

Proceedings of the Joint 9th Asia-Pacific ISTVS Conference and
Annual Meeting of Japanese Society for Terramechanics
Sapporo, Japan, September 27 to 30, 2010

A comprehensive wheel-terrain contact model for planetary exploration rover design optimization

Alexandre Carvalho Leite

German Aerospace Center (DLR) – Institute of Robotics and Mechatronics,
alexandre.carvalholeite@dlr.de

Bernd Schäfer

German Aerospace Center (DLR) – Institute of Robotics and Mechatronics,
bernd.schaefer@dlr.de

Abstract. *In the context of the European Space Agency's (ESA) Aurora Programme there are two near future robotic missions which will provide surface mobility with wheeled rovers: ExoMars and Next Lunar Lander – NLL. These vehicles are the main motivation of this work; they are supposed to drive with high performance over rough terrains containing basically rocks, sand and inclined slopes. Our contact model aims to describe the interaction between each rigid wheel of a rover and three different terrain types: 1) rigid surface, 2) complex-shaped rigid objects (representing rocks), and 3) soft soil. Transition among these three “meta-states” is permitted for each wheel of a multi-wheeled rover. The uneven terrain is smoothed on the space domain giving a continuous three-dimensional surface; the discontinuities are depicted by the rigid objects, and its collision is detected by the proper computation of the contact plane and the contact forces. The uneven terrain can be as rigid as the rocks (modeled by Coulomb's friction law) or as soft as sand (modeled by Bekker's equations). Optimization of the mechanical structure is a crucial task in the design phase to achieve high performance. However, there are three essential features which are required to perform batch simulations with the optimization process: stability, robustness and speed of the simulations. These required features have guided the selection of the used impact/rolling models. The internal states and its transitions are thoroughly explained to clarify the main difficulties in the batch simulations. The contact model is partially validated by drawbar-pull experiments in soft soil testbed with a breadboard model of the ExoMars rover. Future advanced versions of the contact model are focusing on flexible wheel modeling, and first attempts in that direction are also provided and commented.*

Keywords. Contact modeling, multibody simulation, planetary rover optimization.

1 Introduction

In 2001 the European Space Agency (ESA) had initiated the Aurora Programme with its primary objective to cover robotic and human exploration of the solar system. The most likely targets are Mars, the Moon and asteroids; but a manned mission is only possible after successive robotic precursor missions to test technology in-situ and gather sufficient environmental information. In this context there are two near future missions which will provide surface mobility with wheeled

rovers: ExoMars (Patel et al., 2010) and Next Lunar Lander NLL (Gibbesch et al., 2010). The ExoMars Rover and the NLL Rover are powerful rovers with different capabilities under development, respectively with six and four wheels. Such vehicles that drive over rocks, loose soil and general slippery and uneven terrain often need high mobility capabilities.

These capabilities are very demanding from the engineering's point of view. Thus, a designing assistance tool is highly desired to relate dynamic behavior and system requirements on a consistent way and automatically provide advisable designing choices. The present work describes the modeling effort employed to develop a satisfactory dynamic simulation model and integrate such tool.

Section 2 starts with the base of the model, the travelling surface and its mathematical description to form an arbitrary landing site. Hence the driving behavior is geometrically constrained by the terrain; the calculation of the contact forces is detailed in section 3 to complete the model. Section 4 clarifies some emergent problems in the context of variation of design parameters of the simulation model into an optimization procedure scheme. Section 5 presents some results of the modeling effort and locates the present work inside the structure of the rover design assistance tool under development. Finally, conclusions and future directions to improve and advance the current version of the simulation model are stated in the 6th section.

2 Modeling of the terrain

In this work a terrain is considered as a continuous surface partially covered by interspersed discontinuities. The continuous layer can be a hard (bedrock) or soft (sandy) surface. The discontinuities are stones of various sizes, materials and shapes. Based on this abstraction the next two subsections describe the cited layers, the contact dynamics involved is discoursed in the next section.

2.1 Relief pattern

Regarding the relief pattern, it is valid either for rigid or soft soils and defined at the contact patch of each wheel of a multi-wheeled planetary exploration rover. A three-dimensional surface in the form $z = f(x, y)$ is a scalar field used to compute the orientation of the contact's normal vector e_n , where (x, y) are points of the plane and z the respective terrain elevation, see figure 1.

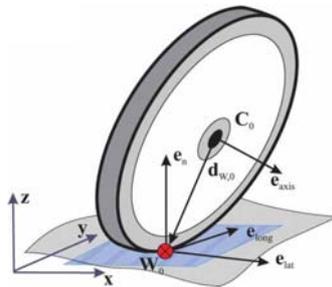


Figure 1. Coordinate system of the contact point, (Zimmer and Otter, 2010).

The rotation axis of the wheel e_{axis} follows the attitude of the wheel, this is used to determine the longitudinal direction ($e_{long} = e_n \times e_{axis}$) of the contact point's coordinate system. The right-handed

coordinate system is completed with the lateral direction as $\mathbf{e}_{lat} = \mathbf{e}_{long} \times \mathbf{e}_n$. In this way the coordinate system of the contact point of the wheel is locally defined as a function of the terrain and orientation of the wheel. The normal vector is the direction where the normal force is computed; longitudinal and lateral vectors are in the tangent plane, where the traction force and motion resistance forces occur.

