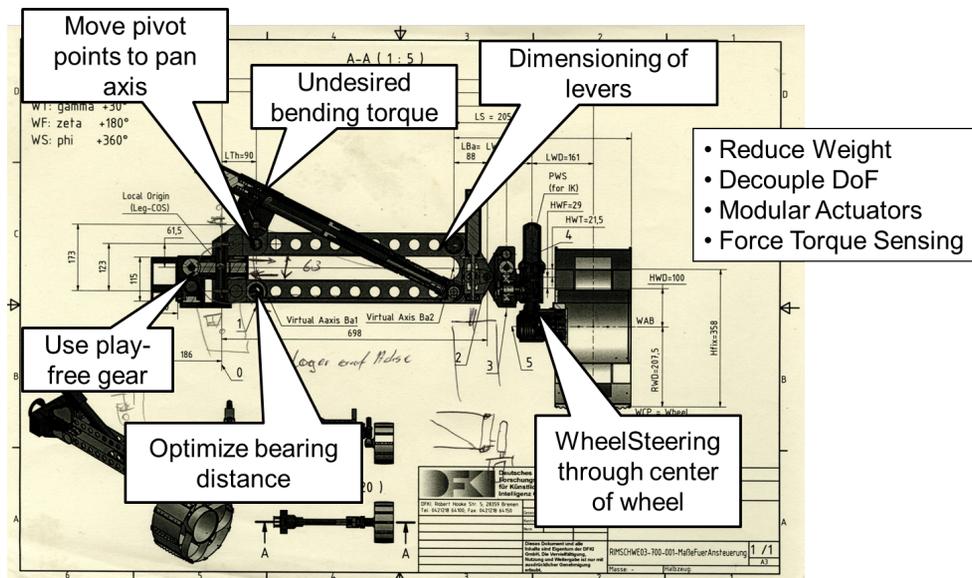


Table of Contents

- 1 SherpaTT Design
- 2 Objective
- 3 Conceptual Design
 - Supervisory Control
 - Motion Generators
 - Compensators
 - Core
- 4 MCS in ROCK
- 5 Videos

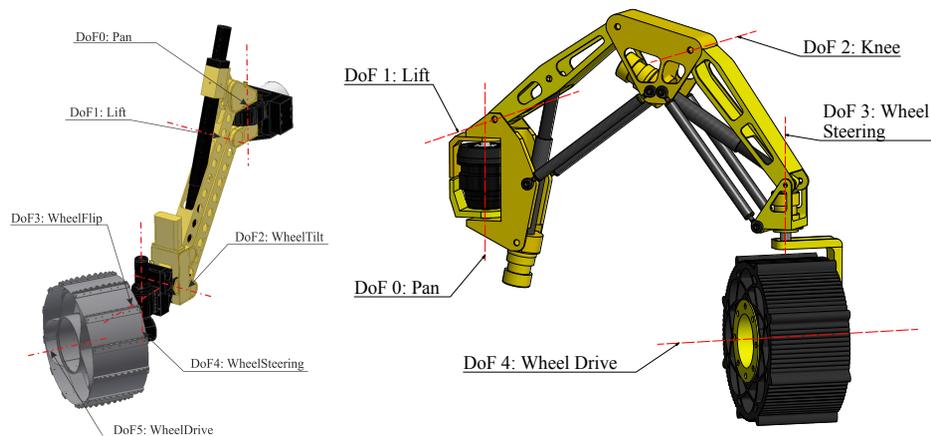


- 1 SherpaTT Design
- 2 Objective
- 3 Conceptual Design
 - Supervisory Control
 - Motion Generators
 - Compensators
 - Core
- 4 MCS in ROCK
- 5 Videos



SherpaTT Design Objective Conceptual Design MCS in ROCK Videos 

Old and new designs



 Universität Bremen SherpaTT Motion Control System (MCS) September 18, 2014 5/20

SherpaTT Design Objective Conceptual Design MCS in ROCK Videos 

Table of Contents

- 1 SherpaTT Design
- 2 Objective
- 3 Conceptual Design
 - Supervisory Control
 - Motion Generators
 - Compensators
 - Core
- 4 MCS in ROCK
- 5 Videos

 Universität Bremen SherpaTT Motion Control System (MCS) September 18, 2014 6/20

SherpaTT Design Objective Conceptual Design MCS in ROCK Videos 

Objective

Mission

- Traverse slope up to 15°
- Traverse 40-50 km
- Transport base camp
- Take soil sample
- Manipulate payloads

Added Features

- Active suspension system
- Morphing of different locomotion

 Universität Bremen SherpaTT Motion Control System (MCS) September 18, 2014 7/20

SherpaTT Design Objective Conceptual Design MCS in ROCK Videos 

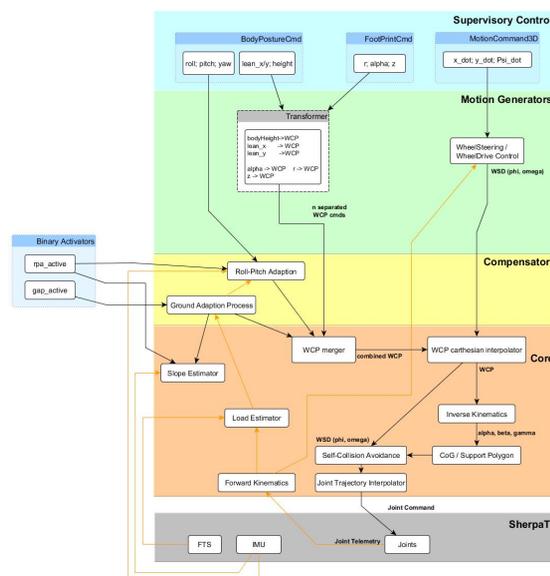
Objective

MCS

- Manage suspension system
- Reusable Modules
- Different locomotion patterns
- Compensator for different scenarios
- Safety (Self-Collision, Support Polygon)

 Universität Bremen SherpaTT Motion Control System (MCS) September 18, 2014 8/20

- 1 SherpaTT Design
- 2 Objective
- 3 Conceptual Design**
 - Supervisory Control
 - Motion Generators
 - Compensators
 - Core
- 4 MCS in ROCK
- 5 Videos



SherpaTT Design Objective Conceptual Design MCS in ROCK Videos 

Supervisory Control

Supervisors

- GUI
- Higher Level Navigation Module

Commands

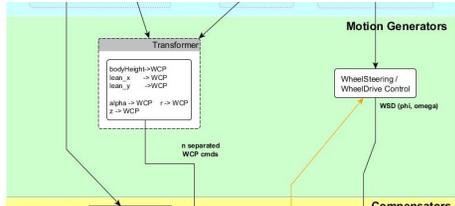
- Body Posture (Rigid-Body Pose)
- Leg Stance
 - Radius
 - Height
 - Yaw
- Motion Commands (OmniDrive)
 - Forward Velocity
 - Lateral Velocity
 - Turn Velocity



Universität Bremen SherpaTT Motion Control System (MCS) September 18, 2014 11/20

SherpaTT Design Objective Conceptual Design MCS in ROCK Videos 

Motion Generators



Motion Modes

- Omni-Directional Drive
- Inch-Worming
- Walking
- ...

Universität Bremen SherpaTT Motion Control System (MCS) September 18, 2014 12/20

SherpaTT Design Objective Conceptual Design MCS in ROCK Videos 

OmniDrive

Foot velocities

Linear velocity of a foot in BCS $v_{Fi}^B = v_B^B + \omega_B^B \times p_{Fi}^B$
 Linear velocity of a foot in FCS $v_{Fi}^{Fi} = R_B^{Fi} v_{Fi}^B$

where

p_{Fi}^B is the Foot position in BCS,
 $v_B^B = [v_x \ v_y \ 0]^T$ is the Linear velocity of the body in BCS,
 $\omega_B^B = [0 \ 0 \ \gamma]^T$ is the Angular velocity of the body in BCS,
 R_B^{Fi} is the rotation from B to Fi
 p_{Fi}^B is the foot position in BCS.

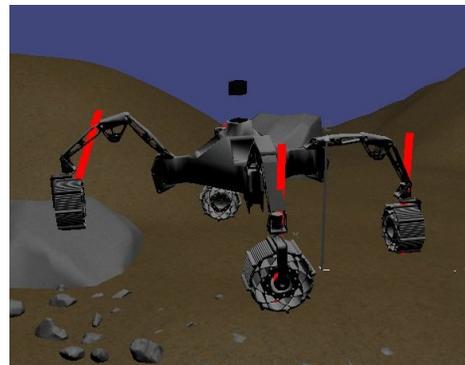
Wheel orientation no stance change

$$\phi_{Fi} = \arctan2(|v_{Fi}^{Fi}|_y, |v_{Fi}^{Fi}|_x)$$

 Universität Bremen SherpaTT Motion Control System (MCS) September 18, 2014 13/20

SherpaTT Design Objective Conceptual Design MCS in ROCK Videos 

Compensators



Ground Adaption Process (GAP)

- Ensures ground contact
- Force-leveling (1D so far)
- Mimics passive suspension

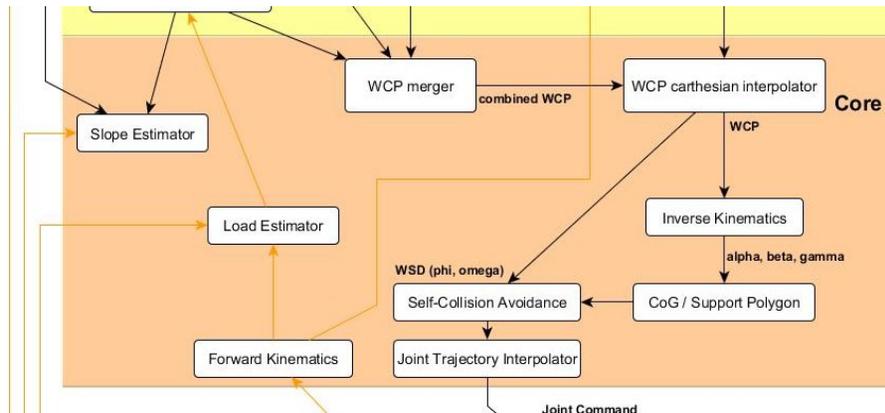
Roll-Pitch Adaption (RPA)

- To be implemented

Support Polygon Controller

- To be implemented

 Universität Bremen SherpaTT Motion Control System (MCS) September 18, 2014 14/20



Leg IK

- Converts Cartesian to Cylindrical coordinates
- Pan = Cylindrical angle
- Lift and Knee from 2D planar IK solution
- Parallel Kinematic makes rest just offsets

Equations

$$t_0 = \frac{x^2 + y^2 + l_1^2 - l_2^2}{2.0l_1}, t_1 = x^2 + y^2 - t_0^2, t_2 = \frac{x^2 + y^2 - l_1^2 - l_2^2}{2.0l_1 l_2}$$

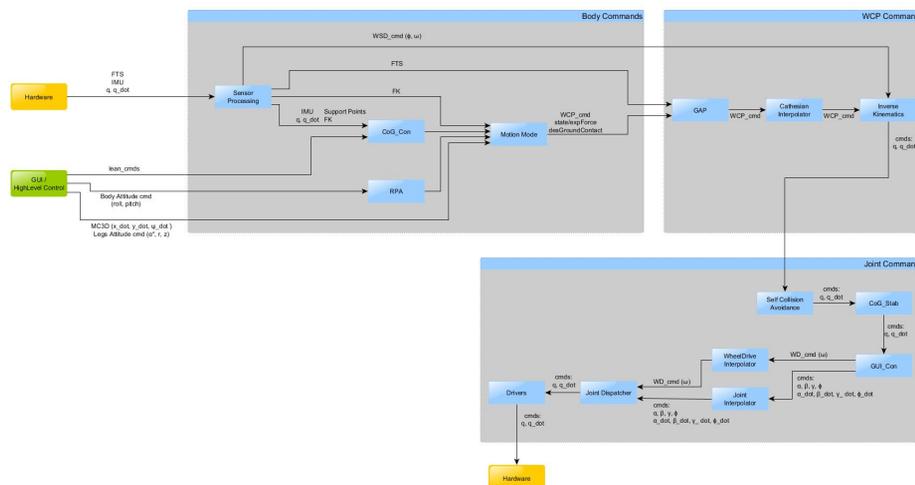
Solution exists only if $t_1 > 0$. If it exists

$$q_1 = 2.0 \operatorname{atan} \left(\frac{y \pm \sqrt{t_1}}{x + t_0} \right)$$

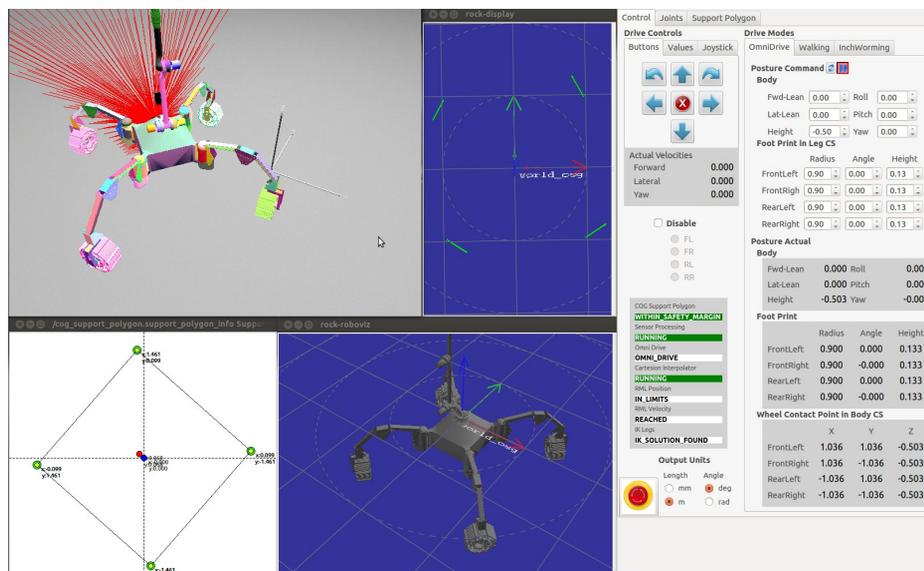
The two solutions from the above equation is used in

$$q_2 = q_1 \pm \operatorname{acos} (t_2)$$

- 1 SherpaTT Design
- 2 Objective
- 3 Conceptual Design
 - Supervisory Control
 - Motion Generators
 - Compensators
 - Core
- 4 MCS in ROCK
- 5 Videos



