

GENERIC ROVER DYNAMICS MODEL FRAMEWORK FOR AUTONOMOUS CAPABILITY DEVELOPMENT, VERIFICATION AND VALIDATION

D. Lachat⁽¹⁾, A. Donchev⁽¹⁾, A. K. Kamath⁽²⁾, P P Menon⁽²⁾, M. Watt⁽¹⁾

⁽¹⁾ Airbus Defence and Space Ltd, Gunnels Wood Road, Stevenage, SG1 2AS, UK, Email: gnc.uk@astrium.eads.net

⁽²⁾ University of Exeter, Exeter, EX4 4QF, UK, Email: P.M.Prathyush@exeter.ac.uk

ABSTRACT

Within the scope of a UK Space Agency CREST-2 project, a Generic Rover Dynamics Model (GRDM) Framework has been developed by Airbus Defence and Space Ltd. and the University of Exeter.

This new Framework provides an essential building block required in a rover Guidance, Navigation and Control (GNC) or Operations simulator. The tool supports improvements in the area of rover GNC and can be used for verification and validation (V&V) for autonomous rover operations. The Framework is capable of auto-tuning the parameters of the physical performance model simulating rover behaviour on a terrain - the GRDM - to match observed field-test data for a variety of rover configurations. The tuned model then simulates a wider range of missions conditions. It also includes tools to rapidly create new rovers configurations and perform basic testing and characterisation of the configuration files. The first version of the GRDM Framework is made commercially available.

1. INTRODUCTION

1.1. Overview

The Framework consists of 3 components:

1. The Generic Rover Dynamics Model (GRDM) for simulating rover dynamics.
2. The GRDM Rover Creator (GRC) to create the rover mechanical configuration.
3. The Auto-Tuning Tool (ATT) for optimising certain design parameters of the model.

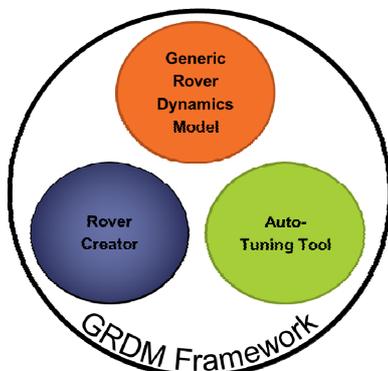


Figure 1. GRDM, ATT and GRC form the 3 components of the Framework.

The model simulates a rover's capability to traverse an unstructured environment including slippage on sand and rock climbing ability. The model is an SMP2 [1] compatible C++ dynamic/shared library which can be used in conjunction with a simulator to simulate the dynamics of the rover.

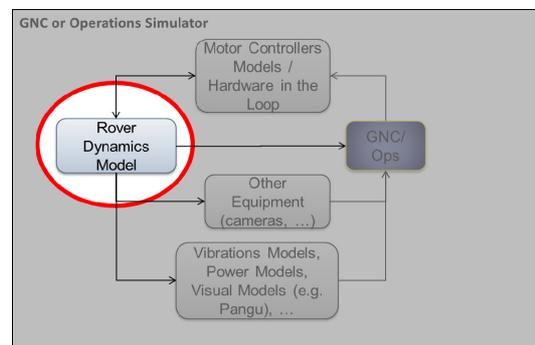


Figure 2. A rover dynamics model is one of the building blocks required in a rover GNC/Operations simulator.

The Rover Creator enables an intuitive and flexible construction of the mechanical configuration by offering two options to build the rover:

- 3D construction using a GUI
- Text-file based with a human readable file format

It also includes a Visual Checker tool providing a visual representation of the rover configuration under study and allowing testing of the correct functioning of the model with this configuration.

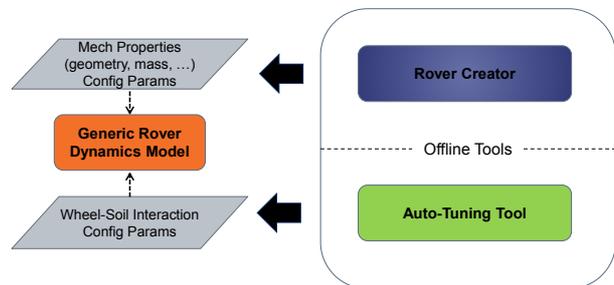


Figure 3. Generating the model configuration files offline.