2.2 Rocks

Our main requirement for a rock is an unsmooth surface. The simulated landing site can have several rocks with that characteristic avoiding as much deterministic rules as possible; it should be accomplished in order to imitate the immanent lack of knowledge about the stones on a real landing site. The rocks are generated during a preprocessing stage and can be used in each simulation run with new placements.

Figure 2 illustrates the procedure of creating an unsmooth rock and place it on a terrain. The first three steps are performed only once (in a preprocessing stage) and stored for future usage; the last step is performed before each simulation run to constitute a new scenario with the previously generated/stored rock geometries.

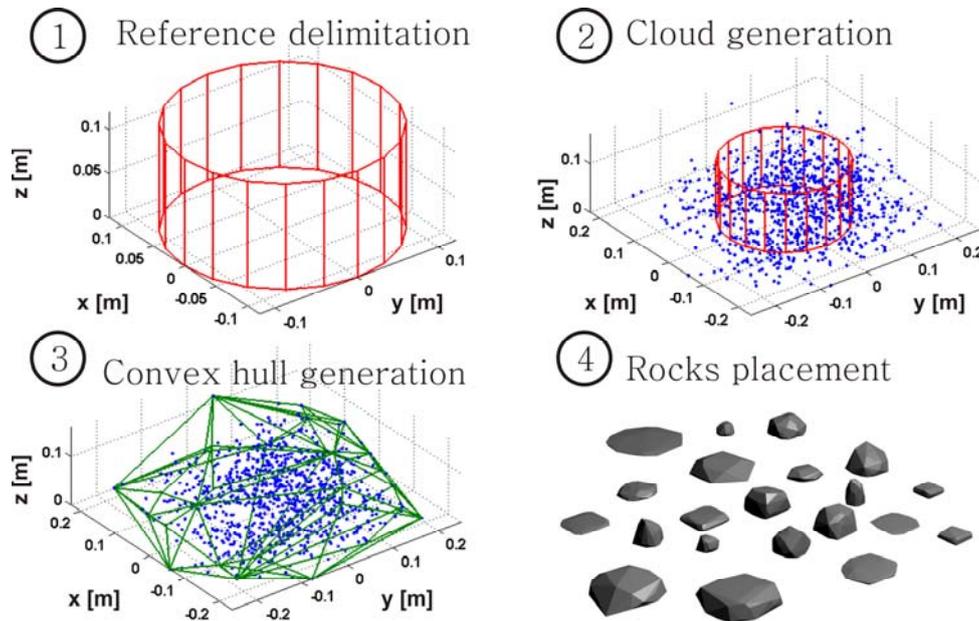


Figure 2. Steps performed to generate rocks (1 to 3) and spread them on a landing site (4).

In the first step a cylinder is used as a reference bound to the rock dimensions, basically the radius of the base and its height are the reference bounding parameters. In the second step a vector defined in cylindrical coordinates runs inside the contour surface of the cylinder with randomly varying radial distance, azimuth and height generating the cloud of points around the reference bounds. A convex envelope (convex hull) for the cloud of points is computed in the third step through (Barber et al., 1996), only the vertices and faces of the envelope are stored as a DXF file for visualization and contact computations during the simulation. The fourth step consists in distributing the rocks to create essentially three scenarios: specific placement for simulation tests; specific placement for worst case evaluation and placement following (Golombek, 2002) to achieve representative landing sites.

One can see in the rendered image (fourth step of figure 2) that assorted shapes and sizes can be obtained by the variation of: reference bounding parameters and standard deviations of the Gaussian distributions used to generate the cloud of points of the second step. The dimensions of the rocks can be controlled by the reference bounding parameters, the smoothness and shapes of the rocks can be controlled by changing the standard deviation or even the probability density function in step two. Obviously there is the possibility that the shape, smoothness and dimension can be further controlled by changing the primitive used as reference bound; instead of a cylinder, polytopes composed of hemispheres, cylinders and similar geometric primitives could be used.

Once the rocks are placed in the simulation model, they can assume a specific friction profile which depends on the position of the contact point or its velocity over the unsmooth rock's surface. Such aspects are discussed in section 3.2, where the contact between wheel and rock is detailed.

3 Contact Modeling

3.1 Wheels - rigid surface

To compute the normal force the penalty method is used. It consists of a compliance system giving forces that are proportional to violation of constraints to solve the constraints, the forces are computed in the normal and tangent direction by spring-damper model (Kraus et al., 1997), which can be linear or non-linear. Equation (3.1) is the general form of the normal force computation and equation (3.2) is the general form of the tangential force vector computation.