- 1 SherpaTT Design
- 2 Objective
- 3 Conceptual Design
 - Supervisory Control
 - Motion Generators
 - Compensators
 - Core
- 4 MCS in ROCK
- 5 Videos



Control | Joints | Support Polygon

Drive Controls: Buttons, Values, Joystick

Actual Velocities: Forward 0.000, Lateral 0.000, Yaw 0.000

Drive Modes: OmniDrive, Walking, InchWorming

Posture Command

Body			
Pwd-Lean	0.00	Roll	0.00
Lat-Lean	0.00	Pitch	0.00
Height	-0.50	Yaw	0.00
Foot Print In Leg CS			
	Radius	Angle	Height
FrontLeft	0.90	0.00	0.13
FrontRight	0.90	0.00	0.13
RearLeft	0.90	0.00	0.13
RearRight	0.90	0.00	0.13

Posture Actual

Body			
Pwd-Lean	0.000	Roll	0.000
Lat-Lean	0.000	Pitch	0.000
Height	-0.503	Yaw	-0.000

Foot Print

	Radius	Angle	Height
FrontLeft	0.900	0.000	0.133
FrontRight	0.900	-0.000	0.133
RearLeft	0.900	0.000	0.133
RearRight	0.900	-0.000	0.133

Wheel Contact Point in Body CS

	X	Y	Z
FrontLeft	1.036	1.036	-0.503
FrontRight	1.036	-1.036	-0.503
RearLeft	-1.036	1.036	-0.503
RearRight	-1.036	-1.036	-0.503

Output Units: Length (mm, m), Angle (deg, rad)

STATUS: WITHIN SAFETY MARGIN, Sensor Processing, OMNI_DRIVE, Gaiter Navigator, RUNNING, IN LIMITS, REACHED, H. SOLUTION FOUND

2.5 'Europa Explorer: Development of a Hydrodynamic Casing for the Autonomous Underwater Vehicle' (LM-P-01)

Philipp Kloss⁽¹⁾

(1) DFKI GmbH, Robotics Innovation Center, Robert-Hooke-Straße 1, 28359 Bremen, Germany

Contact: philipp.kloss@dfki.de

Abstract

The poster gives a compact summary about the „biological inspired“ shape design and shape optimization of the Autonomous Underwater Vehicle (AUV) from the project Europa-Explorer. An adequate shape design for the AUV minimizes the drag and elongates the mission time using active propulsion. In addition to that it supports as well the passive gliding. The main question was how to describe an assumed adequate biological shape from fishes or marine mammals technically and with which method this shape can be further optimized in a natural way with technical boundary conditions. The NACA (National Advisory Committee for Aeronautics)-Profile description offers a good possibility to describe the shape of a tuna technically, this built the basis for the template of the tuna shape design. In addition this „tuna“ shape forms the start geometry for the stern section shape optimization via an evolution strategy. The shape optimization is based on an evolutionary optimization algorithm coupled with numerical fluid dynamics. With this shape optimization method the drag could be reduced by 28,8% The result shows that the chosen shape and the optimization method are, in the present case, convenient to improve the hydrodynamic properties of the AUV.




Europa-Explorer

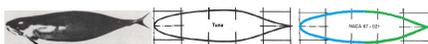
Development of a hydrodynamic casing for the autonomous underwater vehicle: „Biologically inspired“ shape design and shape optimization

Hydrodynamic casing main tasks

- Forms the supporting structure and give protection for the integrated devices
- Reduces the drag (C_w)
 - Supports active propulsion and passive gliding

“Biologically inspired” design

- Searching for a organism with locomotion in similar Reynolds range as the autonomous underwater vehicle (AUV; $Re > 6 \times 10^6$)
 - Similar fluiddynamic effects can be assumed
 - Tuna ($Re > 6 \times 10^6$; Alejev, 1977)
 - Streamlined body shape of tuna for design
 - NACA-Profiles (National Advisory Committee for Aeronautics) provide potential to describe natural body shapes like fishes and maritime mammals technically
 - “Tuna“-Profile NACA 67-021
 - Forms the geometry template for the bow section
 - Start geometry for the shape optimization of the stern section
 - Evolutionary strategy
 - Computational fluid dynamics (CFD)
 - The middle section remain cylindrical due to the project requirement to have the smallest possible AUV diameter



Comparison: Tuna and NACA-profile. **Blue:** Bow section template; **Green:** Start geometry for shape optimization of the stern section (changed according to Hertel, 1963).

On evolution strategy and CFD based shape optimization of stern section and stern thruster housing

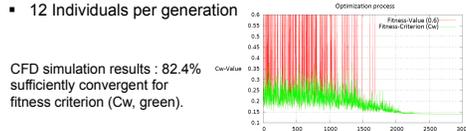
- Flow interaction between the stern section and stern thruster housing is considered for shape optimization
- Optimization method: 2D flow simulation (stern thruster dynamic impulse not implemented) coupled with evolutionary optimization algorithm
 - Fitness criterion: less drag
 - Stern section mutable in y-direction (six parameter)
 - Stern thruster housing mutable in x- and y-direction (eight parameter)
 - Overall stern thruster housing shiftable/mutable in x-direction
 - Dimension restrictions for mutation due to integrated components and mechanics



Left: CAD design of the not optimized stern section including stern thruster.
Right: 2D geometry (half profile) of the stern section and the stern thruster housing as start geometry for the shape optimization process. Mutable parameters are marked in red.

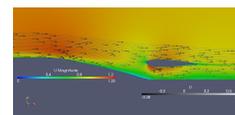
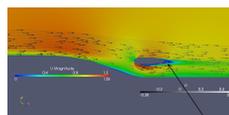
Optimization process

- Optimization method converge to a minimum (decreasing C_w)
- 3000 generated und simulated shapes (individuals)
 - 250 Generations
 - 12 Individuals per generation



Changes in geometry and flow fields

- Stern section in 75% of the length flattened
- Stern thruster housing:
 - In x-direction: 38mm shifted downstream
 - Stern of housing is 112% longer
 - Bow of housing is 48% longer and distinctly asymmetrical
 - No flow separation at the housing
- Fluid flow more efficiently channeled
- Drag reduced by 28.8%

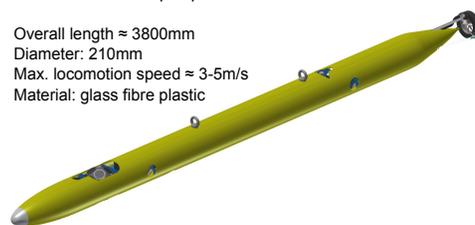


Flow separation (vortex)

Left top: CAD design with optimized stern section und stern thruster.
Right top: Geometry changes (grey and red: start geometry; blue and blue: optimized). Mutable parameters are marked in red and blue.
Left bottom: Flow fields of the non-optimized geometry.
Right bottom: Flow fields of the optimized geometry.
Velocity field (U magnitude) is shown in color in the background; Streamlines with flow direction are colored black and white according to the pressure field (P).

Resulting AUV shape

- Bow section → NACA-Profile
- Middle section → cylindrical
- Stern section → shape optimized
- Overall length ≈ 3800mm
- Diameter: 210mm
- Max. locomotion speed ≈ 3-5m/s
- Material: glass fibre plastic



CAD design of the AUV with hydrodynamic casing and integrated devices without stern thruster housing.

Alejev, Y.G. (1977): Nekton. Dr. W. Junk b.v., The Hague.
Hertel, H. (1963): Struktur - Form - Bewegung. Otto Krauskopf-Verlag GmbH., Mainz.
Generated by Philipp Kloss M.Sc.: philipp.kloss@dfki.de



Universität Bremen



DLR
(Grant No. 50 NA 1217)

Supported by:
Federal Ministry of Economics and Technology
on the basis of a decision by the German Bundestag

Contact:
DFKI Bremen & University of Bremen Robotics Innovation Center
Director: Prof. Dr. Frank Kirchner
E-mail: robotics@dfki.de
Website: www.dfk.de/robotics

2.6 'Introducing Particle Swarm Optimization into a Genetic Algorithm to Evolve Robot Controllers' (LM-P-02)

Malte Langosz⁽¹⁾, Kai A. von Szadkowski⁽³⁾

(1) DFKI GmbH, Robotics Innovation Center, Robert-Hooke-Straße 1, 28359 Bremen, Germany

(3) Universität Bremen, Arbeitsgruppe Robotik, Robert-Hooke-Straße 1, 28359 Bremen, Germany

Contact: malte.langosz@dfki.de, kai.von-szadkowski@uni-bremen.de

Abstract

This paper presents Swarm-Assisted Behavior Graph Evolution (SABRE), a genetic algorithm which combines elements from genetic programming and neuroevolution to develop Behavior Graphs (BGs). SABRE evolves graph structure and parameters in parallel using particle swarm optimization (PSO) for the latter. The algorithm's performance was evaluated on a set of black-box function approximation problems, one of which represents part of a robot controller. We found that SABRE performed significantly better in approximating the mathematically complex test functions than the reference algorithms genetic programming (GP) and NEAT.

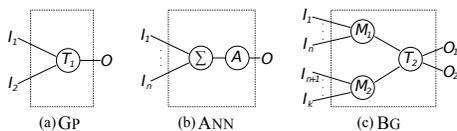
Introducing Particle Swarm Optimization into a Genetic Algorithm to Evolve Robot Controllers

Malte Langosz, Kai A. von Szadkowski, and Frank Kirchner

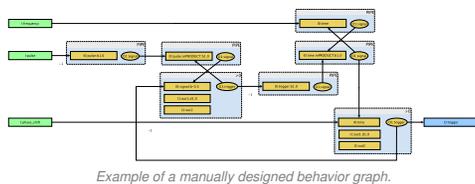
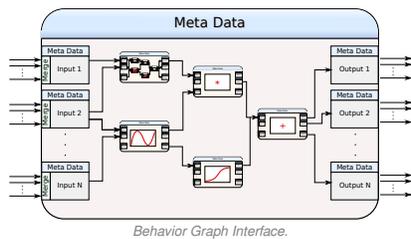
Introduction

This paper presents Swarm-Assisted Behavior Graph Evolution (SABRE), a genetic algorithm which combines elements from genetic programming and neuroevolution to develop Behavior Graphs (BGs). SABRE evolves graph structure and parameters in parallel using particle swarm optimization (PSO) for the latter. The algorithm's performance was evaluated on a set of black-box function approximation problems, one of which represents part of a robot controller.

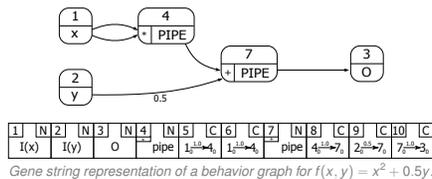
Behavior Representation



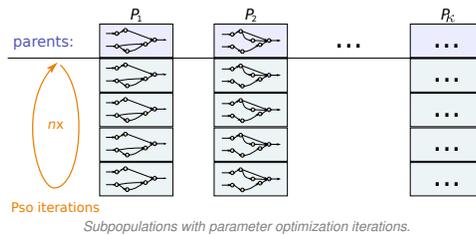
Different representations of nodes used in a GP tree (a), ANN such as used in NEAT (b), and BGs (c).



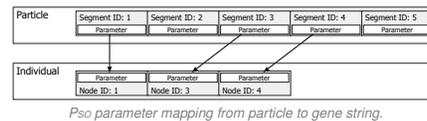
Swarm-Assisted Behavior Graph Evolution



SABRE evolves BGs via two nested loops, an outer one dealing with structure development and an inner one optimizing the parameters, iterating n times per outer loop iteration. To realize this, the population is split into subpopulations P_1 through P_c , whereby each subpopulation P_i represents one structure.

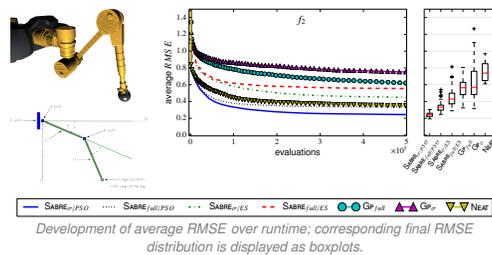


Whenever a new gene is created, it is assigned a unique identifier. In the PSO implementation, every particle of the swarm is given one segment per unique gene it maps to, i.e. the segment encoding as many parameters as required by the particular gene. The unique identifiers of the genes ensure correct mapping even when gene strings are cloned or of varying sizes.



Function approximation results

f_2 : Represents the inverse kinematics calculation of a leg of the SpaceClimber robot and possesses four inputs and four outputs.



Conclusion

As for the overall performance of SABRE, we can conclude from our data that for certain types of complex problems such as inverse kinematics of real robots, SABRE is able to produce superior results compared to standard GP and NEAT. It has to be noted that it may be possible to find configuration parameters for the two comparison algorithms changing these results, however we were unable to obtain such parameter sets despite various tests.

2.7 'The new Space Exploration Hall' (LM-P-03)

Daniel Kuehn⁽¹⁾, Florian Cordes⁽¹⁾

(1) DFKI GmbH, Robotics Innovation Center, Robert-Hooke-Straße 1, 28359 Bremen, Germany

Contact: daniel.kuehn@dfki.de, florian.cordes@dfki.de

Abstract

To specify a variety of complex and less complex robotic systems, to evaluate these and to make them comparable among themselves, a controlled laboratory environment for various experiments is required. Therefore, this poster presents an overview of the DFKI new space exploration hall. Here, at first only the dimensions of the hall and first considerations regarding possible building partition and equipment are given. This poster is intended as a basis for discussion and encourage the reader to develop own ideas and requirements which have to be communicated to the authors.

Locomotion Metrics and Test Facilities

Ideas for the new Space Exploration Hall in RH1

Discussion in the Workgroup Locomotion: How to Compare Different Systems?