The ATT optimisation procedure tunes the physical model w.r.t both sand traversing and rock climbing

empirical data. This is implemented within the ATT using a 3-stage approach with a combination of two different optimisation strategies. This enables to automatically take into account empirical data to configure the dynamics: a model configured in this way can then be used for representative simulation of wheeled off-road vehicles for a range of applications.

The obtained Framework enables intuitive and automatic construction of a new rover dynamics model to match the locomotion performance of a known rover without requiring detailed understanding of the low level implementation of the model.

1.2. Rationale

Technology improvements in the area of rover autonomy capabilities are key to a future ESA Mars Sample Return (MSR) mission [2]. The developed simulation tool supports such improvements in the areas of Guidance, Navigation and Control (GNC) design and development/demonstration of verification and validation (V&V) strategies for autonomous rover operations.

1.3. Project Scope

The Framework developed during this project aims to support:

- Rapid development & testing of rover Guidance, Navigation and Control (GNC) algorithms
- Rover operations training & rehearsal
- Rover field trials planning & review
- Development and demonstration of V&V strategies for autonomous rover operations

This project built on the existing ExoMars Rover Dynamics Model [3]: designed to run much faster than real time, it considers a simplified wheel-terrain interaction model. Hence, the wheel-terrain interaction configuration parameters have to be tuned to obtain good correlation with empirical data. The Auto-Tuning Tool was developed based on the existing Worst Case Analysis Tool (WCAT) [4].

The Rover Creator was developed in a modular manner using existing open-source libraries where possible, in order to enable best possible future enhancement and extension to the tool.

1.4. Target Applications

The GRDM Framework can aid:

- A. GNC algorithm design: taking rover dynamics into account during early project phases leads to more robust and rapid development and reduces the risk of later phase re-work.

- B. Operations training & rehearsal: a higher fidelity operations simulator leads to more optimal planning of traverses and better operator understanding.
- C. Field trials planning & review: improved planning and thorough debug beforehand leads to more cost-efficient trials.
- D. Overall rover autonomy V&V: supporting cost-efficient updates of test benches as/when the design & mission baseline evolve during the development within the project life-cycle.

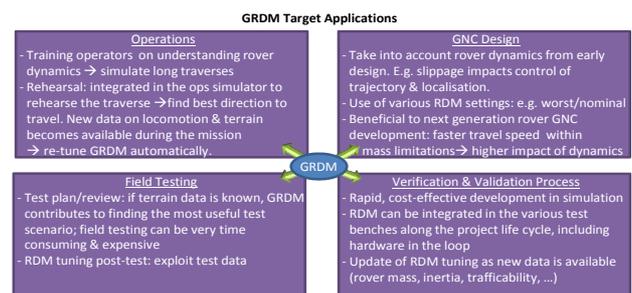


Figure 4. Target applications include GNC design, Operations, Field Testing and Verification/Validation process.

2. DETAILED DESIGN

This section describes the detailed design of the 3 components of the Framework.

2.1. Generic Rover Dynamic Model

The GRDM simulates the dynamics of a wheeled rover on an unstructured terrain, including slippage on sand and rock climbing ability. This is a performance model of rover dynamics which is implemented as a C++ dynamic/shared library. It can be used in conjunction with a simulator.

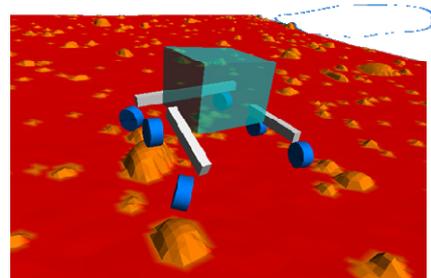


Figure 5. GRDM simulates a rover's capability to traverse unstructured environment such as Martian or Lunar terrain

The GRDM is not suitable for the study of locomotion performance: the model is not used to *study* locomotion performance; it *reproduces* locomotion performance for the purpose of the study of GNC/Operations. Indeed, the purpose of this model is to offer a tool to support the

study of rover GNC and operations as well as for the preparation, rehearsal and replay of field trials.

The tuned model simulates the *overall* locomotion performance, providing representative rover body-level behaviour, i.e. the rover travels at the correct speed and direction.

Examples of expressing locomotion performance are: slip 60% when traveling up a 10° slope, ability to climb rocks less than 20cm step rock but not greater than 30cm, etc.