$$F_{N,i} = f_{N,i}(\delta_{N,i}) + g_{N,i}(\delta_{N,i}, \dot{\delta}_{N,i}) \quad (3.1)$$

$$F_{T,i} = f_{T,i}(\delta_{T,i}) + g_{T,i}(\delta_{T,i}, \dot{\delta}_{T,i}) \quad (3.2)$$

where the terms $f_{N,i}(\cdot)$ and $f_{T,i}(\cdot)$ are representing the spring forces, the terms $g_{N,i}(\cdot)$ and $g_{T,i}(\cdot)$ are representing the damper forces, $\delta_{N,i}$ and $\delta_{T,i}$ are the corresponding displacements of the spring-damper system with its respective time derivatives. At this stage we have a kinematical problem with additional states, this is justified by the avoidance of the static indeterminacy of the normal forces on a multi-wheeled vehicle. This is a reasonably quick and accurate solution for normal force calculation (see figure 3a), the forces developed on the tangent plane are based on Coulomb's model including pure rolling and slippery behavior. The implementation of the Coulomb's friction model is not a simple task in our simulation/modeling tool (Dymola), since the order of the system of DAEs (Differential Algebraic Equations) changes during the simulation. In other words, it becomes a variable-structure system and can be approached in different ways. In (Trinkle et al., 1997) a solution with an approximation of the friction cone by a friction pyramid is proposed by means of a Linear Complementarity Problem (LCP) formulation, with the assumption that the solution is always possible. A regularized Coulomb friction is extensively used (Zimmer and Otter, 2010) to get a continuous and easily tractable version of the Coulomb model, the weakness of this approach is in the zero-crossing in the region of zero-slip (this is reasonable if one assumes that there is no zero-slip situation). Another solution proposed in (Kraus et al., 1997), applied previously in (Sohl and Jain, 2005) and also adopted here, uses a compliance system to compute the tangential forces inside the friction cone with a little amount of slip allowed between rolling-slipping transitions and vice versa. In this approach, the 2-dimensional spring-damper is required to maintain the

nonholonomic rolling constraint until the maximum force μF_N (μ is the Coulomb's friction coefficient) is reached. In this moment the 2D spring-damper can change its direction but its displacement is limited by a circle with the radius proportional to μF_N , see figure 3b. The implementation of the compliance system in three dimensions is shown in figure 4.

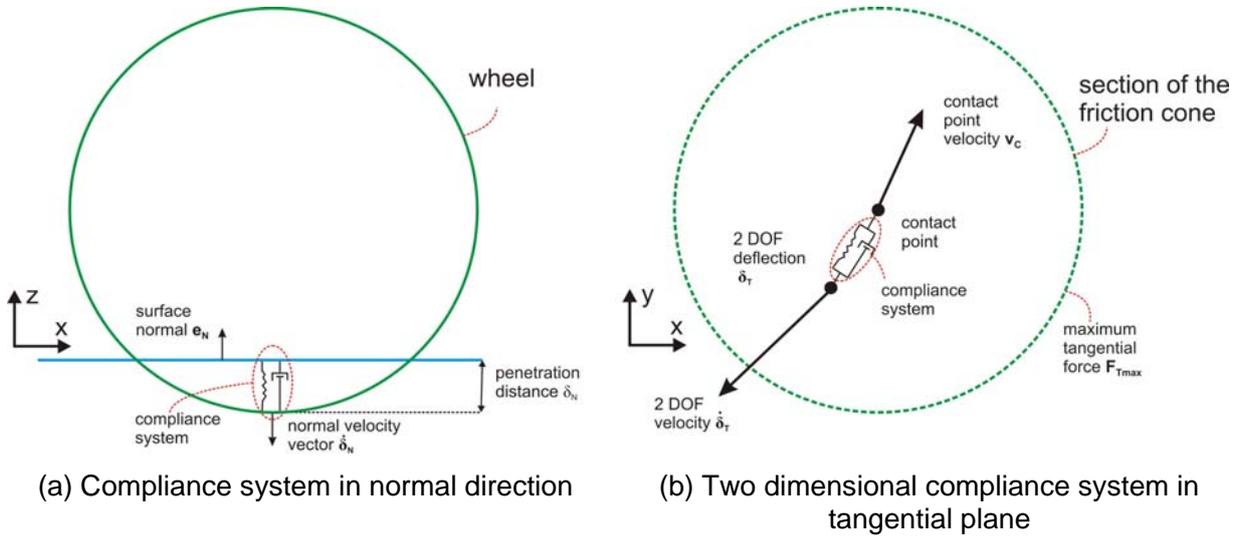


Figure 3. Compliance system interacting with soil (a) and Coulomb constraints (b).

Despite the advantages of the compliance model for the tangential forces, the success of this scheme is sensitive to the parameters of the spring, damper and also sensitive to the equation used to model the spring-damper system. It would not be a problem for a fixed configuration of a planetary rover, i.e. definitive dimensions of the mechanical structure. But since the mechanical structure is changed for each simulation run during the optimization procedure, these parameters must be suitably changed or even fixed in order to accurately comply with each new scenario. Nevertheless there are means to work round unwanted results, the simulation conditions can be fixed and greatly contribute to accommodate simulation stability during the overall optimization process.

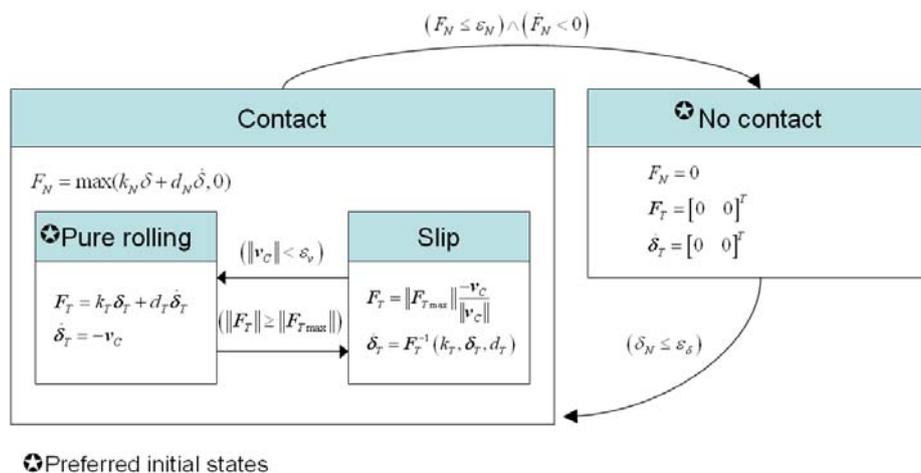


Figure 4. State transitions of the friction contact model.