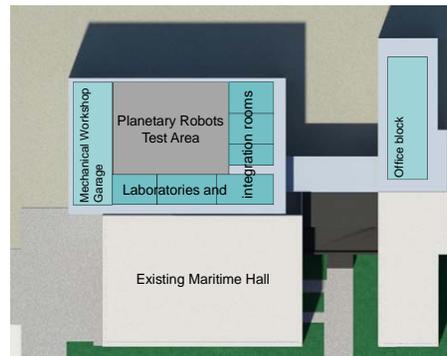
- Use simple quotients – ideally dimensionless
 - Range over robot length
 - Speed over robot length (→bodylength/second)
 - Range over energy consumption
 - Speed over energy consumption
 - Speed over weight
 - Top speed and top acceleration
 - Symmetry of sensor data (front-rear, left-right)
- "Sells-Man Metrics"
 - Costs, man-hours, physical dimensions, weight, range, number of DoFs, runtime, battery capacity
 - Number of sensors
 - Typical sensor sample rate(s)
 - Others
 - Time-to-boot, typical runtime after boot
 - Typical time between "system crashes"/unplanned reboots
- Single robot (evolution over time)
 - Traction on different grounds
 - "Preciseness" of movements
 - Positioning repeatability (of end-effectors and of overall system)
 - Typical odometry error
 - Responsiveness to change of direction in full speed
 - Energy consumption
 - Acceleration+jerk during movements



Overview: Extended DFKI estate at Robert-Hooke-Str. 1

Open Discussion: Needed Test Facilities and Equipment in new Space Exploration Hall (Planetary Exploration / in General: Walking and Driving Robots)

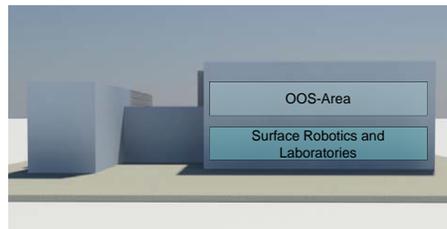
- Ideas for fixed installations
 - Crater as in current SpaceExplorationHall
 - Demonstrations
 - Robustness of locomotion behaviors (irregular terrain)
 - Something like the Darpa Robotic Challenge TestBed
 - SpaceBot Cup Field (loose sand, sand pits, hills...)
 - Treadmill including safety installation
 - Variable slope with exchangeable soil
 - Motion tracking system, maybe better: a mobile one?
- Ideas for mobile facilities and equipment
 - Small excavator or bulldozer for moving and shifting soils



Topview with possible division of the new Space Exploration Hall facilities

Ideas for Test Facilities for On-Orbit-Servicing (OOS)

- OOS part of the space hall could include
 - Gantry crane assembly
 - Vicon MTS
 - CableBot ("SpderCam") assembly
- "2nd Level" in SpaceHall: Above surface robotics test area
 - Height about 7-8m



Rear view

2.8 'Autonomous Path Tracking Steering Controller for Extraterrestrial Terrain Exploration Rover' (LM-P-04)

Mohammed Ahmed⁽¹⁾, Roland U. Sonsalla⁽¹⁾, Frank Kirchner^{(1) (3)}

(1) DFKI GmbH, Robotics Innovation Center, Robert-Hooke-Straße 1, 28359 Bremen, Germany

(3) Universität Bremen, Arbeitsgruppe Robotik, Robert-Hooke-Straße 1, 28359 Bremen, Germany

Contact: mohammed.ahmed@dfki.de, roland.sonsalla@dfki.de, frank.kirchner@dfki.de

Abstract

Extraterrestrial surface missions typically use a robotic rover platform to carry the science instrumentation (e.g., NASAs twin MER rovers). Due to the risks in the rover path (i.e. low trafficability of unrecognized soil patches), it is proposed in the FASTER project that a two rover system should be used. A micro scout rover is used for determining the traversability of the terrain and collaborate with a primary rover to lower the risk of entering hazardous areas. That will improve the mission safety and the effective traverse speed for planetary rover exploration. This poster presents the design and implementation of the path following controller for a micro scout rover. The objective is to synthesize a control law which allows the rover to autonomously follow a desired path in a stable manner. Furthermore, the software architecture controlling the rover and all of its subsystems is depicted. The performance of the designed controller is discussed and demonstrated with realistic simulations and experiments.



Autonomous Path Tracking Steering Controller for Extraterrestrial Terrain Exploration Rover

Mohammed Ahmed, Roland U. Sonsalla, Frank Kirchner – DFKI Robotics Innovation Center, Roboert-Hooke-Str. 1, 28359 Bremen, Germany

Forward Acquisition of Soil and Terrain Data for Exploration Rover (FASTER) Project

Analyzing past and future exploration missions and scenarios like MER or MSR, a need arises to provide faster and safer traversal of exploration rovers.

The aim of the FASTER project includes the concept development, implementation and demonstration of a system for in-situ evaluation of soil properties to improve the mission safety and the effective traverse speed for planetary rover exploration by determining the traversability of the terrain and lowering the risk to enter hazardous areas.

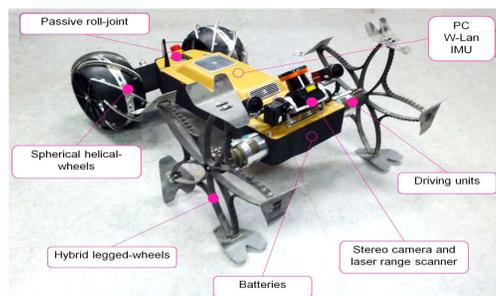


Scenario incorporates autonomous collaboration between a primary rover and a small and agile scout rover used for soil and terrain sensing ahead of a main rover

Coyote II (Scout Rover)

A highly mobile, and versatile micro scout rover that is designed for soil and terrain sensing.

The robot is equipped with a novel locomotion concept, to combine the benefits of wheeled and walking systems with its front hybrid legged and rear spherical helical wheels. In addition to its different sensors for autonomous navigation, it is equipped with two soil sensors (Wheel Leg Soil Interaction Observation (WLSIO) System and motorized Dynamic Cone Penetrometer (mDCP)) for forward sampling of soil and terrain characteristics

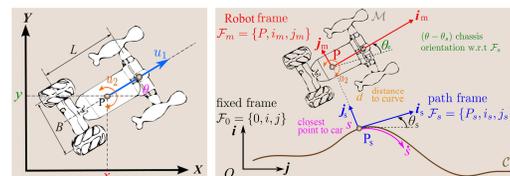


Scout rover test platform – system overview.

Path Following Controller

Implements how the robot tracks a reference path via the control of its wheels. It uses a kinematic model to approximate the nonholonomic robot mobility.

$$\begin{aligned} \dot{x} &= u_1 \cos \theta, & \dot{y} &= u_1 \sin \theta, & \dot{\theta} &= u_2 \\ \omega_r &= \frac{r}{2}(u_1 + Bu_2), & \omega_l &= \frac{1}{r}(u_1 - Bu_2) \end{aligned}$$

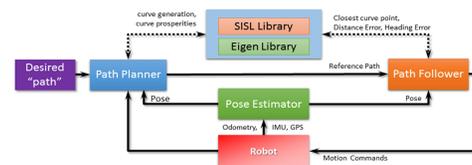


Kinematic model (left) and representation in Frénet frame (right)

A proportional input-scaling controller is used to control the robot to asymptotically stabilize ($d=0, \theta_e=0$) utilizing the canonical chained form of its model.

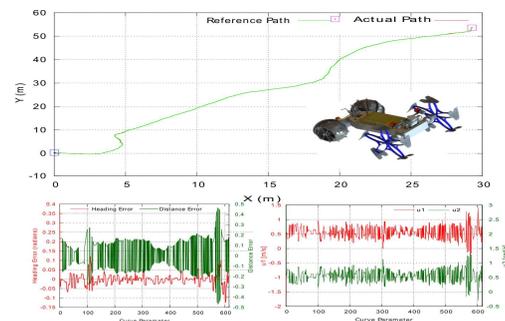
$$v_2 = -|v_1|k_0 \int_0^t v_1 z_2 - v_1 k_1 z_2 - |v_1|k_3 z_3$$

Where k_0 through k_3 are the controller parameters and z_1 through z_3 are the model chained form variables



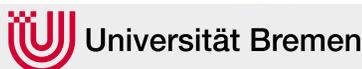
Controller implementation as a ROCK component on the robot.

Experiments and Results



Simulation results for a desired path from GPS coordinates of a field experiment. The robot follows the trajectory very closely as can be seen in the error graph.

This research project ‘Forward Acquisition of Soil and Terrain data for Exploration Rover (FASTER)’, running from 2011 to 2014, is supported by the European Commission through the SPACE theme of the FP7 Programme under Grant Agreement 284419. www.faster-fp7-space.eu



Contact:
DFKI Bremen & University of Bremen
Robotics Innovation Center
Director: Prof. Dr. Frank Kirchner
E-mail: robotics@dfki.de
Website: www.dfk.de/robotics

3 ‘Simulation’

3.1 ‘URDF, SMURF and Phobos - Better robot models for the future’ (SM-T-01)

Kai von Szadkowski⁽²⁾, Malte Langosz⁽¹⁾

(1) DFKI GmbH, Robotics Innovation Center, Robert-Hooke-Straße 1, 28359 Bremen, Germany

(2) Universität Bremen, Arbeitsgruppe Robotik, Robert-Hooke-Straße 1, 28359 Bremen, Germany

Contact: kai.von-szadkowski@uni-bremen.de, malte.langosz@dfki.de

Abstract

Robot models are used in various areas of robotics, such as motion planning, sensor processing or simulation. Often, a number of different formats have to be used to encode the data required by these areas, complicating the task of maintaining a consistent robot model. SMURF, the Supplementable, Mostly Universal Robot Format is an attempt to overcome such problems by extending the Unified Robot Description Format (URDF) with a generic data structure while maintaining full compatibility with URDF. To accomplish this, SMURF does not define a single file, but is comprised of a file tree, adding information encoded in YAML and thus allowing to both annotate URDF (e.g. attaching motors to joints or refining internal collision behavior) and add completely independent data. As files in the tree can link to contents of other YAML files and the tree is parsed recursively, this even allows to centrally save data on e.g. specific sensor and motor types, and re-use them across or between models. An open-source parser for SMURF is hosted on GitHub: https://github.com/rock-simulation/smurf_parser



URDF, SMURF and Phobos

Better robot models for the future

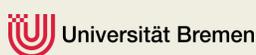
DFKI Bremen & Universität Bremen
Robotics Innovation Center
Director: Prof. Dr. Frank Kirchner
www.dfki.de/robotics
robotics@dfki.de



Motivation



- A (kinematic) robot model is used in different areas:
 - motion planning
 - simulation
 - ...
- The status quo @ DFKI:
 - Ruby: transformation stack for Rock
 - URDF: ROS, kinematic solvers
 - MARS scenes: simulation
- The desirable status:
 - one format that fits all tasks



URDF

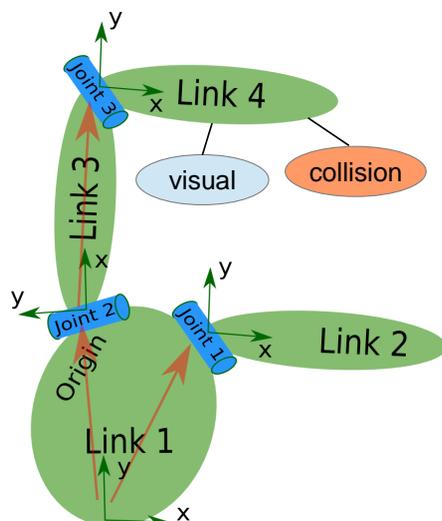


- “Unified Robot Description Format”
- XML format for a robot's kinematic representation
- developed for ROS

URDF



- “Unified Robot Description Format”



```

<robot name="test_robot">
  <link name="link1" />
  <link name="link2" />
  <link name="link3" />

  <joint name="joint1" type="continuous">
    <parent link="link1"/>
    <child link="link2"/>
    <origin xyz="5 3 0" rpy="0 0 0" />
  </joint>

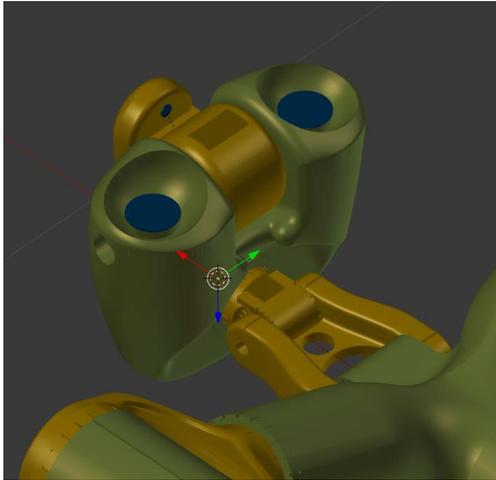
  <joint name="joint2" type="continuous">
    <parent link="link1"/>
    <child link="link3"/>
    <origin xyz="-2 5 0" rpy="0 0 1.57" />
  </joint>
</robot>

```

URDF



- “Unified Robot Description Format”

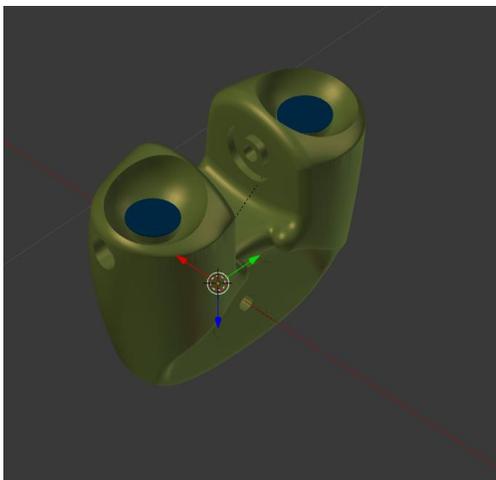


```
<robot name="mantis">  
...  
<link name="head" />  
<visual>  
  <geometry>  
    <mesh filename="head.obj"/>  
  </geometry>  
  <material name="green"/>  
</visual>  
</link>  
...  
</robot>
```

URDF



- “Unified Robot Description Format”