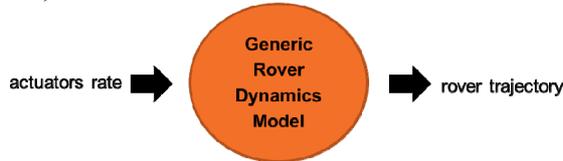


Figure 6. The model takes as input actuators rate (e.g. wheels) and outputs the resulting rover trajectory.

This design approach makes the GRDM a performance model which can be used to provide enveloping test conditions:

- Martian (or other planetary bodies) terrain properties are not fully known, and soil properties & rock type can vary greatly from different areas.
- In general, only knowledge of the approximate locomotion performance is available.
- This modelling approach enables fast run-time (in the contrary to a detailed modelling approach based on terramechanics theory which generally leads to highly computationally intensive models).

The GRDM includes:

- *Articulated rigid body dynamics* calculation.
- *Hard contact constraints*: collision detection and contact enforcement.
- *State integration* between time steps, i.e. including solver of the dynamic system, currently using ODE [5].
- *Terrain model*: modelled from a digital elevation map into a triangular mesh, non-deformable, rocks are incorporated within the mesh and soil properties are associated to each triangle of the terrain.
- *Rover model*: chassis & bogies as rectangular boxes, wheels as cylinders, active hinges for the actuators and passive hinges for the bogie pivots
- *Collision model*: collisions are detected between the terrain and the rover parts and the wheels interact with sand and rocks providing traction and slippage models (along motion and sideways). This includes rock climbing attempts and optional steering slippage. The wheels shape is not modelled exactly, instead the behaviour due to the wheel shape is reproduced (e.g. grousers).
- *Motor Model*: includes a partial motor model of backdrivable and non-backdrivable actuators

property and maximal motor torque capability. This excludes everything else, e.g. rate & acceleration limits, controllers, sensors.

- *Mechanical stops* (e.g. joints limits) can be specified (optional feature)
- In all cases, the GRDM takes as input the actuators rate

2.2. GRDM Rover Creator (GRC)

Since the GRDM can simulate many rover configurations, the Rover Creator has been designed to allow the users to create their own Rover Mechanical Configurations. The GRDM will read the Rover Configuration file and create a rover based on the parameters defined within the file.

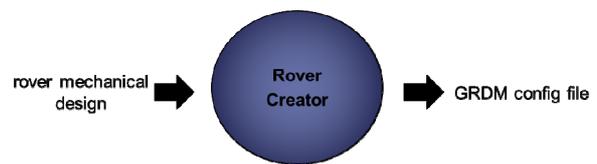


Figure 7. The rover creator takes as input information about the rover mechanical design and outputs the corresponding GRDM configuration file.

The GRC enables an intuitive and flexible construction of the mechanical configuration by including two ways to build the rover: the *basic* GRC makes use of Blender, an open source software providing a 3D constructing tool, and a text-file based *advanced* GRC with a user-friendly human readable file format. The Rover Creator also includes a Visual Checker tool that provides a visual representation of the rover configuration under study and allows to test the correct functioning of the model with this configuration.

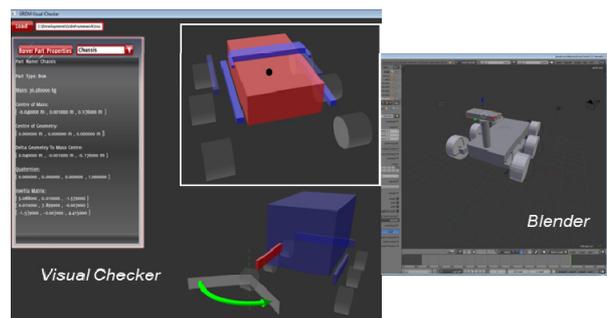


Figure 8. GRDM Rover Creator - GRC

The *advanced* GRC does not strictly require any tools except for a text based editor. The file format is fully documented: users are able to create their own GRDM Rover Configuration files. This option is intended to be used with the GRDM Visual Checker which allows to rapidly perform a visual check of what has been defined in the file. The Visual Checker also performs a

simulation check which runs the GRDM for one simulation step to test if any errors are observed.

