THE DLR ROVER SIMULATION TOOLKIT

Matthias Hellerer [(1)], Stefan Barthelmes [(1)], Fabian Buse [(1)]

[(1)] German Aerospace Center (DLR), Institute of System dynamics and Control, Münchener Straße 20
DE-82234 Weßling, Email:[Matthias.Hellerer|Stefan.Barthelmes|Fabian.Buse]@DLR.de

ABSTRACT

Autonomous exploration rovers are currently the primary means of research on extra-terrestrial bodies. Due to the circumstances of their deployment it is vital to ensure their unassisted performance in a harsh environment. Preliminary simulations are therefore indispensable. To facilitate these simulations we introduce the DLR Rover Simulation Toolkit (RST). The RST constitutes a framework of libraries, allowing engineers to quickly assemble digital rover twins, particularly for early design phases. Enabled by the modelling language Modelica it covers a wide range of aspects from different domains in one unifying framework.

This paper establishes in detail the RST's structure and design decisions before showing its practical application in a Software-in-the loop (SiL) simulator, elaborating on future enhancements and its use during collaborative engineering studies in the DLR Systems and Control Innovation Lab.

1. INTRODUCTION

Space missions are characterized by extremely high costs and very limited opportunities. For most projects this means that actual missions are few and far between. Therefore it is of utmost importance to assure that every part of the project is working correctly and that all eventualities are covered as most space hardware will only get a single chance to be deployed.

Testing hardware in advance under real or realistic circumstances is usually either very expensive or even impossible on earth. Furthermore, even if prototypes are available, only very few are built which results in strictly controlled access to those. To circumvent both of these problems, simulations form an integral part of the development process of every complex system and especially for space missions.

However, creating good simulations is a complicated and time consuming task by itself. To facilitate this in the case of planetary rovers, the Rover Simulation Toolkit (RST), which was first publicly introduced at ASTRA 2015 [1], is developed at the DLR Institute of System Dynamics and Control.

Being the most complicated part of simulating a mobile robot on rough terrain, the wheel-soil interaction is the focus of the majority of work being done in that area (cf. [2, 3]). Existing full rover simulations mostly focus on certain aspects. For the ESA ExoMars mission, a tool for combining quasi-static equations with wheel-soil test data was developed to evaluate different kinematic concepts and dimensions [4]. In [5], the authors used a dynamic multi-body system purely for evaluations of their controller. The most comprising tool to date seems to be ARTEMIS [6] which is an Adams-based mobility evaluation tool that was used for assessment of mobility issues and candidate drive paths for the Mars Exploration Rovers (MER) and the Mars Science Laboratory (MSL) at NASA's Jet Propulsion Laboratory (JPL).

All mentioned tools have very special applications and even the rather detailed multi-body model within ARTEMIS is tailored to simulating the mobility performance of the MER and MSL rovers.

The goals of the RST however are more manifold:

- Support and enrich early development stages and concurrent engineering studies.
- Make tests repeatable thereby making test results and designs comparable.
- Tests in environments which cannot easily be recreated in reality.
- Improve rover design by multi-critiria optimization of parameters or even the rover structure.
- Generate interactive mission demonstrators to visualize key objectives to management or the public.
- Assess the mobility performance over rough terrain – including sands and rocks – and thereby support path planning and/or landing site selection.
- Software-in-the-Loop (SIL) simulations to develop and evaluate chassis controllers as well as path planning and full autonomy and simultaneous localization and mapping (SLAM) algorithms.
- Bundle our knowledge about rovers in an easily accessible library.

To be able to fulfil all these requirements with a high result quality, a modular design with submodules of very different levels of detail is key. The object oriented modelling language Modelica allows this modular design in all relevant physical domains which seamlessly interact with each other. Base objects are defined by equations and interfaces, which are then assembled into larger, more complex objects, which again might be part of an even larger component. Using Modelica also allows us to utilize a large number of custom and commercially available libraries [7].

2. THE DLR ROVER SIMULATION TOOLKIT

2.1. Overview

The RST is designed to make the adaptation or creation of rover simulations as fast and easy as possible. To do so it emphasises clear structures and reusability of components. It prefers composition over inheritance and avoids complex structures such as replaceable packages. Nonetheless it defines a structure for developers to follow, defines clear interfaces and demonstrates how individual components are designed to be reusable [8].