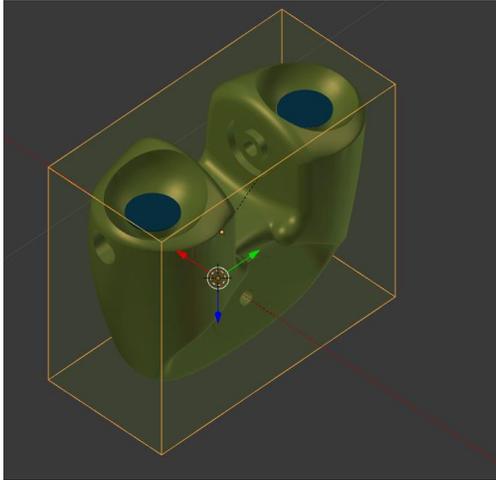


```
<robot name="mantis">  
...  
<link name="head" />  
<visual>  
  <geometry>  
    <mesh filename="head.obj"/>  
  </geometry>  
  <material name="green"/>  
</visual>  
</link>  
...  
</robot>
```

URDF



- “Unified Robot Description Format”

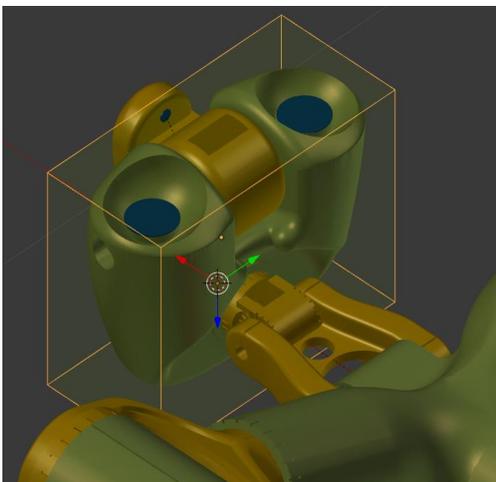


```
<robot name="mantis">
...
<link name="head" />
<visual>
  <geometry>
    <mesh filename="head.obj"/>
  </geometry>
  <material name="green"/>
</visual>
<collision>
  <geometry>
    <box size="0.1 0.2 0.2"/>
  </geometry>
</collision>
</link>
...
</robot>
```

URDF

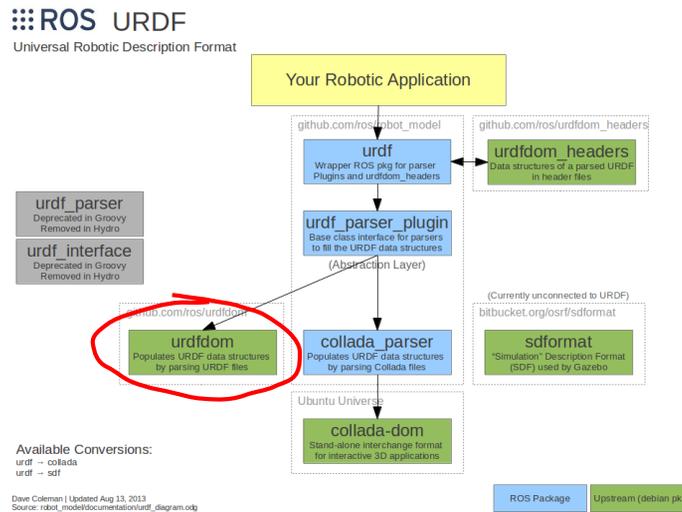


- “Unified Robot Description Format”



```
<robot name="mantis">
...
<link name="head" />
<visual>
  <geometry>
    <mesh filename="head.obj"/>
  </geometry>
  <material name="green"/>
</visual>
<collision>
  <geometry>
    <box size="0.1 0.2 0.2"/>
  </geometry>
</collision>
</link>
...
</robot>
```

URDF



URDF's upsides & downsides



• Up

- Community
- Tools (ROS-dependent)
- Documentation
- existing parser

Down

- formatting not strictly conform to standard XML
- structure very different from MARS scene
- limited information:
 - collision groups (SRDF)
 - sensors
 - materials
 - motors
 - environment

Extending URDF



- using URDF, but not altering it (compatibility)
- two options:
 - (1) additional URDF tags
 - (2) “annotate” URDF in separate file(s)

(1) Gazebo:

- <gazebo> elements

(2) DFKI:

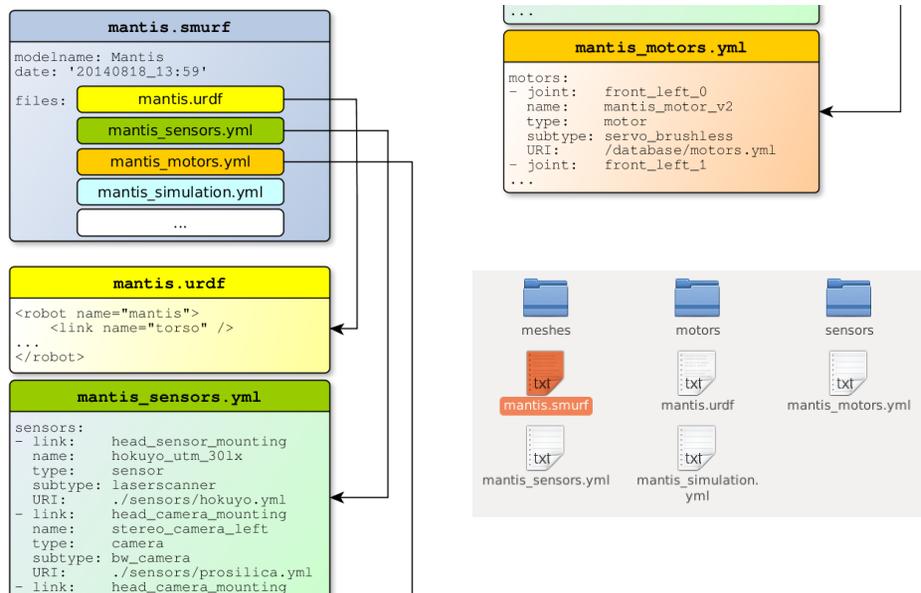
- SMURF !

SMURF



- **S**upplementable, **M**ostly **U**niversal **R**obot **F**ormat
- YAML format that extends URDF

SMURF



Status



- ✓ first draft SMURF specification
- ✓ import SMURF / URDF in MARS simulation
- ✓ editing tools
- ✓ migrating models:
 - ✓ Charlie
 - ✓ SpaceClimber
 - Mantis
- ✗ comprehensive documentation
- ✗ rock integration

So, I have a robot model...?



- MARS remains backward-compatible
- migration made easy:
 - import MARS scenes in 3D editor
 - modify what's necessary
 - export model in SMURF
- creation from scratch:
 - CAD2SIM → URDF → SMURF
 - CAD data export as STL → SMURF

Future development



- SMURF
 - finalize specification
 - scenes: multiple SMURFs in one scene
 - ▶ physics (gravity), GUI elements...
 - dynamic environments: configure MARS plugins which dynamically generate objects (obstacle fields etc..)
 - modular SMURFs
 - ...
- use SMURF in Rock
 - replace Ruby scripts for transformation stack with URDF
 - store parameters of robot and simulation alongside one another

Support



Kai von Szadkowski



Malte Langosz



Stefan Rahms



Ole Schwegert

3.2 ‘Phobos: 3D Robot Modelling made easy’ (SM-T-02)

Kai von Szadkowski⁽²⁾, Malte Langosz⁽¹⁾

(1) DFKI GmbH, Robotics Innovation Center, Robert-Hooke-Straße 1, 28359 Bremen, Germany

(2) Universität Bremen, Arbeitsgruppe Robotik, Robert-Hooke-Straße 1, 28359 Bremen, Germany

Contact: kai.von-szadkowski@uni-bremen.de, malte.langosz@dfki.de

Abstract

Phobos is an Add-On for the free 3D modeling software Blender, turning it into a powerful open-source WYSIWYG editor for robot models to be used with MARS or Gazebo. The software allows to create robots from scratch or derive them from imported CAD data - it’s even possible to import existing URDF files, which are widely used with ROS. Consequently, robot models can be exported to URDF, but also in SMURF, the DFKI’s versatile robot data format based on URDF and compatible with Rock. To represent the hierarchical kinematic structure of URDF models, Phobos makes use of Blender’s armature objects, which are designed for character animation and thus allow constraining movements similar to URDF’s joint constraints. This enables verifying ranges of motion of a robot directly in Blender. By attaching meshes or primitives to the bones, Phobos simplifies adding visual and collision objects to a model. Additional objects with custom properties allow to place and orient sensors such as laser scanners and cameras, or refine the physical representation of a robot by redefining centers of masses and inertia tensors. In fact, any custom information the user wishes to add for a specific task can be written into the Blender file and will be exported to SMURF, thus enabling the use of Phobos as the principal model editor over the lifetime of a robot. Phobos is hosted on GitHub: <https://github.com/rock-simulation/phobos>



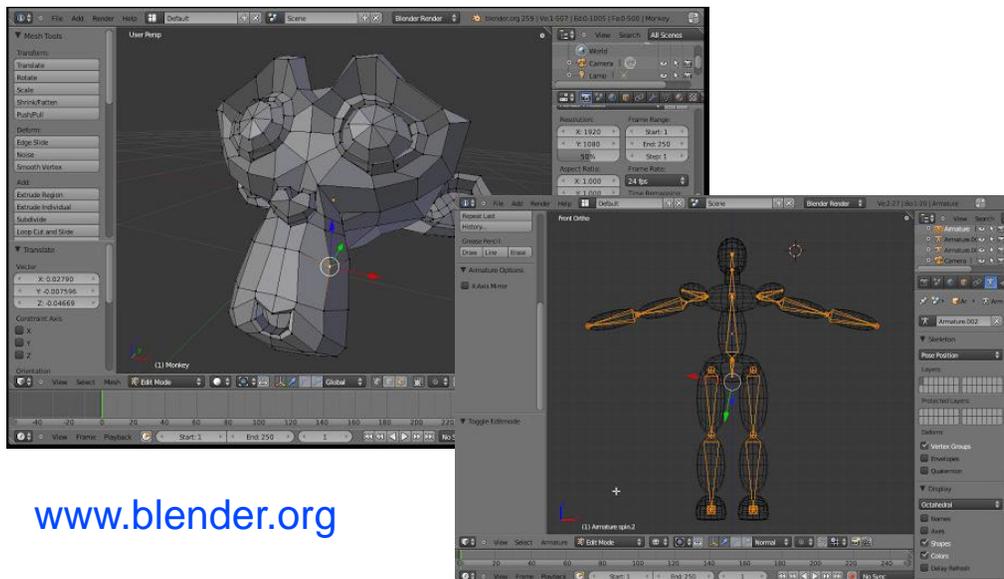
Phobos

3D Robot Modelling made easy

DFKI Bremen & Universität Bremen
Robotics Innovation Center
Director: Prof. Dr. Frank Kirchner
www.dfki.de/robotics
robotics@dfki.de



Blender



www.blender.org



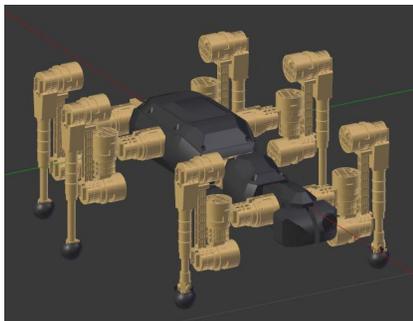
Phobos



- A Blender plugin for creating robots (for MARS)
- allows to import, edit and export URDF and SMURF



Robot representation

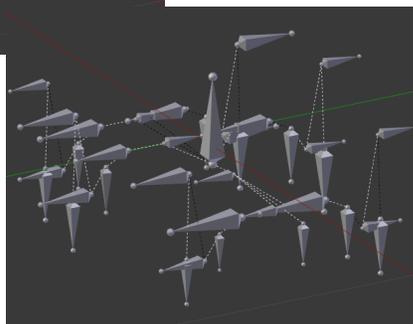


visuals

Robot representation

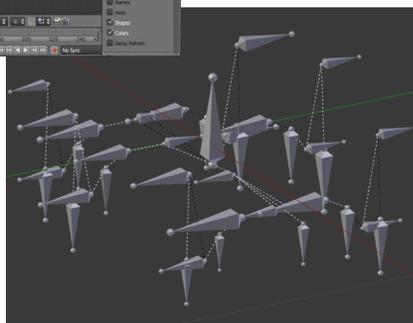
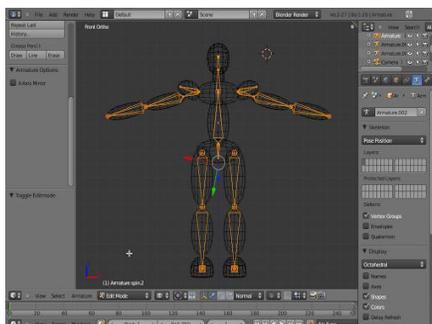