3.2 Wheels - rocks

The main difference between the rigid contact model of the last section and the present model is that in the previous the static surface of contact is analytically defined and in the present case not, thus it must be detected by a numerical collision detection procedure. The stones are generated as described in section 2.2 and returned in a DXF file accessible by our C API Wrapper. This C API uses the information preprocessed in Matlab to represent it in a convenient way as an input to the collision detection library (SOLID 3.5.6). SOLID takes the description of the stone in vertices and faces and the shape of the wheel is described by a support mapping of a cylinder. Figure 5a shows the main task performed by SOLID inside Dymola, locate the two nearest collision points p_m and p_s . They are respectively located on the surface of the moving object (the wheel) and the static object (the stone).

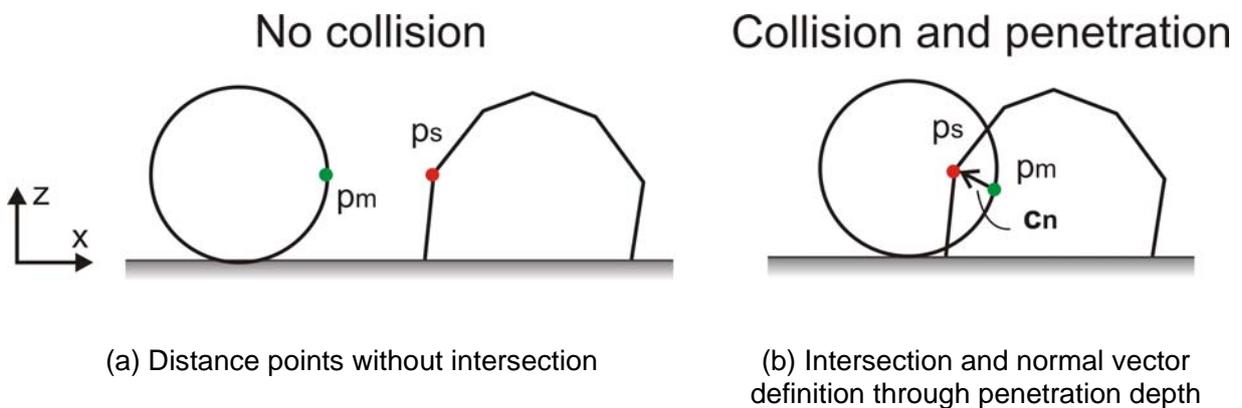


Figure 5. Computation of closest points (by SOLID) and normal vector in intersection.

Contact forces are computed only in case of intersection, figure 5b. When intersection occurs those points are used to compute the contact normal c_n . After that the tangent plane is computed as for a conventional relief pattern in section 2.1. The forces can be computed with Coulomb behavior as in section 3.1 or with a regularized Coulomb behavior. The user of the simulation model can choose the contact force calculation according with its purposes.

One can ask about the penetration depth, it would not be a realistic assumption for collision between rigid bodies. Without the penetration depth, say $p_m - p_s = 0$, the normal vector is undefined and the contact calculation can no longer proceed. The maximal penetration depth is also a simulation parameter, this is a tolerance value which impacts in the speed and accuracy of the simulation. In the performed tests we used a tolerance value of $5\mu\text{m}$ for penetration depth.

If Coulomb behavior is desired the compliance system can assume linear and nonlinear equations for the force computation. Usually the nonlinear equation of Hunt-Crossley (Hunt and Crossley, 1975) gives the best results because it avoids increasing the penalty on the velocity term due to lower penetration depths.

3.3 Wheels - soft soil

Bekker (Bekker, 1956) developed the theory which is the basis of the wheel-soft soil contact model in this section. Furthermore our approach allow for a broader range of cylindrical wheel designs and complex varying normal and shear stress distributions. Firstly we describe the longitudinal forces and then the lateral forces in the following. At last, soft soil driving modes are described. The outcome is a model for interaction between a rigid wheel and a sandy surface on a specific relief pattern as is illustrated by the screenshot (figure 6) of one of our single wheel driving simulations.

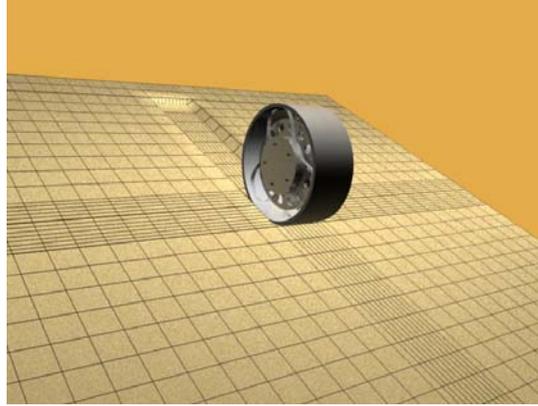


Figure 6. Visualization of a single wheel driving simulation on a sandy relief pattern.