The structure of a rover in the RST reflects both the practical view of its most important components and its physical structure, thereby making it intuitively understandable and usable. Additionally this structure allows the straight-forward replacement of certain parts with hardware testbeds for Hardware-in-the-loop (HIL) simulators. At the highest level a complete Rover is placed in an environment. On this layer only the most relevant configuration parameters, a user might typically adjust when simulating a specific rover. Examples for such parameters include the starting position and the loaded control program.

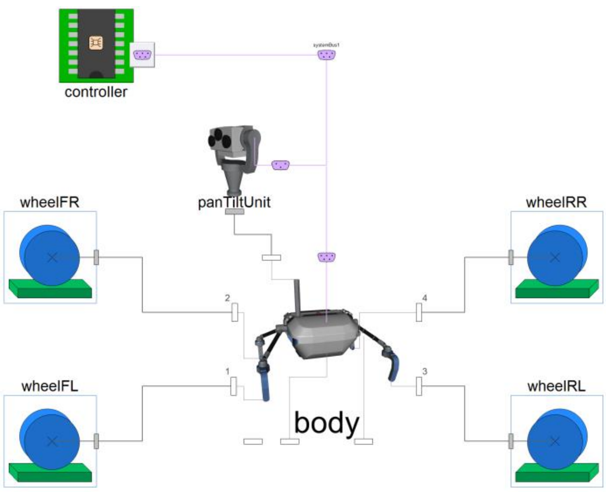

Figure 1. The LRU Rover in the RST at the fist level of detail.

One level down the rover is structured as reflected and elaborated on in the following subsections and shown in Fig 1. Depicted is the Lightweight Rover Unit (LRU), a small and agile rover prototype for planetary exploration, developed at DLR. The central component is a multi-body model of the rover's physical structure. The locomotion is such a critical point and because this is a central aspect of our research, the wheels and with them the configuration of the wheel ground contact, the wheels are separated from the rest of the model at this level. These wheel components allow adjustment of wheel parameters, contact detection as well as the contact force model itself. Also at this level is the rover's controller. The controller represents all the control algorithms required to perform a given task, not the physical components the algorithms are run on. The last component(s) at this level is(are) the rover's payload(s). In Figure 1 this is a pan-tilt camera unit.

2.2. Multi-Body System

The multi-body system is the core of the rover. It encompasses all its physical components and their connections. Typically a rover has a central body around which a kinematic structure connects it to the wheels. The most commonly used components for this connection, such as rockers, bogies, steering servos and drive motors, are part of the RST.
Drives are also part of the multi-body system. For they are modelled as general as possible and to allow maximum reusability, they have been split from the RST into an independent library called "Servos". While it is a separate library, it still developed as part of the RST. Servos are built modularly and are composed of individually replaceable motors, gears, sensors, drivers and some auxiliary components.

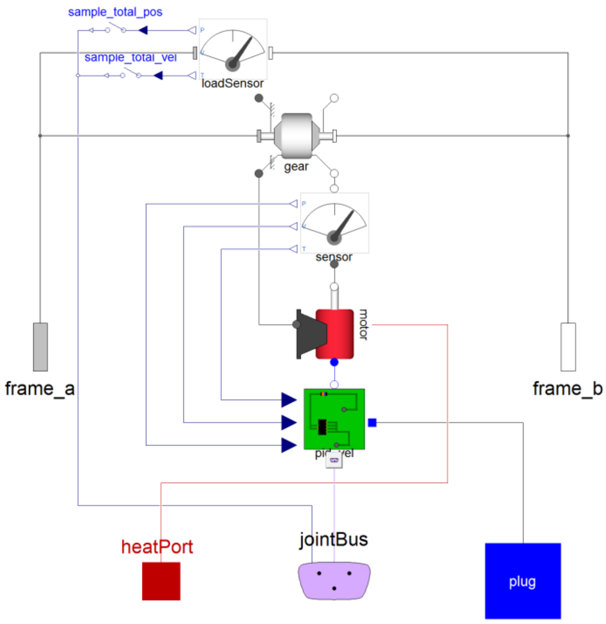

Figure 2. The structure of a servo

The structure of a servo is depicted in Fig 2. The servo receives commands for target positions from a bus connection. The command is relayed to the servo driver which contains the low level control algorithms and controls the motor based on read sensor data. The motor drives a gearbox which then turns the multi-body frames connecting the servo to its surroundings. Standard components and a number of complete servo models are part of the Servos library.

2.3. Wheel-Ground interaction

The interaction between rover wheels and the ground is the most important aspect for the rover locomotion. In our research this also takes a central role. Just as the Servos library, this part has also been converted into a separate, reusable library, named “ContactDynamics”.