visuals



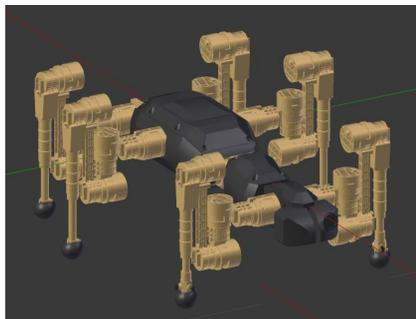
links

Robot representation

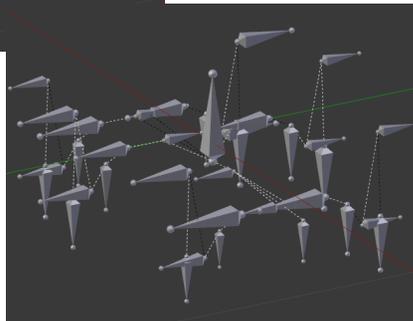


links

Robot representation

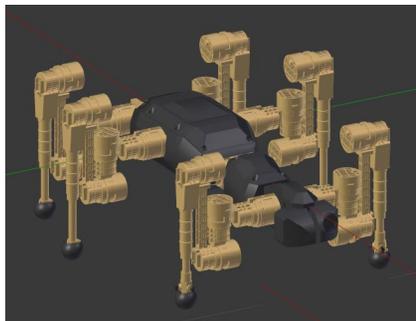


visuals

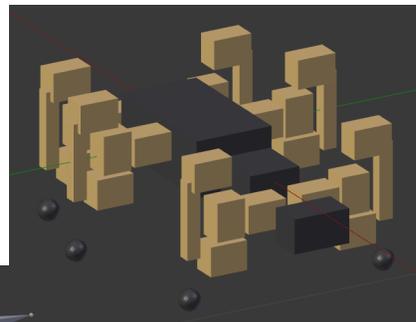


links

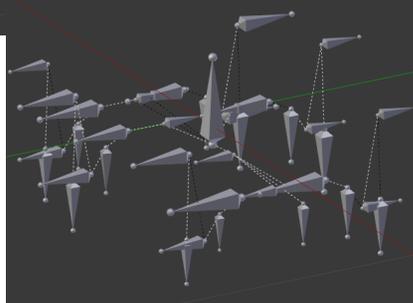
Robot representation



visuals

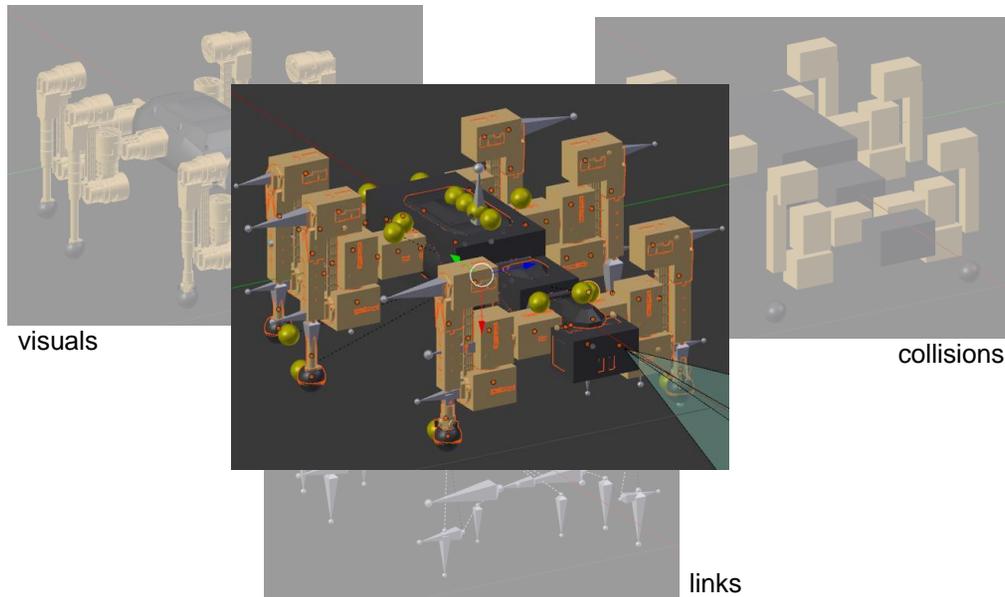


collisions

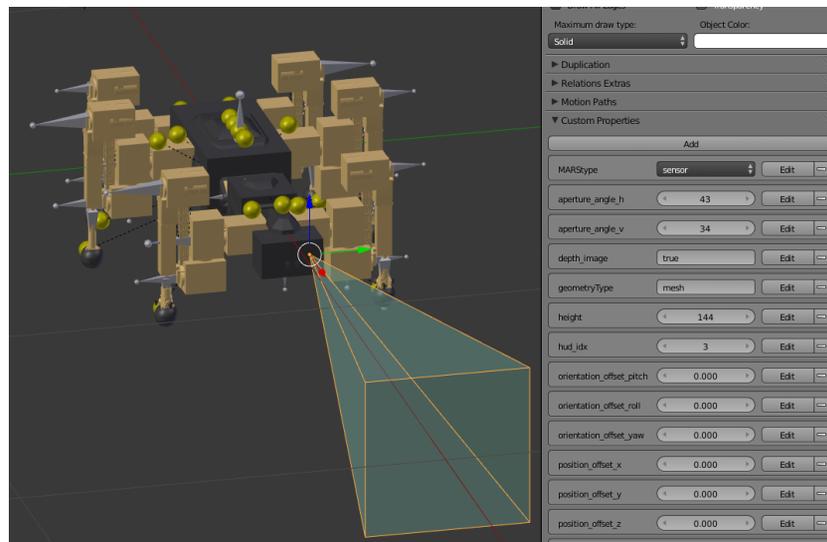


links

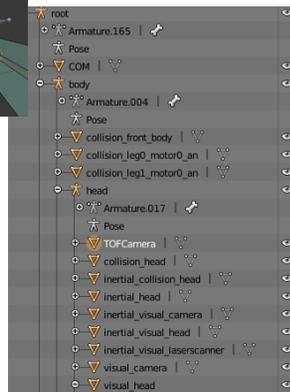
Robot representation



Custom properties



Robot data in Phobos



- Blender objects tree
- internal Python dictionary
 - contains all robot data
 - “middle man” for import & export
- customizable data
 - Blender custom properties for every object
 - exported to SMURF

Import / Export



- Import
 - URDF / SMURF *
 - MARS scene *
 - SDF **
- Export
 - YAML
 - URDF / SRDF * / SMURF *
 - MARS scene **
 - SDF **
 - meshes (.obj, .bobj, .stl)



* work in progress ** currently-not-planned nice-to-haves

Next steps...



- ship version 1.0
 - bugfixes
 - tutorials & documentation
- Phobos meets ROS
 - getting people interested
 - support for Gazebo (SDF)
- your feedback



Getting Phobos...



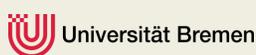
The screenshot shows the GitHub interface for the repository `rock-simulation / phobos`. The file `phobos / doc / installation.md` is selected, showing a commit by `kavonszadkowski` from 26 days ago. The file content includes a heading `Installing Phobos {#phobos_installation}` and a section for `Linux` installation instructions.

Installing Phobos {#phobos_installation}

Phobos is an Add-On for Blender which simplifies editing robot models and environments to be used in the MARS simulation. Phobos can also be used to visually create urdf files for ROS. This document aims to provide an instruction to set up blender and install the phobos Add-On.

Linux

The simplest way to install blender in a linux environment is to use your favourite package manager to install blender. In Ubuntu for example just type `apt-get install blender` into a terminal with root permissions. If you want to use the



Support



Kai von Szadkowski



Stefan Rahms



Ole Schwegert

3.3 ‘IMPAC SOTTL’ (SM-T-03)

Yong-Ho Yoo⁽¹⁾

(1) DFKI GmbH, Robotics Innovation Center, Robert-Hooke-Straße 1, 28359 Bremen, Germany

Contact: yong-ho.yoo@dfki.de

Abstract

In the DFKI industry project IMPaC SOTLL, we developed the 6-DOF robot crane arm concept for a floating LNG terminal. This talk presents some of the results from the project about the modeling and simulation of the crane robot control system for the transfer of the loading device between a floating LNG terminal and a LNG cargo. One of the key figures for the crane robot control system is the robot arm kinematics. In order to prove the concept of the crane robot arm, the forward and inverse kinematics modeling of the robot arm has been developed and implemented in MATLAB / ADAMS co-simulation environment. To test the SOTLL whole scenario of the physical behavior of the system, such as position, speed and forces, as well as to evaluate accurately and graphically the 3D movements of the crane and the ships, the 3D simulation of the floating LNG terminal and the LNG cargo was built in the multi-body simulator ADAMS. To realize the motion of the floating LNG terminal and the LNG cargo in the simulation, the motion inputs provided by the customer records of vessel movements were used for the ship movements in the ADAMS simulation.



IMPac SOTTL

Kinematics and Controls for Sideways Offshore Transfer of LNG Terminal

Yong-Ho Yoo

DFKI Bremen & Universität Bremen
 Robotics Innovations Center
 Director: Prof. Dr. Frank Kirchner
www.dfki.de/robotics
robotics@dfki.de



A bit of background of offshore LNG transfer



- Floating Liquefied Natural Gas (FLNG) solutions:
 - produces, stores and periodically transfers liquefied gas to shuttle tankers
- LNG is a hazardous cargo:
 - cryogenic conditions of -162°C
 - liquid volume that is approximately $1/600^{\text{th}}$ of the gas volume
- Cryogenic Transfer Technologies:
 - tandem offloading, when the LNG carrier is aligned with the FLNG with a safety distance.
 - side-by-side offloading, when the LNG carrier is moored on the side of the FLNG.



Tandem offloading: Technip



Side-by-side offloading: FMC



Work overview

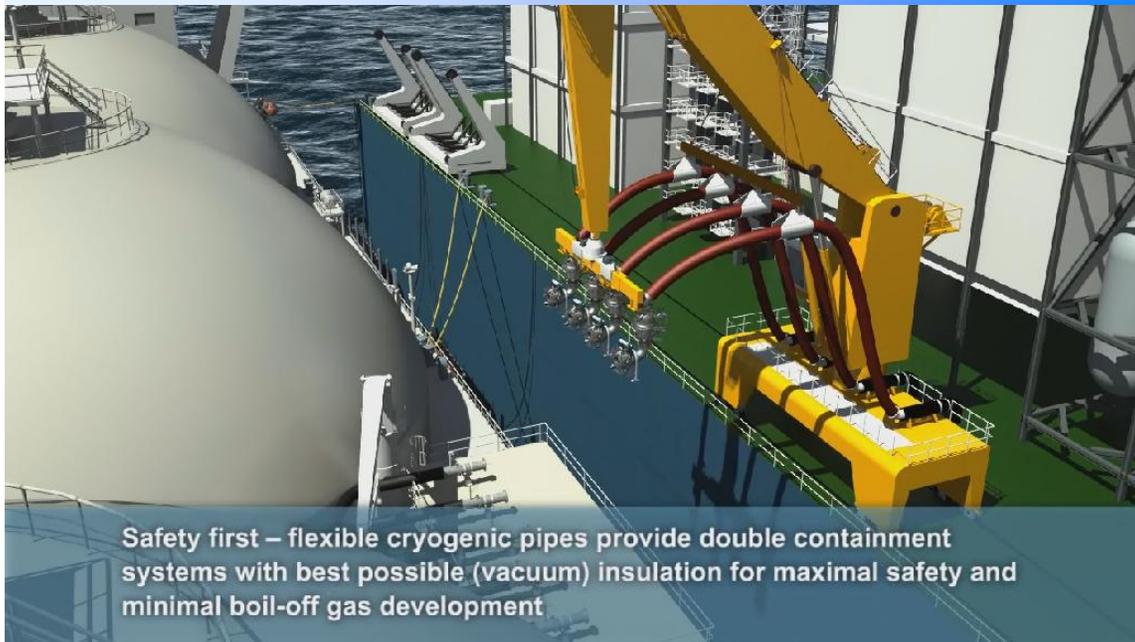


- Goal
 - Modeling and control of a crane system for *side-by-side offloading* between a floating LNG terminal (FLNG) and a LNG cargo (LNGC).
- Kinematics
 - Forward/Inverse Kinematics
 - Verification
- Dynamic Simulation and Controls
 - Crane dynamics
 - LNGC + FLNG dynamics
 - Overall dynamics: Crane + LNGC + FLNG