The *basic* GRC provides an add-in for an existing 3D visual editor (Blender). This allows creation of the Rover model in Blender which can then be exported to a GRDM Rover Configuration file. This option can also use the Visual Checker. The basic GRC was developed for Windows Operating System only.

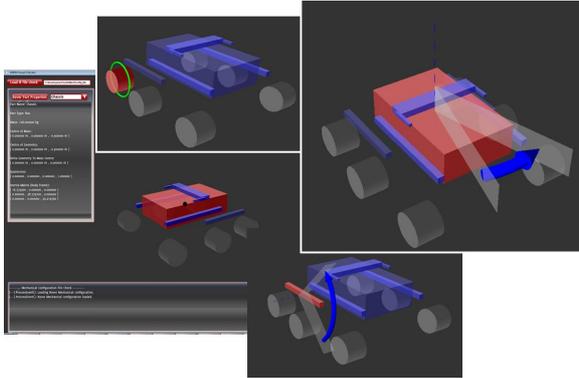


Figure 9. Examples of rover configuration design loaded in the Visual Checker

2.3. Auto-Tuning Tool

The Auto-Tuning Tool is used to tune the wheel-soil interaction parameters of the GRDM. It aims to match the GRDM behaviour to empirical data of locomotion performance. This is needed as the GRDM was designed to run much faster than real-time by considering a simplified wheel-terrain interaction model. Hence, the wheel-terrain interaction configuration parameters have to be tuned to obtain good correlation with empirical data.

In order to automate the tuning procedure, optimisation methods are applied. The optimisation procedure tunes the Generic Rover Dynamics Model w.r.t both sand traversing and rock climbing empirical data.



Figure 10. The Auto-Tuning Tool takes as input test data and outputs the corresponding GRDM configuration file.

The tuning requires an existing locomotion performance dataset. The rover locomotion performance is the ability of the rover to traverse a terrain, for example:

- Slipping along the motion while climbing up a sandy slope.
- Slipping sideways while driving across a slope.
- Slipping down a slope while turning on the spot

(point turn).

- Rock climb ability: being able to climb rocks of various shapes and sizes.

In order to determine the locomotion performance of a rover, one would ideally conduct performance field testing with a physical rover. This could be as simple as driving the rover up a sandy slope and measuring its actual speed using a stopwatch and a tape measure.

Note that it is also possible to use assumed locomotion performance from specification or the dataset could be generated with other tools.

The ATT implements two modes as illustrated in Figure 11:

1. *Tuning* – to generate the wheel-soil interaction parameters using a limited number of manoeuvres and terrain configurations (i.e. Tuning Cases).
2. *Characterisation* – to characterise the generated tuning using a wider number of manoeuvres and terrain configurations (i.e. Characterisation Cases).

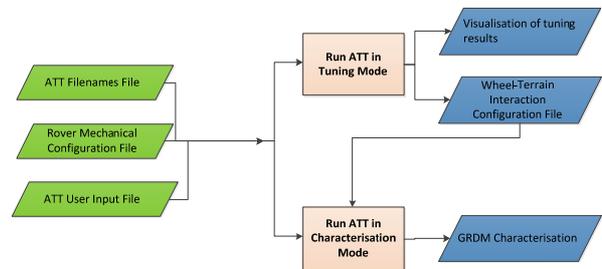


Figure 11. ATT implements two modes: tuning and characterisation.

The Tuning Mode is ran first and is described in the following sub-section.

The Characterisation mode takes as input the output of the tuning process and generates a series of figures describing the tuned simulated model locomotion performance over an extended range of cases. The characterisation information enables to have a wider understanding of the behaviour of the tuned rover model – which is essential for confidence in using the model in an overall simulator. The level of details of the generated characterisation can be selected using an input configuration file.

2.3.1. ATT Tuning Mode

The tuning mode is implemented using a 3-stage approach with a combination of two different optimisation strategies: differential evolution and Latin hypercube sampling plus local optimisation.

While the detailed design of the ATT tuning mode is the subject of another publication [6], a summary is provided here.