The ContactDynamics library encompasses three major aspects: Contact detection, ground description and contact reaction. All of them are designed to be replaceable and extensible, they can be combined freely and the library contains a selection of implemented instances.

Different contact detection methods are implemented for both, wheel-soil contact and wheel-object contact. The former is based on contact search in certain directions. For a contact search on relatively smooth terrain, only one search direction from the wheel hub in vertical direction leads to decent results. For non-smooth terrain however, this yields problems as depicted in Figure 3: A step in the soil is detected too late causing unrealistic behaviour. With various contact search directions distributed equally over the wheel rim and a suitable averaging between them, non-smooth terrain can be handled. For wheel-object contact, a full contact search algorithm for convex bodies is available as well.

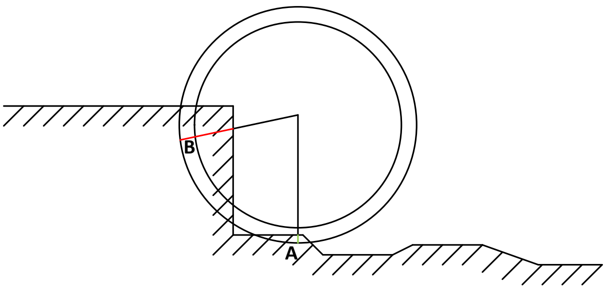

Figure 3. Simple contact detection for one search direction. A shows a detected penetration of the ground, B shows an undetected penetration.

Based on a contact detected by the previous step, a reaction force can be calculated. Here the ContactDynamics library is again designed modularly and allows the user to select from a range of algorithms for calculating the forces and torques that are associated with the detected contact. The primary differentiation is a trade-off between accuracy and run-time. An overview of the available models and their properties is given in Figure 4.

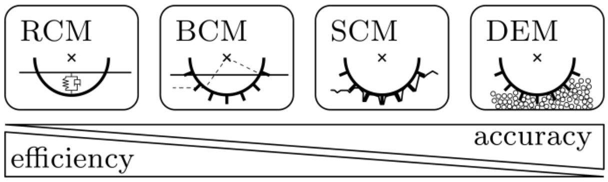

Figure 4. Contact reaction algorithms differentiation by efficiency and accuracy

The simplest and fastest algorithm is based on a rheological contact model (RCM). The force in normal direction is calculated with a spring-damper model based on the penetration of the wheel in the ground and the tangential force, representing friction, is based on the speed of the wheel rim relative to the ground. More complex models are based upon empirical Terramechanics models, e.g. Bekker Wong and Janosi Hanamoto (BCM) [9]. To cover sand deformation and its effects on the forces, SCM features a 2.5-dimensional spatial discretization. For three-dimensional physical modelling, a discrete elements model (DEM) has been developed. A more detailed discussion of the available algorithms can be found in [2].

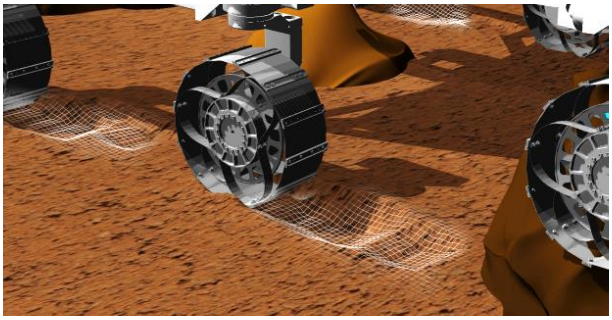

Figure 5. Wheel ground contact model with soil compression and deplacement

2.4. Control algorithms

As discussed earlier, low level controllers are implemented like their real counterparts as part of the drive units. Higher level algorithms, controlling multiple drives simultaneously and thereby defining the robot’s behaviour are implemented in a specialized controller module. The controller module is then connected to the drives and sensors via a system bus.

Higher level control structures are implemented using a hierarchical state machine. The user can first define very general operation states and refine those in sub states. This is implemented using Modelica’s state machines with some minor improvements.