Longitudinal forces

We start from the drawbar pull equation to clarify which forces are assumed in soft soil. Drawbar pull DP is simply defined as the difference between traction force F_t and motion resistance force F_r , that is the additional force available to transport the payload and is defined as

$$DP = F_t - F_r \quad (3.3)$$

Traction in soft soil is limited by the available soil thrust, which depends on properties of the soil so that the maximum available traction force is considered as a function of the longitudinal shear stress τ_x :

$$F_t = \int_P \tau_x dP \quad (3.4)$$

where P is the surface area of the contact patch and τ_x is distributed over the entire area P . In spite of the cylindrical shape, the contact patch cannot always change in a well behaved way as the rover drives and surmount obstacles. We take this into consideration by calculating the longitudinal shear stress for each slice of the discretized wheel (see figure 7a-b) as in (Wong, 2001):

$$\tau_{xs} = \left[c + \sigma(\theta_s) \tan \phi \right] \left(1 - e^{-j_s(\theta_s)/k_x} \right) \cos \theta_s \quad (3.5)$$

where c is the cohesion module, σ the normal pressure distribution and ϕ the internal friction angle. The subscript s denotes the slice in which τ_{xs} is integrated along the domain θ_s of a

slice of the discretized wheel. The longitudinal soil deformation $j_x(\theta_s)$ is defined as in (Ishigami, 2008). The normal stress distribution is defined individually for each slice, see figure 7c.

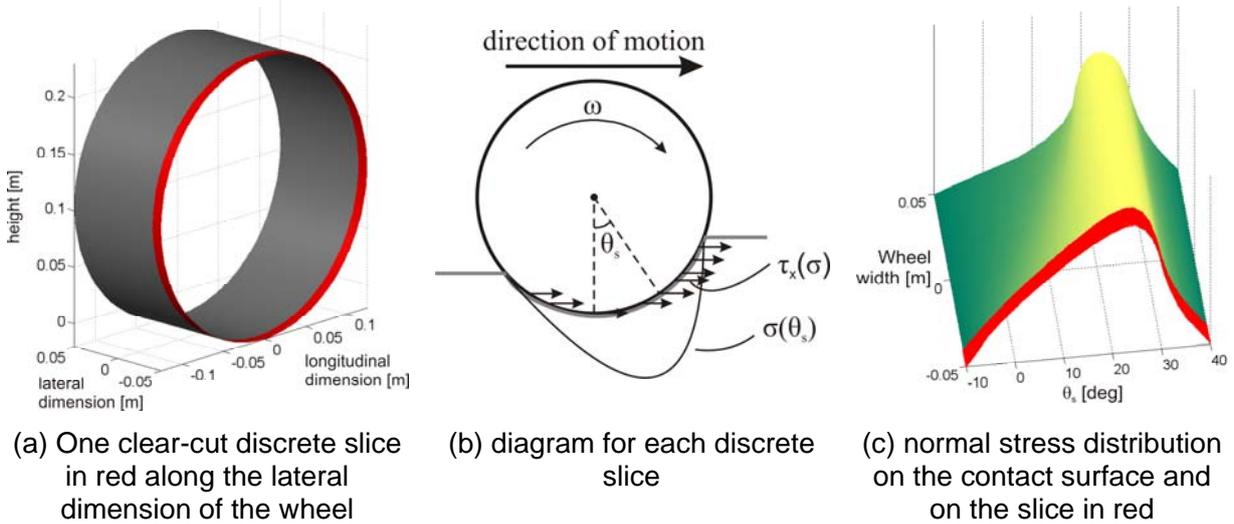


Figure 7. Laterally discretized wheel approach illustrated.

Note that the determination of the stress distribution is relatively complicated and there are analytical propositions to represent experimental data as in (Yoshida et al., 2003). Seeing that the traction force is computed numerically, we try to take advantage of that procedure by defining the normal stress distribution also numerically. The basis of our approach is the application of the high-speed Chaikin curve generator (Chaikin, 1974), where “control points” are displaced to shape the final curve accordingly. In figure 8a there are six “control points” represented as squares following a path to convert the light gray curve into the dark gray one. Figure 8b shows that six “control points” can represent experimental data better than an analytical proposition like that used in (Ishigami, 2008) and (Yoshida et al., 2003). The experimental data of figure 8b were digitized from (Hegedus, 1962). Six “control points” are sufficient and give a broad control of the curve’s shape; other quantities were tested without significant performance and fitting improvements.

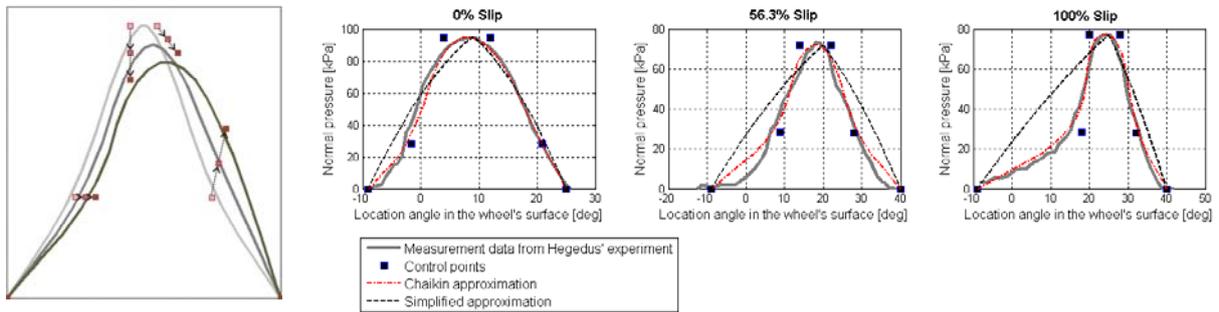


Figure 8. Normal stress distribution shaping based on Chaikin curve generator.