The ATT tunes the wheel-soil interaction parameters of the GRDM by considering two separate optimisation problems defined on rock and sand. This is because the rock climbing performance will have logical values: zero for being able to climb a rock and 1 for not being able to climb it. The resulting optimisation problem defined on rock will be in the discrete domain and falls under the category of ‘*satisfiability*’ problems. *Satisfiability* problems are difficult to solve due to the presence of many local minima and there is no guarantee that a feasible solution exists. Whereas, the locomotion performance on sand is evaluated by the slippage on sand while performing different manoeuvres. Slippage is evaluated either by using slip ratio, slip angle or slippage distance depending on the type of manoeuvre being performed. Hence the locomotion performance on sand is defined in the continuous domain and, therefore, standard optimisation methods like gradient or evolutionary based techniques can be utilised. The ATT utilises the global evolutionary optimisation method called the Differential Evolution (DE) method [7][8].

In order to solve the optimisation problem on rock, the rock climbing performance has to be assessed for various scenarios at as many points in the parameter space as possible. The ATT implements a multiple start local search procedure, which first generates samples from the tuning parameter space and then performs a local search optimisation method around each sample. This ensures maximum coverage of the tuning parameter space. The Latin hypercube sampling method is utilised to generate samples. A Nelder-Mead simplex algorithm is used to perform local search around each latin hypercube sample.

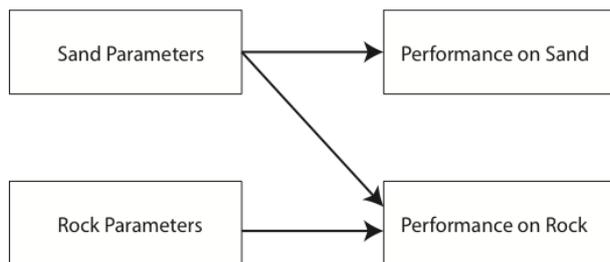


Figure 12. Impact of sand and rock tuning parameters on locomotion performance

Figure 12 shows the impact of tuning parameters on the locomotion performance on sand and rock. Rock tuning parameters do not affect the rover’s performance on sand, whereas sand tuning parameters affect the locomotion performance on both sand and rock surface. This allows to implement a tuning procedure where sand parameters are tuned first followed by the rock parameters. A three-step procedure is implemented as follows:

1. Optimisation on sand parameters using evolutionary algorithms
2. Check for tuning feasibility of rock test cases by keeping the sand parameters fixed at the solution found in step 1.
3. Optimisation on rock parameters by performing local optimisation around sample points generated by using Latin Hypercube sampling.

In order to tune the rock parameters, the sand parameters are fixed to their optimal values from step 1. This practical approach is chosen as it enables to find a solution to the optimisation problem in a reasonable amount of processing time. The drawback is that it limits the tuning parameter space and therefore in theory also limits the ability of the model to represent some locomotion performances which could be found in the parameter space not searched by this algorithm.

2.4. Implementation Technology

The GRDM is implemented in C language with an additional C++ layer which provides SMP2 compatibility. It is compatible with both Linux and Windows environments.

The Auto-Tuning Tool is implemented in C, C++ and Matlab. The Matlab compiler toolbox was used to create a stand-alone executable application which does not require a Matlab license to execute.

The Rover Creator is implemented in C, C++ and Python and makes use of several open source libraries: mainly Ogre 3D, libRocket and OgreProcedural.

The GRDM is a dynamic/shared library. Both the ATT and the Rover Creator connect dynamically to the same library that a user simulator would – ensuring that they all make use of the same model.

2.5. Limitations

The first version of the GRDM Framework has the following limitations.

Rover Mechanical Configuration and Terrain

- Terrain and rover are non-deformable
- Ball joints cannot be motorised
- Mechanical parts limited to cylinder and box shapes
- Wheels can only be of cylinder shapes
- Only one joint can be attached between 2 mechanical parts
- Terrain is a square Digital Elevation Map (DEM) with side length in pixels multiple of 100

Wheel-Terrain Interaction Model

- Locomotion performance is independent of drive speed
- Longitudinal slippage vs slope angle has a linear

relationship

- The soil properties are homogeneous across the terrain

Software Implementation

- API is in C++
- Only the following OS are supported: Windows 7 64 bits & Linux RedHat/CentOS 6.3 64bits
- Full state saving not supported: if wanting to re-start in the middle of a simulation, the results will not be exactly reproducible to the numerical accuracy. Note that it will however be reproducible if re-starting from the beginning of the simulation.
- Currently used SJLJ exception handling which may not work correctly if linked to an executable which uses a different exception handling runtime.