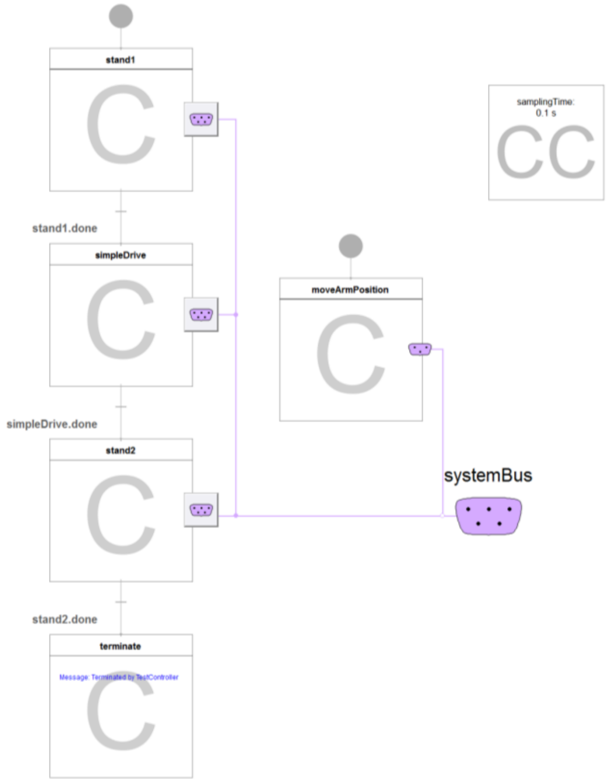

Figure 6. A simple state machine example

A simple example for a state machine in the RST is depicted in Fig 6. Shown on the left side is a very simple sequence. When the simulation starts, the rover stands still for a given time, then it drives a few meters in a straight line, before stopping again and then terminating the simulation. To the right of the sequence, executed in parallel is a sequence composed of only one step, keeping an arm attached to the robot fixed in its position. In the upper right corner a block for advanced configurations, such as the sampling time is shown.

Similar to a real rover, the controller and system bus work at a certain frequency by using Modelica's synchronous control feature.

For each rover the RST implements a number of basic commands which allow the user to easily assemble complex tasks. Typically this includes commands such as driving in crab drive (all wheels parallel) for a given time, driving a certain distance using Ackermann steering or moving an attached robotic arm to a specific location.

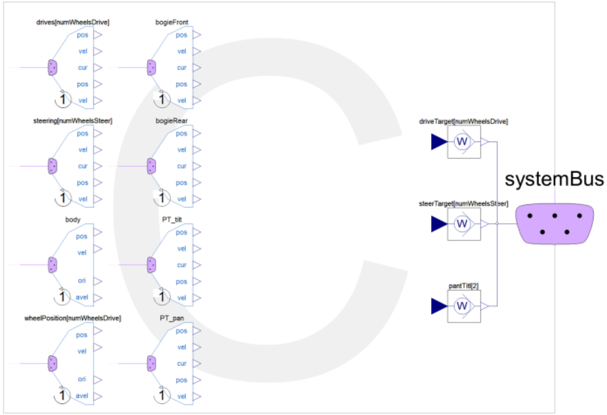

Figure 7. The base command for a rover in the RST

Each command is structured as shown in Fig 7, with a number of read-out blocks on the left, providing details about the rover's current state and a number of output blocks on the right to set servo target values. Commands can either be programmed using Modelica language or contain co-simulation interfaces to controllers e.g. in Simulink.

3. STRUCTURE

Section 2 and Fig 1 already introduce the model structure of a rover in the RST, strongly modelled after the rover's physical structure. Besides that the rover elements are also organized logically in the Modelica library as depicted in Fig 8.