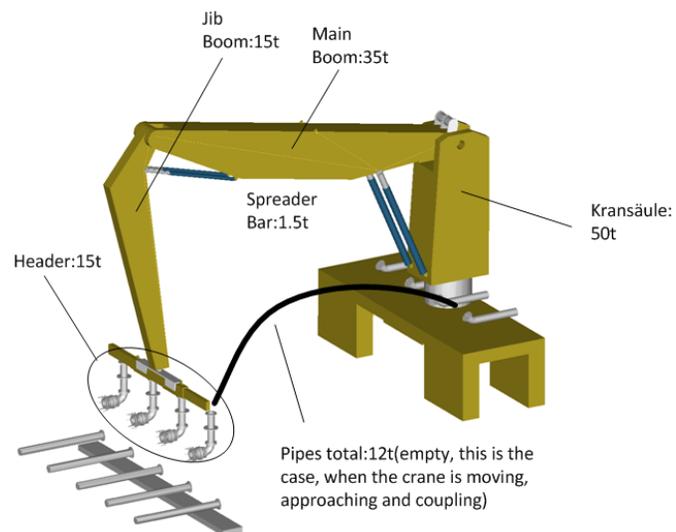
Current solution



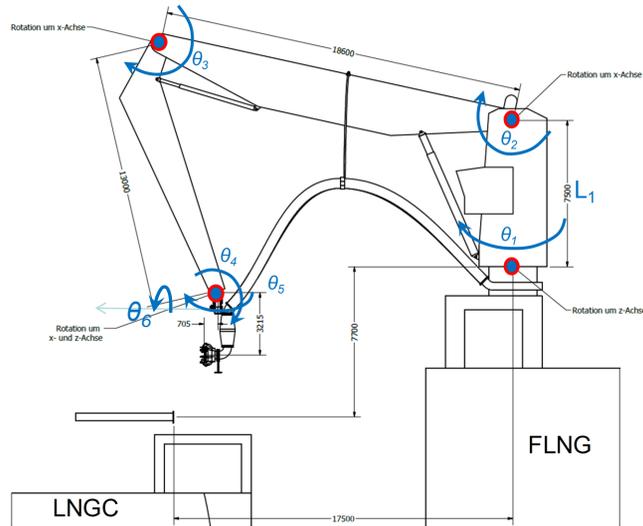
Developing solution



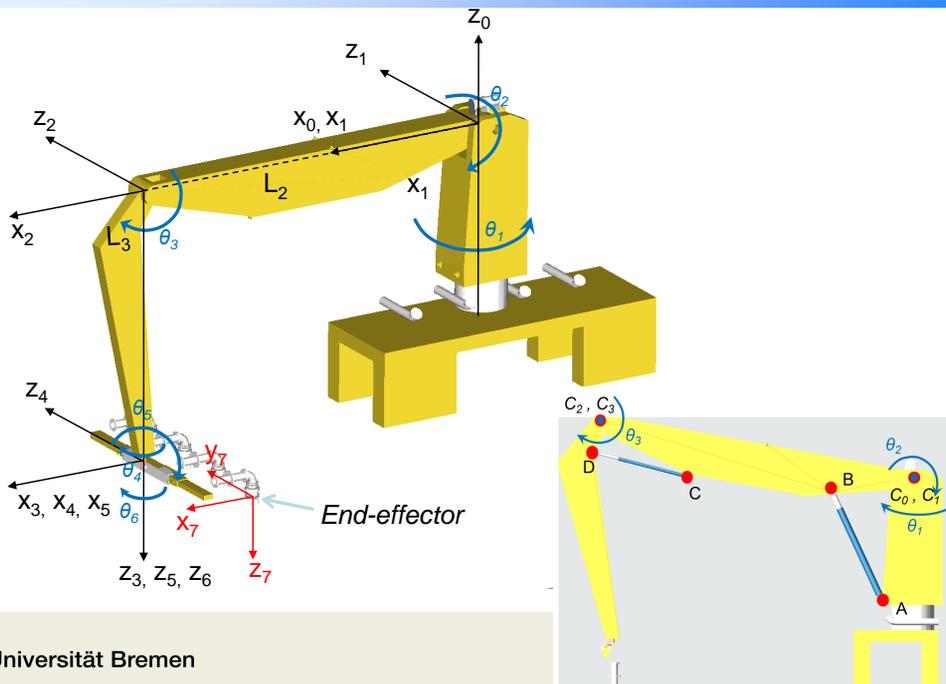
The crane concept



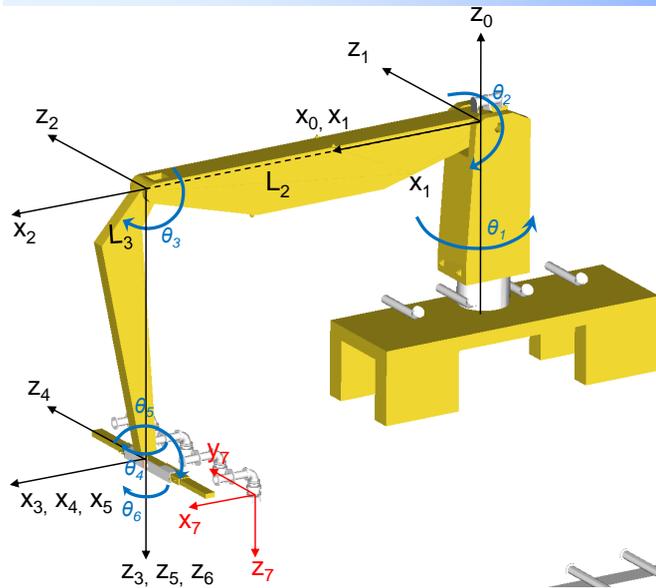
The crane concept



Joint coordinate system



Forward Kinematics



DH (Denavit-Hartenberg) parameter

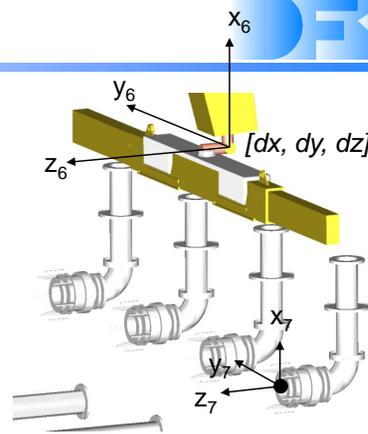
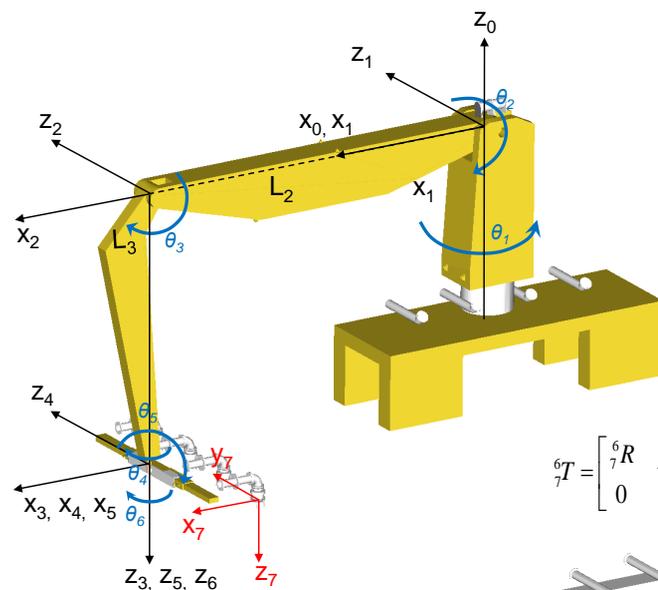
	a_i	α_i	d_i	θ_i
1	0	90°	0	θ_1
2	L_2	0	0	θ_2
3	0	90°	0	θ_3
4	L_3	-90°	0	θ_4
5	0	90°	0	θ_5
6	0	0	0	θ_6

Position and orientation of the end-effector:

$${}^0T_7 = {}^0T_1 {}^1T_2 {}^2T_3 {}^3T_4 {}^4T_5 {}^5T_6 {}^6T_7$$

$$T = \begin{bmatrix} R & p \\ 0 & 1 \end{bmatrix}$$

Inverse Kinematics



$${}^6T_7 = \begin{bmatrix} {}^6R_7 & p_{67} \\ 0 & 1 \end{bmatrix}$$

$${}^0T_7 = {}^0T_1 {}^1T_2 {}^2T_3 {}^3T_4 {}^4T_5 {}^5T_6 {}^6T_7$$

$${}^0T_6 = {}^0T_7 {}^6T_7^{-1}$$

$$p_{67} = [x_{67} \ y_{67} \ z_{67}]$$

Inverse Kinematics



$${}^0T_7 = \text{input}$$

$$p_{67} = [x_{67} \ y_{67} \ z_{67}]$$

$${}^6T_7 = \begin{bmatrix} {}^6R_7 & p_{67} \\ \mathbf{0} & 1 \end{bmatrix}$$

$${}^0T_7 = {}^0T_7 {}^6T_7^{-1}$$

θ_1 from dx, dy

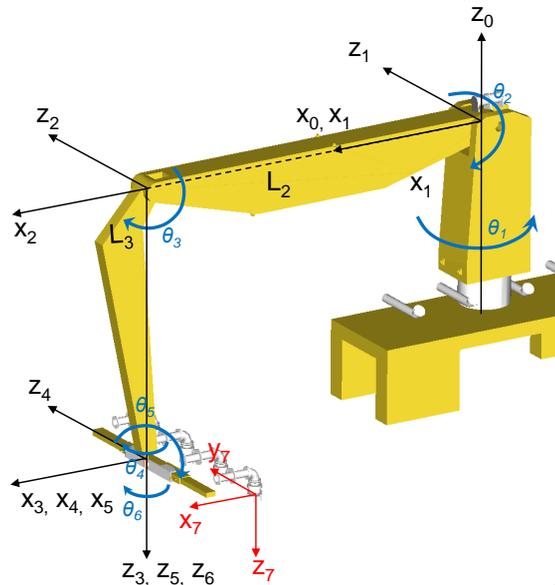
θ_2 from $a \cos \theta_2 + b \sin \theta_2 = c$

θ_3 from L_2, L_3, dx, dy, dz

0T_3 from ${}^0T_7 = {}^0T_1 {}^1T_2 {}^2T_3 {}^3T_4 {}^4T_5 {}^5T_6$

$${}^3R_6 = {}^0R_3^{-1} {}^0R_6$$

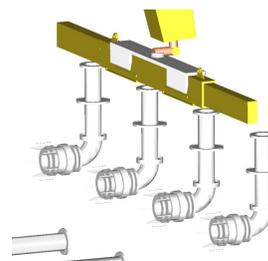
$\theta_4, \theta_5, \theta_6$ from 3R_6



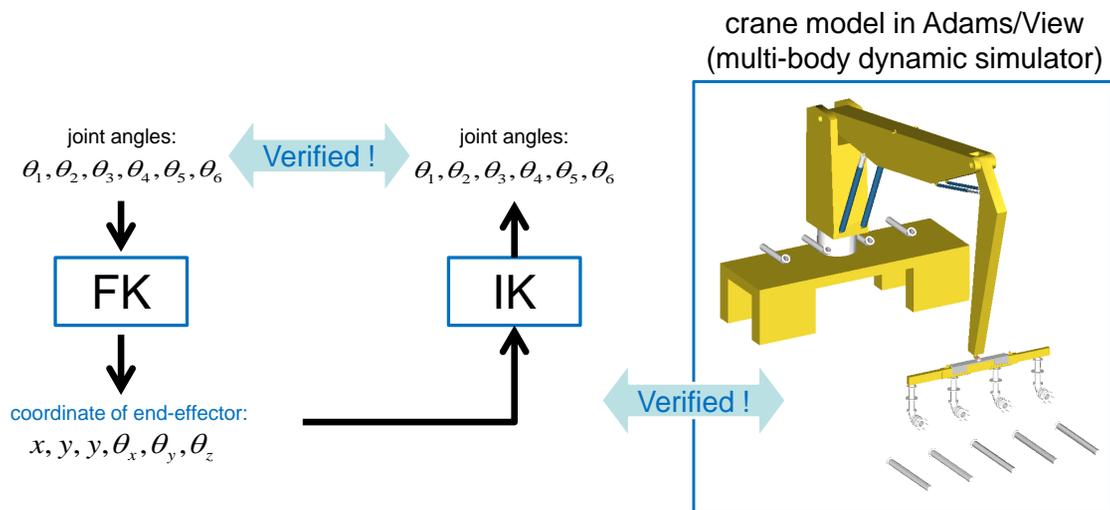
Configuration



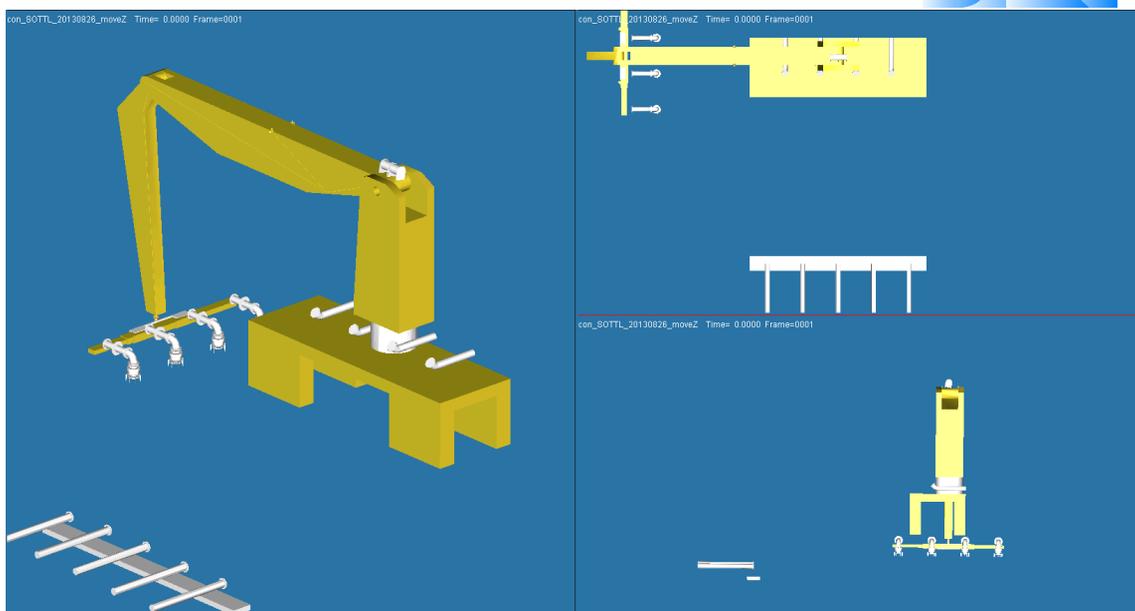
- Motion input of End-effector
 - Position (x, y, z) from zero position
 - Rotation matrix or Euler angles (Rx, Ry, Rz)
 - ▶ zero orientation: Rot(0°, 0°, 0°)
 - ▶ starting orientation: Rot(-90°, 0°, -90°)
- Position parameters of End-effector are selectable (static)