Motion resistance force F_r is defined as the sum of three resistance forces

$$F_r = F_c + F_b + F_l \quad (3.6)$$

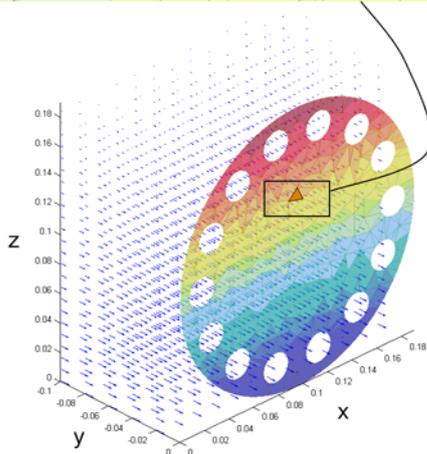
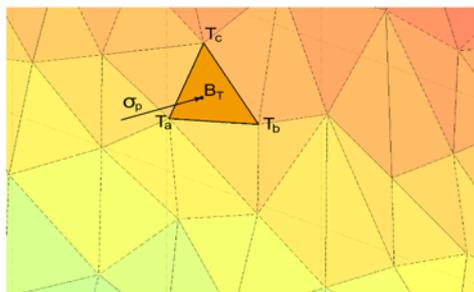
where F_c is the soil compaction resistance, F_b the bulldozing resistance and F_l the lateral drag resistance. The soil compaction resistance occurs in the contact patch when it compresses the soil. This can be calculated from the normal pressure distribution:

$$F_c = \int_p \sigma dP \quad (3.7)$$

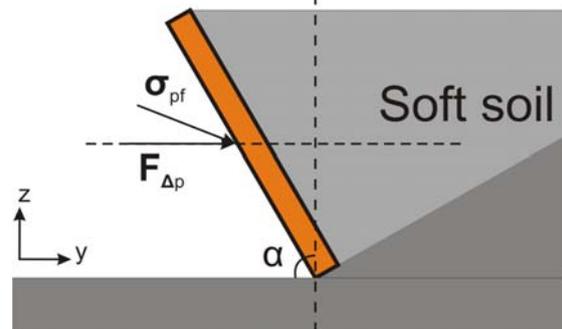
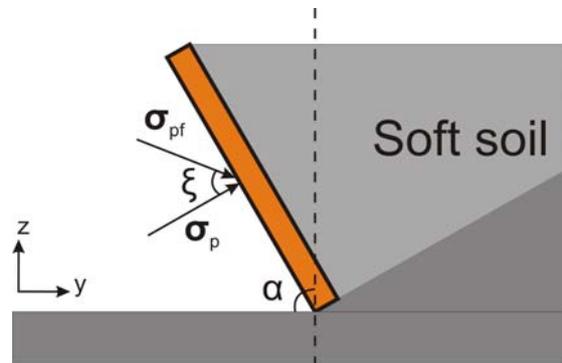
The bulldozing resistance is a bit more complicated because it is a result of displacement of the soil in front and also at the side of the wheel when it slips sideways with the slip angle β :

$$F_b = F_{b,front} + F_{b,lat} \cos \beta \quad (3.8)$$

Since the wheel is considered as a cylinder, the bulldozing resistance term $F_{b,front}$ could be obtained in a closed form as in (Scharringhausen, 2009). But the term regarding the lateral soil displacement depends on the shape of the sides of the wheel, which can be changed by the rover's designer to improve trafficability performance. To compute $F_{b,lat}$ we represent the lateral shape of the wheel as a triangle mesh submerged on a pressure vector field defined according with earth passive pressure; see the circular plate in figure 9a-bottom spiked with circular holes as the lateral design of the wheel in figure 6. Each triangle $\Delta T_a T_b T_c$, see figure 9a-top, of the mesh is hit by a vector σ_p in its barycenter B_T , where σ_p is defined in the vector field according with the sinkage as in (Wong, 2001).



(a) Triangle mesh in the pressure vector field



(b) Contact geometry of a mesh element

Figure 9. Computation procedure of bulldozing resistance due to side slip

Each triangle may possess an inclination α relative to the normal direction of the relief pattern, regarding the wheel's construction or traveling conditions during the simulation. The friction angle between the rough surface of the wheel and the soil is ξ . The geometry of contact for each triangle is shown in figure 9b. In figure 9b-top the vector σ_p is normal to the face of the triangle and σ_{pf} is the correction as a function of the friction angle ξ . In figure 9b-bottom the bulldozing force $\Delta F_{b,lat,p}$ is given in the horizontal direction for one triangle of the mesh. It means that $F_{b,lat}$ is the sum of $\Delta F_{b,lat,p}$ distributed over the mesh:

$$F_{b,lat} = \sum_{p(\text{mesh triangles})} \Delta F_{b,lat,p} \quad (3.9)$$

where the index p stands for the triangles in contact with the soil. The lateral drag F_l (which acts on longitudinal direction) caused by the shearing of the soil along the lateral extent of the wheel is considered as proportional to the sunk area at that extent. It may be readily estimated as:

$$F_l = k_l A_s \quad (3.10)$$

where A_s is the sunk area computed numerically during each integration step of the simulation, and k_l is a lateral drag constant which can be determined through experimental data.