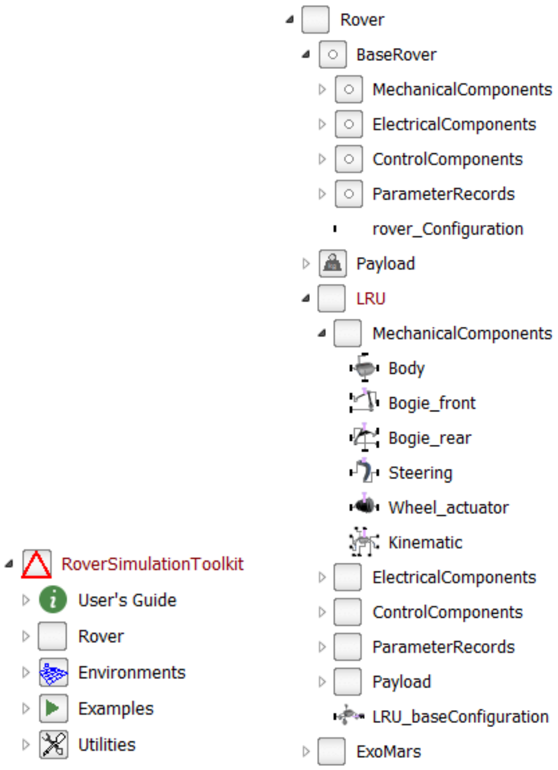

Figure 8. The logical structure of the RST

- User's Guide
  The RST includes an extensive documentation
- Rover
  The rover models
  - Base Rover
    The basis for all rovers. Models the structure and includes a number of basic components.
  - Payload
    Generic payload components. All components which are not an integral part of the main rover, are considered payload. For example an attached robotic arm.
  - LRU
    The LRU rover as an example rover model in the RST. The rover follows the base rover structure and uses it to implement the components as shown here for the mechanical components.
    Some rovers also feature specific payloads, only applicable to this rover. In the case of the LRU this is the pan-tilt unit depicted in Fig 1. Those elements are also considered part of the rover.
  - Further rovers
    The RST contains a number of further rover models
- Environments
  A collection of basic terrains and props such as a lunar landscape and a lander module.
- Examples
  A collection of fully modeled examples so users can see the library in action and start modifications from a fully functional simulation.
- Utilities
  A collection of utilities, mainly for internal use.

This structure is very strongly recommended for all rover models because a common structure makes orientation within a model very easy to learn, yet, to allow for maximum flexibility, it is not enforced.

As alluded to in section 2, we tried to carve out all generic components into their own libraries. These libraries are designed to be as generic as possible and may be used by other projects but they originate in the RST and are maintained under the RST umbrella.

- Servos
  See section 2.2
  A library for servo motors including a bus interface, controllers, drivers, gears and motors.
- ContacDynamics
  See section 2.3
  One of the most important aspects of locomotion simulation is the contact between the wheel and the ground. For loose and sandy surfaces this is especially relevant and complicated to simulate.
- Commands
  See section 2.4
  The Commands Library is small convenience Library, adding some features for command sequences (e.g. a simple to configure sample time) to the Modelica_Synchronous state machines.

Of course not all libraries the RST depends on are actually part of the RST. Besides the ones listed above it also uses a number of Modelica standard libraries and the DLR Visualization Library [10].

4. APPLICATIONS

The RST is already used for various projects within DLR.

4.1. MOREX

In the five-year DLR project Modular Robotic Exploration (MOREX) the goal is to raise the competence of the DLR in robotic exploration. The four DLR institutes Robotics and Mechatronics, System Dynamics and Control, Communications and Navigation and Optical Sensor Systems are involved and the findings are to be combined into one pilot project rover towards the end of the project runtime. Currently, the RST is being used to support the early development phase with parameter studies and optimizations for dimensioning and analysis of the rover kinematics and dynamics. The simulation model is subject to ongoing development to deliver continued support throughout all stages of the rover design. In return, experience in conducting an actual rover design process and accompanying it with the simulation will bring extensive insights and lead to continuous improvement of the RST.

4.2. Software-in-the-loop Simulator

As described earlier, access to robot hardware is often limited and not all environments can be provided in reality. These problems can be mitigated by employing Software-in-the-loop (SiL) simulators. A SiL simulation replicates all the interfaces between a rover's hardware and its software in such way that the software may run on both the real and the simulated system without modifications [2]. Currently, such a SiL simulation is used for all levels of rover controllers from autonomy development at the DLR Institute of Robotics and Mechatronics to chassis controllers at the Institute of System Dynamics and Control.

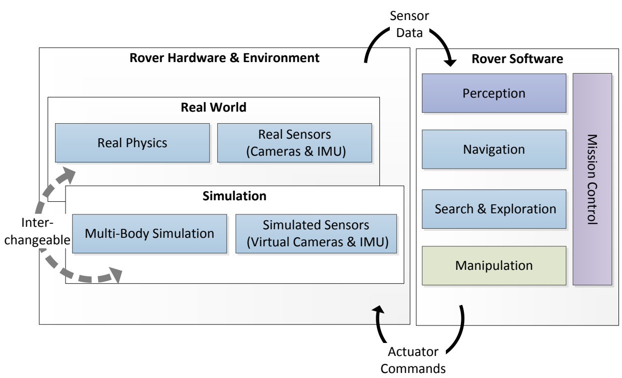

Figure 9. A Software-in-the-loop simulation

4.3. Simulated Stereo-Cameras for optical navigation

Utilizing the RST a SiL simulation of the LRU which transparently replaces the real rover as depicted in Fig 9 was built. Most of the functionality for such a simulator was already present but for this application the RST had to be augmented with simulated cameras. This was necessary because the LRU software is primarily based on optical navigation which is also one of the components which is developed using the simulation. Here our 3D visualization was utilized [10] and an off-screen rendering of images added, which was then connected to our middleware similar to a real camera.