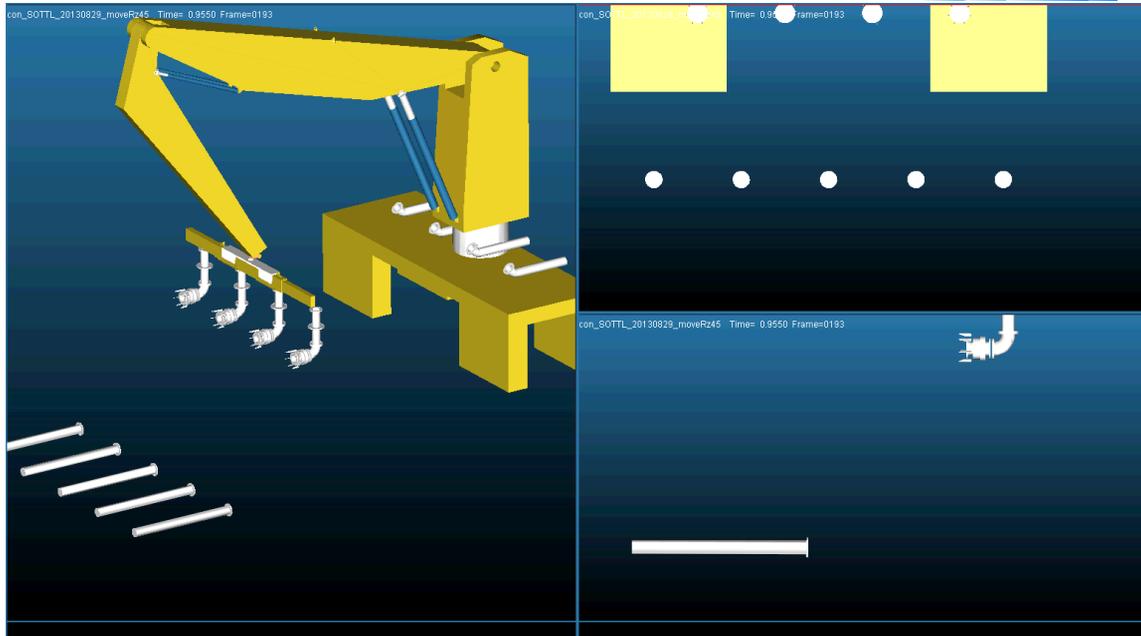
Verification in Adams/Matlab Co-Sim



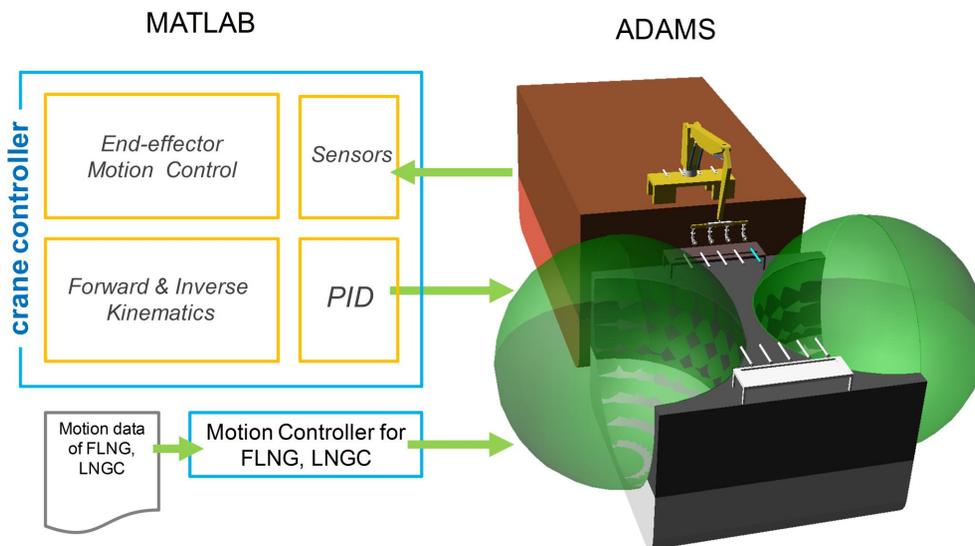
Position Controls



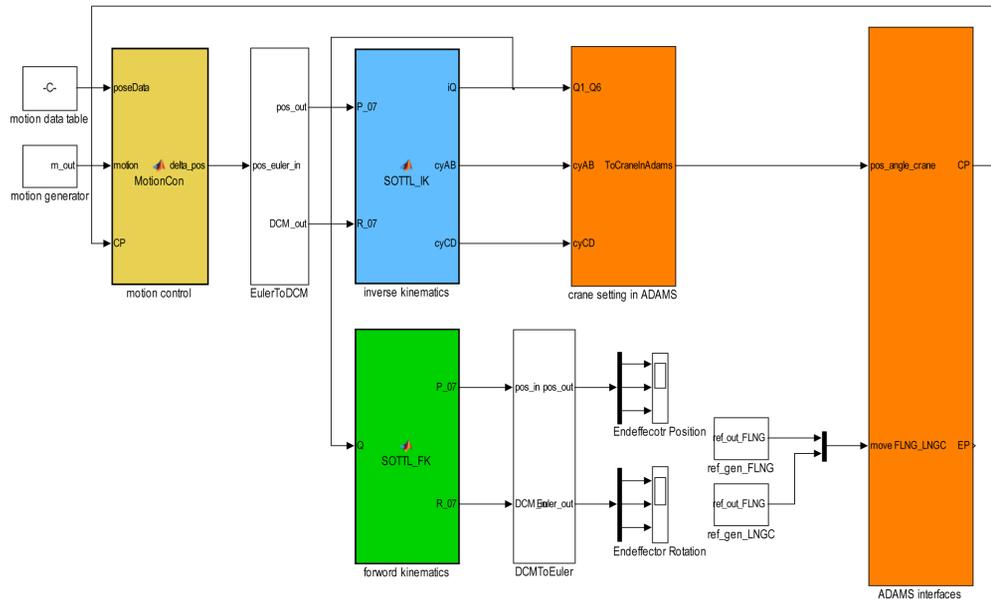
Position + Orientation Controls



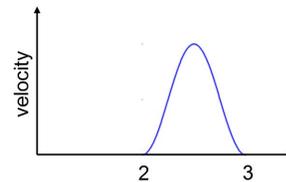
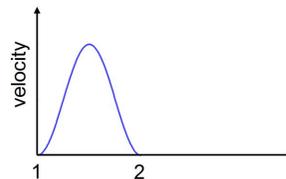
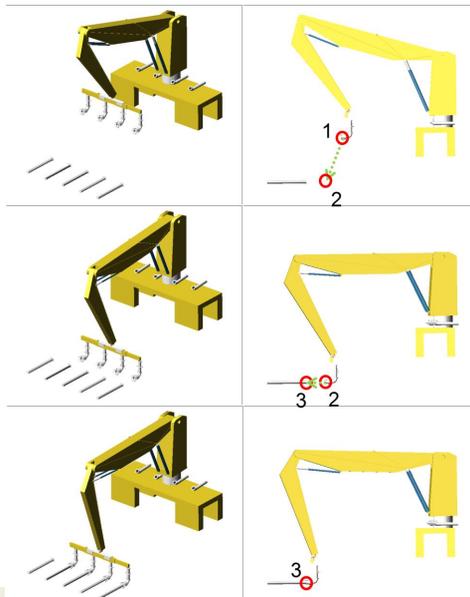
Dynamic control layout



Matlab model



End-effector motion control

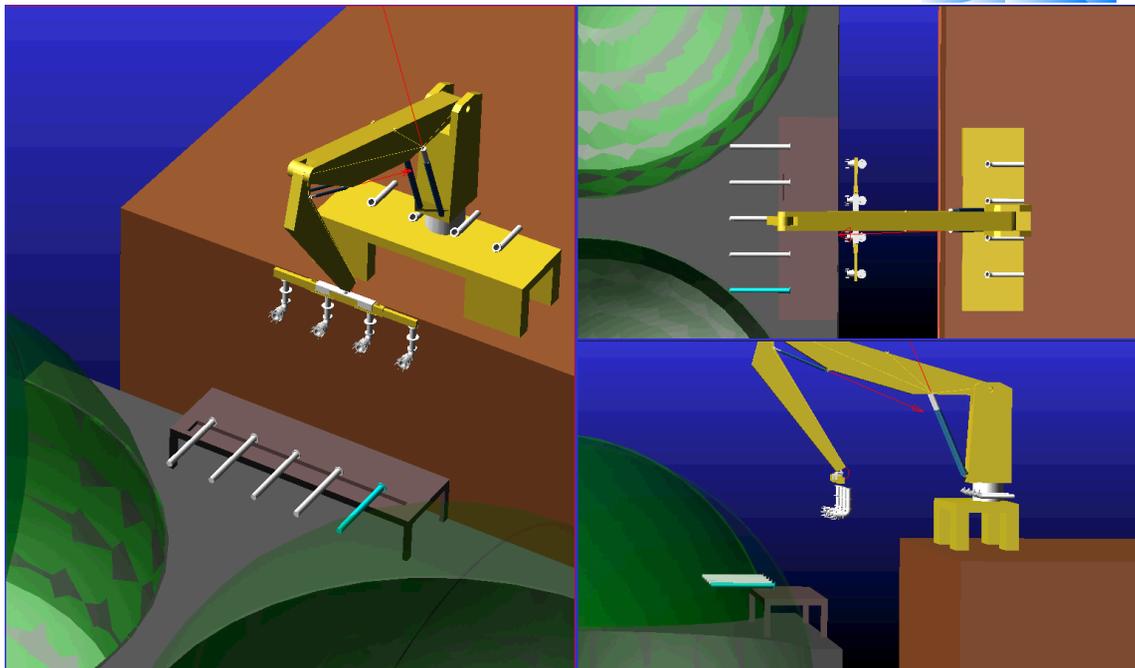
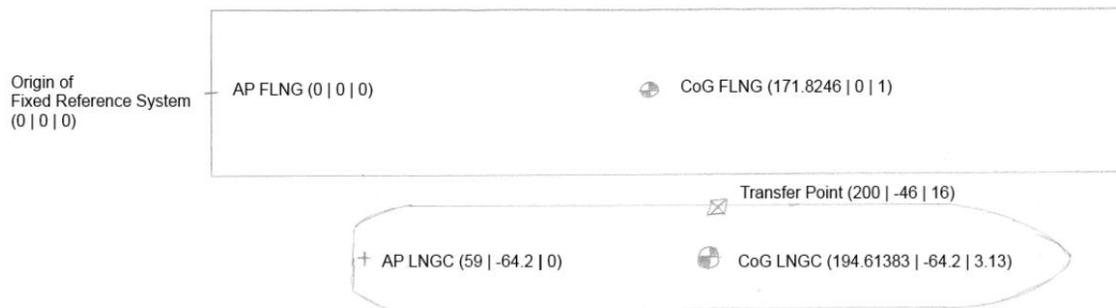


motion input table						
	x	y	z	R _x	R _y	R _z
1	-4.5	13	-10	-90°	0	-90°
2	-6	15	-15.7	-90°	0	-90°
3	-6	17.5	-15.7	-90°	0	-90°

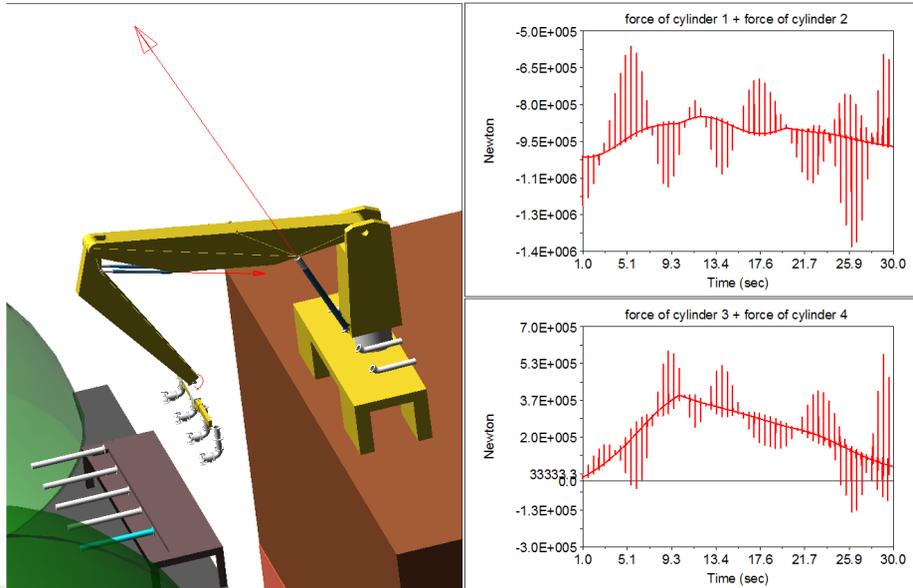
Measurement of real motion behavior



- Trajectory of FLNG, LNGC and coupling points
- Trajectory data has been used as simulation inputs to FLNG and LNGC models in Adams/Matlab co-simulation.



Cylinder forces



Questions?

DFKI Bremen & Universität Bremen
 Robotics Innovations Center
 Director: Prof. Dr. Frank Kirchner
www.dfki.de/robotics
robotics@dfki.de



3.4 ‘From CAD to a working simulation model’ (SM-P-01)

Jan Paul⁽¹⁾, Bertold Bongardt⁽¹⁾

(1) DFKI GmbH, Robotics Innovation Center, Robert-Hooke-Straße 1, 28359 Bremen, Germany

Contact: jan.paul@dfki.de, bertold.bongardt@dfki.de

Abstract

CAD2SIM is a tool that can export a MARS simulation model for a Robot from a SolidWorks CAD construction. The advantage is that with some additional settings a simulation model can directly be exported from the original construction model instead of creating a new simulation model from scratch, trying to match the CAD original as close as possible. This also enables updating the simulation from an updated CAD-Model, so that changes in the CAD Model of the robot can quickly be tested in simulation. The simulation models generated are not only restricted to MARS but also URDF, Matlab, OpenRAVE and RBDL models are exported. The URDF-Models can later be adapted using Phobos, a plugin for Blender.



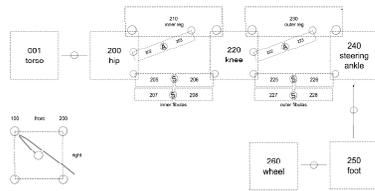
CAD2SIM

From CAD to a working simulation model

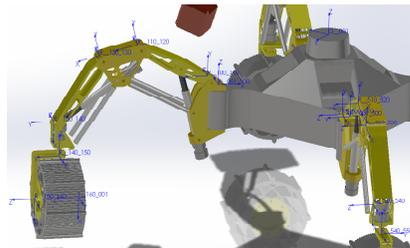
CAD2SIM creates simulation models from a CAD model

Necessary prerequisites:

- A CAD-Model in SolidWorks
- Mass information based on densities set for each moving part
- Movement axes as additional coordinate systems in CAD
- A topology definition containing systematic naming conventions
- A configuration file that associates SolidWorks groups to the limbs from the topology, determining the groups which will be used as graphical representations in the simulation



Topology information and naming convention for the Sherpa-TTA Rover

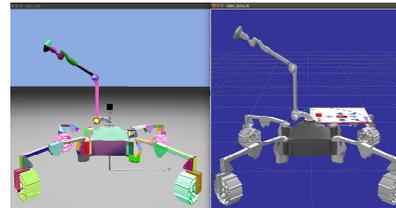


Coordinate systems for movement axes, here rotational axes for Sherpa-TTA

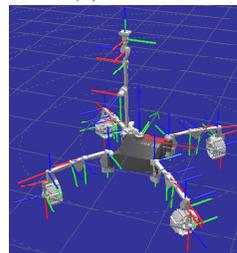
Output

- Ready to use simulation models in these formats:
 - MARS (classic scene format)
 - URDF (basis for the new mars scene format SMURF and usable in Vizkit3D and Gazebo)
 - Matlab
 - OpenRAVE
 - RBDL

The models can later be adapted, for example in the new MARS helper software "Phobos" when URDF is used or as currently done in TransTerra by ruby-scripts that adapt the MARS and URDF scene files to specific needs. This enables quick changes like for example "twice the maximum torque on all motors" etc.



MARS (classic file format) and Vizkit3D (URDF) simulation models working together in rock. Vizkit3D displays sensor readouts of the simulated robot.