Auto-Tuning Tool

- Generates a wheel-terrain interaction configuration file to its best effort.
- Depending on the complexity of the rover, tuning cases and the amount of information provided as input to it, it may not be able to find a tuning matching every locomotion performance.

3. RESULTS

The development of the first version of the GRDM Framework is completed. The following sub-section describe the testing which has been performed during the project.

3.1. Testing Overview

Functional regression tests were performed of the GRDM with respect to the ExoMars RDM and comparative runtime performance tests were executed. The GRDM SMP2 wrapper was integrated and tested within an SPM2 compatible simulator. The Rover Creator and the GRDM model were tested with various mechanical configurations, including the NASA MSL and the FASTER rover configuration. The Auto-Tuning Tool was tested using ExoMars Rover LPM breadboard datasets [9] and dataset shape variations to enable testing of the behaviour of the tool with various types of inputs.

3.2. GRDM Testing

The model functionalities were tested with mechanical configurations of the ExoMars rover, the NASA MSL rover and the FASTER rover as well as simple 4-wheel rover configuration.

The GRDM has shown to run slower than its non-generic ExoMars counterpart, which is as expected due to a) the generic aspect of the GRDM implementation and b) the implementation has not been optimised. As an example, using a model of the ExoMars Rover LPM

breadboard, the GRDM runs $\sim 19.5x$ faster than real time on a desktop computer (standard office computer: intel core i3 3.3GHz, 4GB RAM). This is in average 40% slower than the ExoMars non-generic model.

Nevertheless, the current runtime performance is considered very good and adequate for most applications.

3.3. Rover Creator Testing

The Rover Creator was tested by creating several different rover configurations using the basic and advanced GRC as listed in Tab. 1. Various erroneous configuration files were written to test the Visual Checker parsing functionality and error checking capability. In addition, during development of the Blender add-in (basic GRC), all exported files were loaded in the GRDM Visual Checker to ensure that the Blender model was exporting the Rover models correctly.

Table 1: List of Rovers used to test the GRC.

Rover	Test Type
ExoMars rover	Basic & Advanced GRC
MSL rover	Advanced GRC
Small basic rover	Basic GRC
FASTER Bridget	Advanced GRC and Visual Checker to check the configuration of the rover

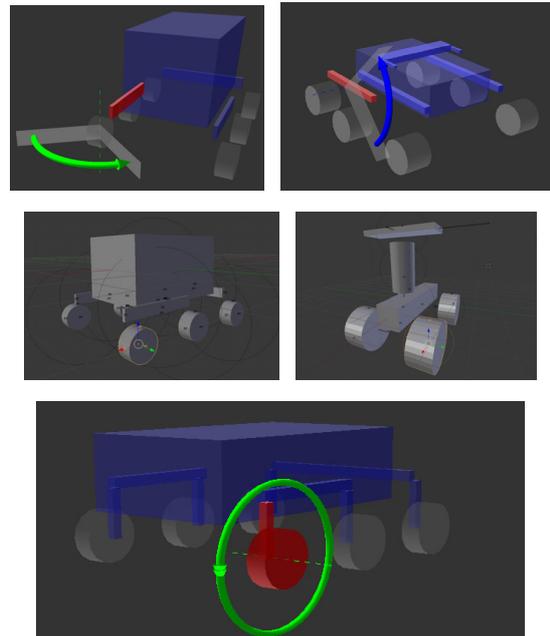


Figure 13. Several different rovers were created using both the basic and advanced GRC.

3.4. ATT Testing

The Auto-Tuning Tool tests using the ExoMars Rover LPM breadboard datasets are presented here. The dataset consists of 10 test cases for sand and 8 test cases for rock, which is covering a wide range of operations

of the GRDM. More details about the test cases can be found in [6].

The first step of the ATT procedure is to find an optimal solution for the optimisation problem on sand. The tool outputs a normalised design metric plot and a percentage error plot. The normalised design metric plot shows the number of test cases which have their design metric values within their desired bounds. Figure 14 shows that out of 10 test cases 8 are within their desired bounds. In **Figure 15**, the percentage error plot shows the errors of the design metric values which are lying outside the bounds. Note that zero percent error (0%) is assigned to the test cases which have their design metric values within the bounds. All test cases have percentage error less than or equal to ~5%. This error percentage is considered to be very small and acceptable for the current application. It is to be noted that the reference dataset also has its own uncertainty as it is measured by field trail.