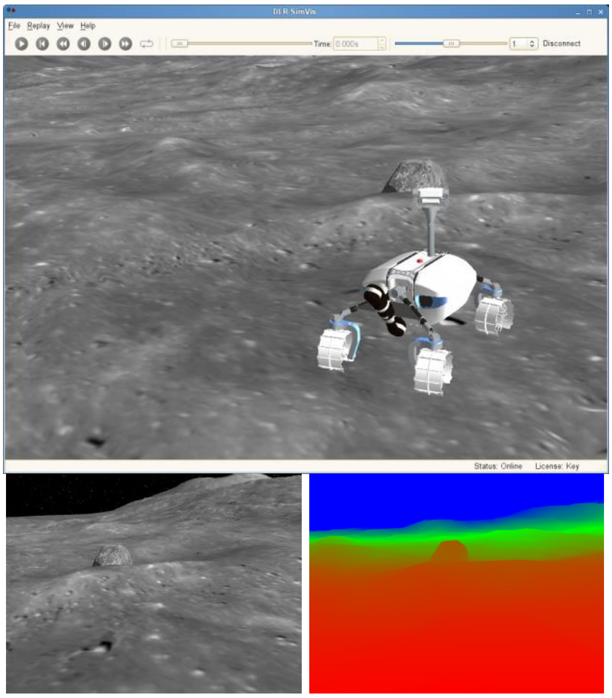

Figure 10. Virtual cameras simulating real cameras; first: Simulation view, second: virtual camera, third: virtual depth camera.

Fig 10 shows the generation of virtual camera images. The first image shows a 3D visualization of the simulation in our viewer application. The second image shows a virtual camera image of the same scene. In the first image the boulder is in front of the rover, in the second image it is seen from the rover's perspective. Finally the third image shows a virtual depth image. The colour denotes the distance from the camera, its gradient indicating the receding ground with the boulder sticking out. 3D reconstruction from stereo images is a very important aspect of image perception but this is also computationally extremely expensive. On the real rover this is done using a specialized FPGA. For the developers those FPGAs are very limited and therefore we decided to include a 3D depth image generated from the OpenGL depth buffer of the virtual camera. This is again connected to our middleware the same way the FPGA processed image is. Thereby other parts of the rover can be worked on even when no FPGA is available.

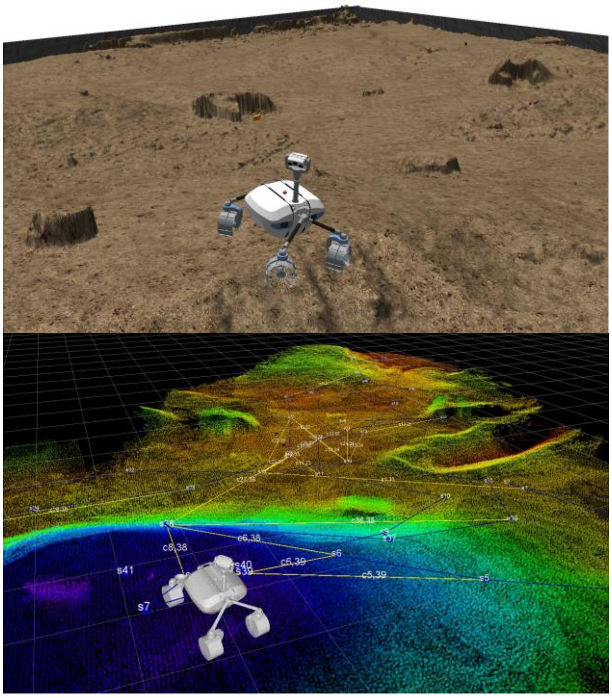

Figure 11. Top: The rover in a virtual environment; Bottom: the reconstruction of the environment created by the navigation software

This SiL simulator was originally developed in preparation for the 2015 DLR SpaceBotCamp, a German national robotics tournament. The rover had to autonomously navigate and explore a large area with rough terrain, localize and pick up two objects and return those to a base station [11]. Amongst ten competitors the LRU was the only rover to complete all mandatory tasks and did so in half the given time. This great success was in part made possible by a simulator which allowed the rover software developers to test their systems extensively beforehand. Fig 11 shows the SiL simulator as used in preparation for the SpaceBotCamp [1, 12].