Lateral Forces

The lateral forces arise in two different cases, steering maneuvers and side slip situation. They are computed as functions of lateral shear stress and bulldozing resistance due to soil displacement at the side of the wheel:

$$F_{lateral} = F_{ls} + F_{b,lat} \sin \beta \quad (3.11)$$

note that $F_{b,lat}$ is computed as in (3.9), and the term F_{ls} is computed similarly as in (3.4) but taking the lateral shear stress τ_y into account, instead of the longitudinal one.

$$F_{ls} = \int_p \tau_y dP = \int_p [c + \sigma(\theta_s) \tan \phi] (1 - e^{-j_y(\theta_s)/k_y}) dP \quad (3.12)$$

There is also a functional dependence of τ_y for each discrete slice (figure 7a) of the wheel. The lateral forces are defined in the opposite direction of the wheel's lateral velocity.

Soil driving modes

We considered basically three modes in soft soil driving: Stuck Mode, Accelerating Mode and Breaking Mode. At the start of the mission and when the rover stops in order to steer the wheels and drive again, a mathematically indeterminate form takes place. This is the case of indeterminate slip ratio ($\omega = 0, v = 0, s = (r\omega - v)/v = \text{indeterminate}$), where ω is angular velocity of the wheel, v longitudinal translational velocity and r its radius. Since the contact model in soft soil is a function of slip ratio, the forces can no longer be calculated at that instant.

To overcome this problem we assume that the vehicle is in Stuck Mode (i.e. $r\omega - v = 0$) and the maximum force sustainable before occurrence of slippage is the soil thrust at zero slip $F_t(s = 0)$. Still in Stuck Mode we assume that the soil is already prepared when $\omega = 0$ and $v = 0$ occurs, and compaction and bulldozing resistance become both exponential (in order to have a continuous description) functions of the wheel's forward velocity as a matter of describing a kind of entering phase in unprepared terrain. Each resistance force term in (3.6) is reviewed:

$$F_r = \text{sign}(v)(F_c + F_b + F_t)\left(1 - e^{-|v|/k_r}\right) \quad (3.13)$$

The velocity constant k_r is suitably determined taking simulation issues and nominal velocity into consideration. When the rover leaves the Stuck Mode, there is a transition to the Accelerating Mode. In this mode the forces are computed exactly as in the equations above. From Accelerating Mode the vehicle can go smoothly to the Breaking Mode (negative slip ratio) or Stuck Mode again. However, soil deformation j_x behaves in a different way for negative slip ratio values. The shear stress distribution can always easily be shaped (through slippage dependent control points) to fit the current slip value.

To implement this hybrid system we used a similar compliance system approach to that of section 3.1, regarding the traction and resistance forces in soft soil. Thus we have 13 discrete states in the entire contact model, see figure 10.

Rigid Surface	Soft Surface	Stone
<ul style="list-style-type: none"> • No Contact • Contact • Rolling • Sliding 	<ul style="list-style-type: none"> • No Contact • Contact • Stuck/Roll • Slip Accelerate • Slip Break 	<ul style="list-style-type: none"> • No Collision • Collision • Rolling • Sliding

Figure 10. Discrete states present in the contact models for each wheel.

4 Simulation issues

At this point we obtained a hybrid contact model with: iterative computational procedures (normal pressure curve shaping, triangle mesh integration and collision detection); additional states for simulation purposes (compliance systems); discrete state event transitions. The most important impacts in the simulation results are related with its quality and velocity; they are discussed in the next two subsections.

4.1 Quality of the simulation results (robustness and solvability issues)

Robustness in the sense of simulation model's solvability is not an easy task, since there are parameters (e.g. velocity constants, stiffness and damping parameters) inherent to the simulation model itself. It means that the simulation parameters should be independent of the design parameters, or at least known functions of them. Sadly this is not the case in our model, maybe it can be approached analytically in future versions of the model.

It is important to keep in mind that our simulation model must be capable of solving the system of equations despite the variation of the design parameters (in distinct simulations). There are some sets of design parameters capable of inducing unsolvable iterations which may halt the

simulation. Inside our optimization procedure this is not a desired behavior because it can also halt the optimization run. To avoid the latter we assign a high value to the related objective function, it keeps the optimization procedure away from that undesired region, since it becomes a local maximum.

Currently we adopted the most immediate solution in order to proceed with our main application, numerical optimization and sensitivity analysis, which require intensive parameter variation. Our current solution is: choose one set of simulation parameters capable of dealing with a broad range of design parameters' variation. However, it can hide a minimum of an objective function and obligate the optimization algorithm to keep distance from a 'good design parameter set' because it is not a 'good simulation parameter set'. The idea is illustrated by figure 11: in figure 11a is a simple objective function immersed on a negative gradient field pointing inward (to the minimum of the function); figure 11b shows the same objective function damaged by the local maximum created exactly on the minimum of the function, the negative gradient vectors are pointing outward (away from the desired minimum).