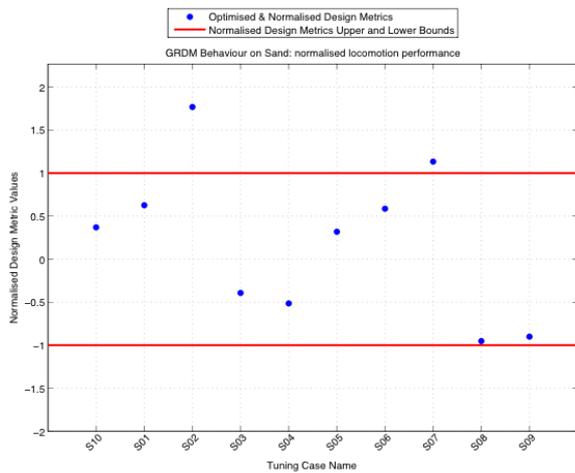


Figure 14. Sand optimisation results: normalised design metric

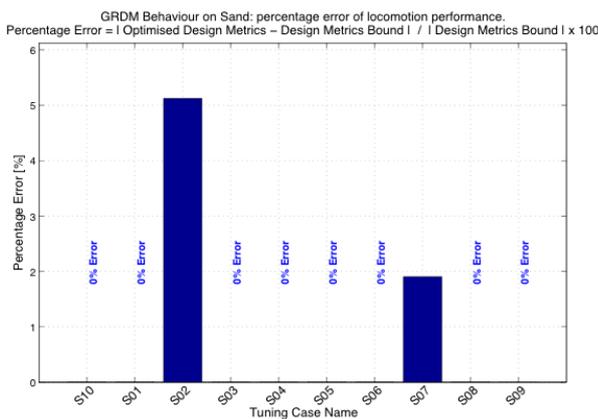


Figure 15. Sand optimisation results: percentage error

For the second step, the optimised sand parameter values are fixed and the feasibility of tuning the rock test cases with these settings is tested. Out of 8 test

cases, two cases could not be tuned as the rover model is never able to climb these rocks with the parameters tuned for the sand cases. The remaining 6 test cases are considered for tuning rock parameters.

In order to tune the rock parameters, a multi-start local optimisation method as described in [6] is utilised. A total of 15,000 simulations are performed to get a good coverage of the tuning parameter space. Even so a single solution satisfying all the test cases were not found. In such a scenario, the ATT plots the next five best solutions. One such solution is shown in **Figure 16**: when the cross is inside the circle, then the corresponding test case is satisfied. The circles represent the required climbability (i.e. the rover ability to overcome a rock) while the crosses represent the climbability achieved by the rover during the simulations. The x-axis represents the rock test cases while the y-axis has two values, namely ‘Climb’ and ‘Not Climb’.

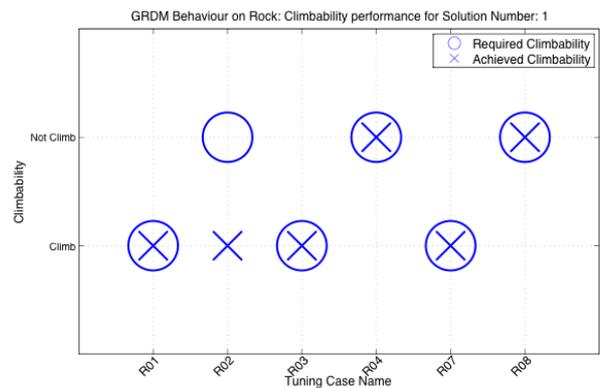


Figure 16. Rock test case solution no. 1

Figure 17 shows the satisfiability compatibility map which describes the compatibility between two test cases w.r.t satisfying both of the test cases simultaneously. It indicates the percentage of evaluated solutions that satisfy both test cases together and enables to identify if some of the test cases are incompatible. The map is colour coded indicating the number of solutions present in each cell. White colour represents $\geq 90\%$ of solutions satisfying both test cases together. Dark brown colour represents 0% of the solutions satisfying both test cases together. This map can help the designer to understand which rock test cases are difficult to tune and which are easier. It will be easier to tune parameters if there are a greater number of solutions satisfying two test cases together. The optimisation procedure can then be re-run with modified settings and/or trading-off between the requirements of the application for the various test cases.