The Helmholtz Alliance "Robotic Exploration of Extreme Environments" (ROBEX) is a cooperation of sixteen institutes in the field of space and underwater research in extreme environments [13]. The idea is to find common technology between those fields and to share and benefit from an exchange of knowledge. For the final presentation of this project in the summer of 2017, the LRU will present a lunar analogue mission. The presentation will demonstrate the rover's ability to autonomously set up an active seismic network (ASN) in a moon like environment [14]. Mt. Etna on Sicily, Italy was selected as the site for the analogue mission, as it features a moon-like surface and has constant seismic activity. The simulator was again used extensivly in preparation of this event.

5. FUTURE DEVELOPMENTS

While the development of the RST has come a long way, we still have many new plans to extend it. The next step, after the ROBEX Demo mission on Mt. Etna is finished, will be to use the data gathered during the mission to verify the simulations and to refine them where necessary.

Furthermore we are currently working on a system to automatically analyse the kinematic structure of a model. This should allow us in the future to write generic controllers which can adapt to any rover kinematic. For example an Ackerman steering could be given a centre point of a curve and the controller would analyse the relative position of each wheel and calculate the appropriate steering angle. A generic controller could do so without being configured for a specific rover kinematic.

Generic controllers are of crucial importance for our next improvement: Optimization of whole rovers, including their kinematic structure and the controllers. We want the optimization to not only optimize single aspects of a rover, like different bogie lengths. We want the optimization to change the kinematic structure, e.g. to change the number of wheels or the way those are attached to the main body. Of course comparisons between the performances of different rovers in certain environments can only be meaningful if both the compared rovers have optimized controllers. So at the same time as changing the rover kinematic, a new controller has to be generated automatically and then optimized before a new design can be evaluated.

Besides this, the RST has been designed for early design studies. While it has already been successfully used in the evaluation of early designs, we will also apply it in the context of future missions. Such studies can take place in the Systems and Control Innovation Lab, a newly founded Helmholtz Innovation Lab with the aim to integrate commercial and public research and foster technology transfer in the fields of simulation, optimization and control methods. In the new Lab, infrastructure such as a VR Laboratory (depicted in Fig 13), gives engineers a collaborative work environment which enables them to combine single system simulations with the Rover Simulation Toolkit using the Functional Mockup Interface Technology or interfaces to our terramechanics and mechatronic testbeds.

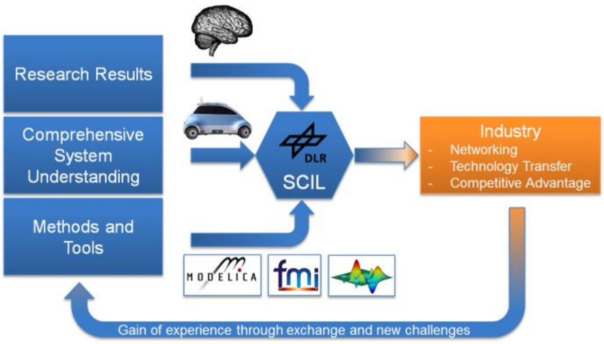

Figure 12. Schematic of knowledge flow in the Systems and Control Innovation Lab

Figure 13. A cooperative design session for an aircraft controller at the SCIL

6. CONCLUSION

The DLR Rover Simulation Toolkit was presented as a framework for the design and simulation of planetary rovers. It is built from the ground up to be modular and to encourage code reuse which enables users to quickly create or modify rovers. Many parts were separated from the library but are still maintained under its framework and may be used for other purposes, further improving its reusability. The structure of the framework was introduced in detail and explained how this supports the development. Users may easily create and evaluate new rover designs or modifications especially in early design phases such as CEF or phase A-B studies, yet the frameworks ability to switch between different levels of detail also allows for in depth analyses. Furthermore the framework's performance in applications was shown with examples such as a Software-in-the-loop simulator and finally future plans for new features were discussed.

7. REFERENCES

1. Wedler, Armin and Hellerer, Mathias and Rebele, Bernhard and Gmeiner, Heiner and Vodermayer, Bernhard and Bellmann, Tobias and Barthelmes, Stefan and Rosta, Roland and Lange, Caroline and Witte, Lars and Schmitz, Nicole and Knapmeyer, Martin and Czeluschke, Alexandra and Thomsen, Laurenz and Waldmann, Christoph and Flögel, Sascha and Wilde, Martina and Takei, Yuto (2015) ROBEX – COMPONENTS AND METHODS FOR THE PLANETARY EXPLORATION DEMONSTRATION MISSION. In: 13th Symposium on Advanced Space Technologies in Robotics and Automation (ASTRA). ESAWebsite. 13th Symposium on Advanced Space Technologies in Robotics and Automation (ASTRA), 10-13 May 2015, Noordwijk, The Netherlands.