Coordinate systems (=joint axes) transferred from CAD to simulation



Gui of the Blender based MARS tool "Phobos"

"Phobos" can edit CAD2SIM URDF models and also create simulation models standalone. It can add or change simulation model attributes like joint constraints, motors, collision objects or object masses.



Contact:
DFKI Bremen & University of Bremen
Robotics Innovation Center
Director: Prof. Dr. Frank Kirchner
E-mail: robotics@dfki.de
Website: www.dfk.de/robotics

3.5 ‘Procedural content generation’ (SM-P-02)

Jan Paul⁽¹⁾

(1) DFKI GmbH, Robotics Innovation Center, Robert-Hooke-Straße 1, 28359 Bremen, Germany

Contact: `jan.paul@dfki.de`

Abstract

Creating large artificial environments for testing robots in simulation can take a lot of time when done completely manually. Procedural content generation however can automate such a task when it is not the goal to match a real existing environment exactly but to have an artificial environment that has a natural structure. Such environments can be parameterized, so that they can easily be adapted to certain needs just by changing a few parameters. It is also possible to base such a content generation on base data from a real environment like a map of a city or an elevation map of a moon crater and then procedurally add details that are not present in the original data, like details on buildings or little irregularities on the surface of a crater.



Procedural content generation

Quickly create large parametric environments

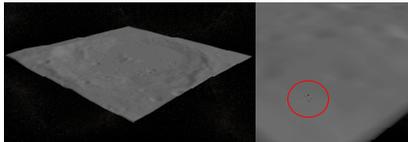
Why procedural content generation?

- Saves a lot of disk space, as large content can be created from a small set of parameters
- Saves time and effort compared to creating large environments by hand
- Large environments are needed to test long missions in simulation
- Environment properties can quickly be changed based on parameters
- Existing rough terrain Data can be enhanced by procedurally added Details

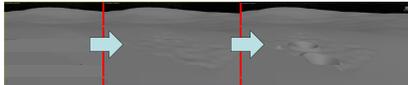
Example 1: Amundsen crater

The Amundsen moon crater is needed as a simulation environment in TransTerra:

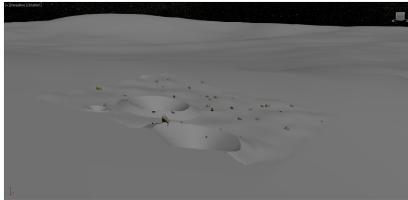
- 26km Radius
- Rough height map exists but insufficient for realistic simulation environment
- ⇒ Focus on freely choosable 1km² sections that are extended with procedural details in several steps for the simulation



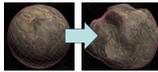
Complete crater based on real measured height map and 1km² section



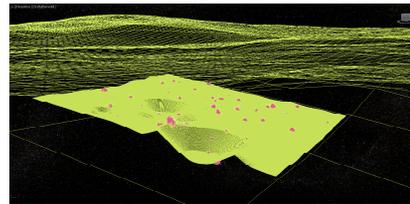
Procedurally adding surface details and additional craters



Result with added details, craters and surface-aligned rocks



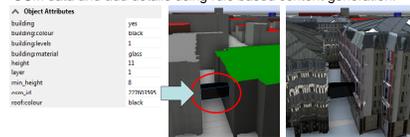
Creating rocks out of spheres by adding procedural details



Increased detail level in selected section

Example 2: Bremen in 3D based on Open Street Map data

Needed for simulations and visualizations in electro mobility projects. Rather than modeling a city manually or scanning the environment in 3D, use as much 3D information as possible from OSM data and add details using rule based content generation.



Building part (black glass passage) derived from OSM attributes. Increased quality based on more detailed rule set.



Virtual part of Bremen in CAVE simulation



Contact:
DFKI Bremen & University of Bremen
Robotics Innovation Center
Director: Prof. Dr. Frank Kirchner
E-mail: robotics@dfki.de
Website: www.dfki.de/robotics

3.6 ‘SMURF: A new, easily extendable robot model format based on URDF’ (SM-P-03)

Kai von Szadkowski⁽²⁾, Malte Langosz⁽¹⁾

(1) DFKI GmbH, Robotics Innovation Center, Robert-Hooke-Straße 1, 28359 Bremen, Germany

(2) Universität Bremen, Arbeitsgruppe Robotik, Robert-Hooke-Straße 1, 28359 Bremen, Germany

Contact: `kai.von-szadkowski@uni-bremen.de`, `malte.langosz@dfki.de`

Abstract

SMURF stands for Supplementable, Mostly Universal Robot Format and is an attempt to overcome the limitations of URDF (and SRDF) to create a more versatile format that can be easily adapted to the needs of different groups of users and projects, while at the same time maintaining full compatibility to URDF. Instead of trying to define an extensive and complex format with a limited set of elements, SMURF defines how to structure a hierarchical tree of YAML files and provides a parser to generically and recursively read in all information. This allows anyone to freely extend the SMURF models of their robots beyond the core set of data elements pre-defined by the Rock robotics framework and the MARS simulation. An open-source parser for SMURF is hosted on GitHub: https://github.com/rock-simulation/phobos/smurf_parser



SMURF

A new, easily extendable robot model format based on URDF

What is SMURF?

SMURF stands for **S**upplementable, **M**ostly **U**niversal **R**obot **F**ormat and is an attempt to overcome the limitations of URDF (and SRDF) to create a more versatile format that can be easily adapted to the needs of different groups of users and projects.

Structure

SMURF is not so much a format for a single file but rather defines a hierarchy of files that together provide information on different aspects of a robot. Thus a SMURF robot definition consists of one main `.smurf` file containing general data on the model and a list of files that should be included in the SMURF representation of the robot. At a minimum, a `.urdf` file that defines the robot's kinematics is listed here. Normally, the list will also contain other files - encoded in the YAML format - defining different parts of the robots such as sensors, motors or additional data, e.g. simulation parameters.

In these YAML files, information can be linked to specific parts of the robot by referencing them by type and name as specified in the URDF (combinations of type/name are unique in URDF). For instance, a sensor can be defined by attaching it to a link, then specifying its update rate and other parameters. Similarly, motors can be attached to joints, specifying maximum torques and turning speeds. This information does not have to be provided redundantly, as it is possible to define URIs of additional YAML files containing the data which is not explicitly listed in an entry. This makes it possible to maintain a central database of devices used multiple times in the same robot or even in different robots, without the need to update every single model independently if any single part changes and thus reducing repetitive workload and errors.

Extensibility

Apart from a core set of defined features such as the ones listed above, SMURF provides the user with the freedom of specifying any kind of information related to any part of the robot or the model as a whole, using a subset of the YAML syntax. The existing C++ SMURF parser recursively traverses the entire SMURF file tree and provides all the data of the model in an easy-to-work-with data format called a *ConfigMap*, which is a C++ implementation of a Python dictionary using templates. The parser is currently distributed as part of the MARS simulation on GitHub:

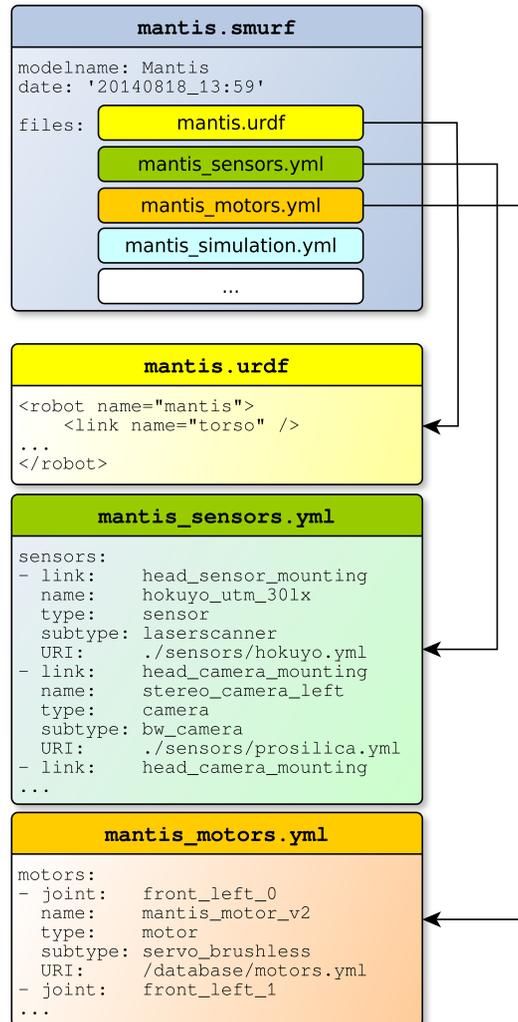
github.com/rock-simulation/mars

Compatibility & Editing

SMURF is compatible with the MARS simulation and will be compatible with future versions of the Rock robotics framework. URDFs contained in SMURF do not have any custom tags and are thus compatible with the ROS robotics framework and the Gazebo simulation.

Phobos is a plugin for the free 3D modelling software Blender, turning the latter into a powerful tool for creating and editing SMURF robot models. It is itself open source software, distributed under the LPGL license, and is hosted on GitHub:

github.com/rock-simulation/phobos



Example of a SMURF robot model – here a modified excerpt of the representation of the DFKI's Mantis robot.

3.7 ‘Phobos: A Blender Plugin for Creating Robot Simulation Models’ (SM-P-04)

Kai von Szadkowski⁽²⁾, *Malte Langosz*⁽¹⁾

(1) DFKI GmbH, Robotics Innovation Center, Robert-Hooke-Straße 1, 28359 Bremen, Germany

(2) Universität Bremen, Arbeitsgruppe Robotik, Robert-Hooke-Straße 1, 28359 Bremen, Germany

Contact: `kai.von-szadkowski@uni-bremen.de`, `malte.langosz@dfki.de`

Abstract

Phobos is an open-source plugin for Blender, making use of its powerful 3D modeling functionality to simplify editing URDF and SMURF robot models. By fully exploiting Blender’s existing data structures and scene objects, and extending them where necessary, Phobos simplifies the workflow for building even complex robot representations. Phobos is hosted on GitHub: <https://github.com/rock-simulation/phobos>



Phobos

A Blender Plugin for Creating Robot Simulation Models

What is Phobos?

Apart from one of the moons of Mars, Phobos is the name of a plugin for the open-source 3D modeling software Blender. It uses Blender's powerful Python scripting API to extend Blender with GUI-supported tools that simplify the creation and modification of robot models for use in real-time simulations such as MARS or Gazebo. It is both possible to create robots from scratch as well as to derive them from exported CAD data - and Phobos allows to export completed models in formats such as URDF (widely used in ROS) and SMURF (the DFKI's versatile robot data format based on URDF). Meshes can currently be exported as .stl, Wavefront (.obj) and binary Wavefront (.bob) for quicker importing into the MARS Simulation.

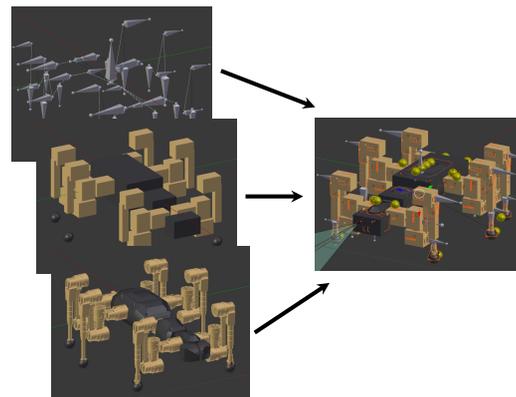


Blender's user interface with the Phobos plugin running. Phobos' custom toolbars are shown alongside the SpaceClimber robot.

Robot representation

Phobos makes use of Blender's hierarchical object graph and its *bone* objects. These objects, normally used for animating 3D characters, allow to store 3D coordinate systems and apply constraints to their movements, for instance to restrict the movement of an object to a certain range on a specific axis. This allows to replicate the *links* and *joints* defined in a URDF model and together with the hierarchical tree of parent and child objects, the complete, branching kinematic chain of a robot can be represented.

By attaching meshes or primitives to the bones, Phobos allows to add *visual* and *collision* objects to a model. Additional objects allow storing further information, e.g. centers of mass of each part of a robot, thus refining the physical representation. Sensor objects can be added to correctly place and orient devices such as laser scanners, cameras or contact sensors. Making use of Blender's custom object properties, any necessary information can be added to the model, from inertia tensors to opening angles of cameras.



A complete robot model created with Phobos consists of different types of objects which can be organized on different layers in Blender. Some objects, such as collision shapes, can be auto-generated.

Features

- Model error checking
- Batch editing of object properties
- Measurements (e.g. distance of parts, sum of masses)
- Synchronization of mass information between different layers
- Auto-generation of collision objects
- Auto-generation of simplified inertia; combination of manually defined precise inertia
- Online definition and testing of joint constraints
- Sensor attachment to single or multiple parts of a robot
- Import and export of URDF, SMURF and other formats
- Exporting with defined floating point precision
- Growing number of tutorials

Getting Phobos

Phobos is distributed under the LGPL and freely available on GitHub: github.com/rock-simulation/phobos
Its first release version is compatible with Blender 2.69 and later.

Getting Blender

Blender is a free 3D modeling software and can be obtained from www.blender.org
The site also provides a wealth of tutorials on working with Blender.



Contact:
DFKI Bremen & University of Bremen
Robotics Innovation Center
Director: Prof. Dr. Frank Kirchner
E-mail: robotics@dfki.de
Website: www.dfk.de/robotics

German Research Center for Artificial Intelligence (DFKI) GmbH

DFKI Bremen

Robert-Hooke-Straße 1
28359 Bremen
Germany
Phone: +49 421 178 45 0
Fax: +49 421 178 45 4150

DFKI Saarbrücken

Stuhlsatzenhausweg 3
Campus D3 2
66123 Saarbrücken
Germany
Phone: +49 681 875 75 0
Fax: +49 681 857 75 5341

DFKI Kaiserslautern

Trippstadter Straße 122
67608 Kaiserslautern
Germany
Phone: +49 631 205 75 0
Fax: +49 631 205 75 5030

DFKI Projektbüro Berlin

Alt-Moabit 91c
10559 Berlin
Germany
Phone: +49 30 238 95 0

E-mail:

reports@dfki.de

Further information:

<http://www.dfki.de>