These results demonstrate that the optimisation approach adopted in the current version of the ATT is suitable for this application as it has given very good results, this despite some limitations induced by the

implemented 3-stage approach selected for practical reasons.

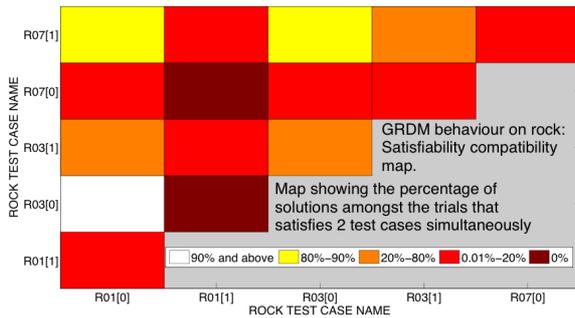


Figure 17. Satisfiability compatibility map

4. NEXT STEPS

The GRDM Framework is currently being used by the University of Glasgow within the scope of a UKSA CREST-2 project which develops a rover autonomous guidance using inverse modelling techniques. The Framework is also being used as an integration demonstration case for the ESA Harwell Robotics Autonomous Facility (HRAF) pilot project 1.

4.1. Potential Additional Features

The following additional features are possible future work:

GRDM: runtime performance optimisation and model extension (e.g. add different type of joints, write more languages API). Implement a GRDM Viewer tool for user simulator development/debug purposes.

GRC: support more file formats (e.g. vrm, catia), implement a fully integrated rover mechanical configuration building tool and enhance the Visual Checker user experience (e.g. interactive integrated 3D viewer for the GRDM test).

ATT: add a user defined design metric to enable any kind of manoeuvre case to be used for the tuning and add palliative functionality for under-defined locomotion performance dataset. Expand the optimisation algorithms to enable coverage of the whole parameter space.

5. CONCLUSION

This project has developed a fully working Generic Rover Dynamics Model Framework which is now commercially available. The testing has demonstrated the generic aspect of the model and the rover creator with some identified limitations. While the implementation has not been optimised, the model runtime performance is considered adequate for most applications. The auto-tuning tool has provided very good results with the dataset under test, however further testing with different rover datasets would be very

beneficial and evolution of the optimisation algorithm may be required to ensure ability to match most locomotion performances.

Several improvements have been identified. Early user feedback may indicate the priority of the improvements, in order to concentrate the potential future efforts on the most critical aspects..

6. ACKNOWLEDGEMENT

This research work has been funded by Airbus Defence and Space Ltd., University of Exeter and the UK Space Agency.

7. REFERENCES

1. C. Cazenave, W. Arrouy, Astrium. Implementing SMP2 Standard within SimTG Simulation Infrastructure. *SESP (ESA) : Simulation and EGSE for Space Programmes*, 2012.
2. J. Vago, B. Gardini, G. Kminek, P. Baglioni, G. Gianfiglio, A. Santovincenzo, S. Bayon, and M. van Winnendael. Exomars-searching for life on the red planet. *ESA bulletin*, vol. 126, pp. 16–23, 2006.
3. R. Lancaster, N. Silva, A. Davies, and J. Clemmet. Exomars rover gnc design and development. *ESA GNC Conference*, 2011.
4. D. G. Bates, W. Wang, P. P. Menon, and S. Bennani. Verification and validation of controllers using the worst-case analysis tool. *Proceedings of the 8th International ESA Conference on Guidance, Navigation and Control*, 2011.
5. <http://www.ode.org>
6. A. K. Kamath, D. Lachat, M. Watt, P. P. Menon. Automated Parameter Tuning of a Generic Rover Dynamics Model for Autonomous Planetary Space Capability Development, Verification and Validation of Space Applications. *Proc. of the American Control Conference*, 2015.
7. R. Storn and K. Price. Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.
8. P. P. Menon, D. G. Bates, and I. Postlethwaite. Hybrid evolutionary optimisation methods for the clearance of nonlinear flight control laws. *44th IEEE Conference on Decision and Control, European Control Conference*. CDC-ECC '05., December 2005, pp. 4053–4058.
9. P. Meacham, N. Silva, and R. Lancaster. The development of the locomotion performance model (LPM) for the ExoMars Rover vehicle. *ASTRA Conference*, 2013.