2. Lichtenheldt, Roy and Barthelmes, Stefan and Buse, Fabian and Hellerer, Matthias (2016) Wheel-Ground Modeling in Planetary Exploration: From Unified Simulation Frameworks Towards Heterogeneous, Multi-tier Wheel Ground Contact Simulation. Multibody Dynamics - Computational Methods and Application Computational Methods in Applied Sciences, 42. Springer. pp. 165-192. ISBN 978-3-319-30612-4

3. Ding, L.; Deng, Z.; Gao, H.; Nagatani, K. & Yoshida, K. (2011), Planetary rovers' wheel—soil interaction mechanics: new challenges and applications for wheeled mobile robots, Intelligent Service Robotics 4(1), pp. 17-38.
4. Michaud, S.; Krpoun, R.; Wismer, S. & Gloster, D. (2011), Exomars locomotion subsystem analytical tool development and correlation, Published at ESA/ESTEC, the Netherlands.
5. Ishigami, G.; Miwa, A.; Nagatani, K. & Yoshida, K. (2007), Terramechanics-based model for steering maneuver of planetary exploration rovers on loose soil, Journal of Field robotics 24(3), pp. 233-250.
6. Zhou, F.; Arvidson, R. E.; Bennett, K.; Trease, B.; Lindemann, R.; Bellutta, P.; Iagnemma, K. & Senatore, C. (2014), Simulations of Mars rover traverses, Journal of Field Robotics 31(1), pp. 141-160.
7. Elmqvist, H. and Mattsson, S. E. and Otter, M. (1999) Modelica - A Language for Physical System Modeling, Visualization and Interaction. In: Proceedings of CACSD'99, pp. 630-639. IEEE International Symposium on Computer-Aided Control System Design, Hawaii, Aug. 22-27, USA, Hawaii. ISBN 0-7803-5449-4

8. Gamma, Erich and Helm, Richard and Johnson, Ralph and Vlissides, John (1995) Design patterns: elements of reusable object-oriented software. ISBN 0-201-63361-2

9. Wong, Jo Yung (2008) Theory of Ground Vehicles. John Wiley & Sons, Inc., Hoboken, New Jersey, USA, 4 edition

10. Hellerer, Matthias and Bellmann, Tobias and Schlegel, Florian (2014) The DLR Visualization Library - Recent development and applications. In: Proceedings of the 10th International Modelica Conference - Lund, Sweden - Mar 10-12, 2014, pp. 899-911. LiU Electronic Press. The 10th International Modelica Conference 2014, 10-12 March 2014, Lund, Sweden. DOI: 10.3384/ecp14096899 ISBN 978-91-7519-380-9

11. DLR SpaceBot Camp. https://web.archive.org/web/20160320051557/http://www.dlr.de/rd/en/desktopdefault.aspx/tabid-8101/13875_read-35268/ (2015). Accessed 2017-05-30

12. Schuster, Martin J. and Brand, Christoph and Brunner, Sebastian and Lehner, Peter and Reill, Josef and Riedel, Sebastian and Bodenmüller, Tim and Bussmann, Kristin and Büttner, Stefan and Dömel, Andreas and Friedl, Werner and Grixa, Iris and Hellerer, Matthias and Hirschmüller, Heiko and Kassecker, Michael and Marton, Zoltan Csaba and Nissler, Christian and Ruess, Felix and Suppa, Michael and Wedler, Armin (2016) The LRU Rover for Autonomous Planetary Exploration and its Success in the SpaceBotCamp Challenge. In: 2016 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC). IEEE International Conference on Autonomous Robot Systems and Competitions, 04-06 May 2016, Braganca, Portugal.

13. ROBEX - Robotic Exploration of Extreme Environments. http://www.robex-allianz.de/en/ . Accessed 2017-05-30

14. Czeluschke, A. and Knapmeyer, M. and Sohl, F. and Bamberg, M. and Lange, C. and Luther, R. and Margonis, A. and Rosta, R. and Schmitz, N. (2014) The ROBEX-ASN - A concept study for an active seismic network on the Moon. European Lunar Symposium 2014, 15.-16. Mai 2014, London, United Kingdom.

15. SCIL | Systems Control Innovation Lab. http://www.systemcontrolinnovationlab.de/ . Accessed 2017-05-30