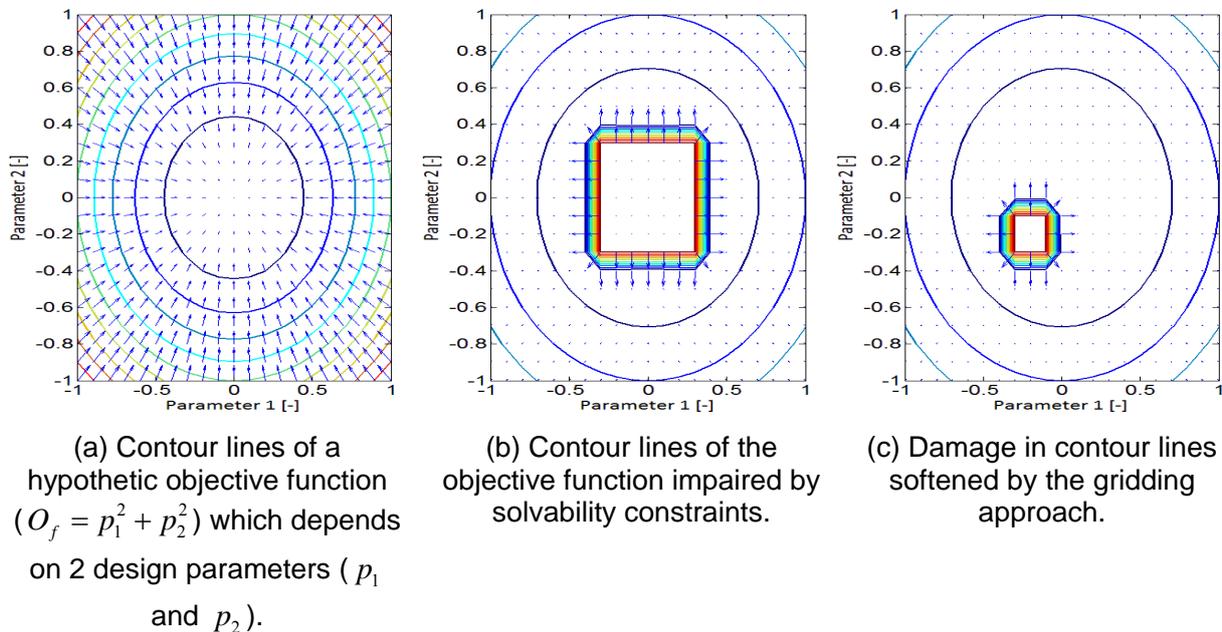


Figure 11. Damage of the objective function in the contour lines.

Although, the amount and area occupied by those peaks can be reduced by using a “gridding” procedure. This procedure consists of searching a suitable simulation parameter set to attend extremes of a bounded solution space. The result is like that of figure 11c, there are still restrictions but they can be softened.

4.2 Velocity of the simulation runs (stiffness and chattering issues)

Numerical stiffness is an intrinsic characteristic of our simulation model; this is due to the differences in the involved masses, stiffness and damping of the components through the model. These characteristics were employed to implement the contact models in a physically meaningful way, their values were determined to deal with the robustness in the solvability sense as explained in the previous section. Although, we use suitable numerical integration tools (LSODAR, numerical integrator with root-finding) to handle this problem and achieve acceptable computational savings.

The whole model embodies several discrete states for each wheel composing a planetary vehicle (which contains usually 4 or 6 wheels), the thirteen discrete states for each wheel raise from rigid soil, soft soil and stone collision situations (see figure 10). These discrete states express switching between trajectories of continuous states of the whole Differential Algebraic Equation's system. Additionally, we experienced two main difficulties in handling such discrete states; they are illustrated in figure 12.

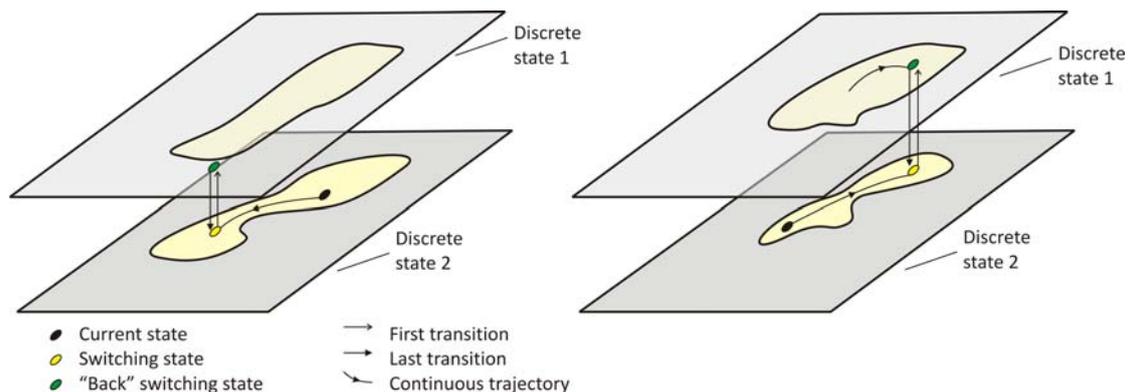
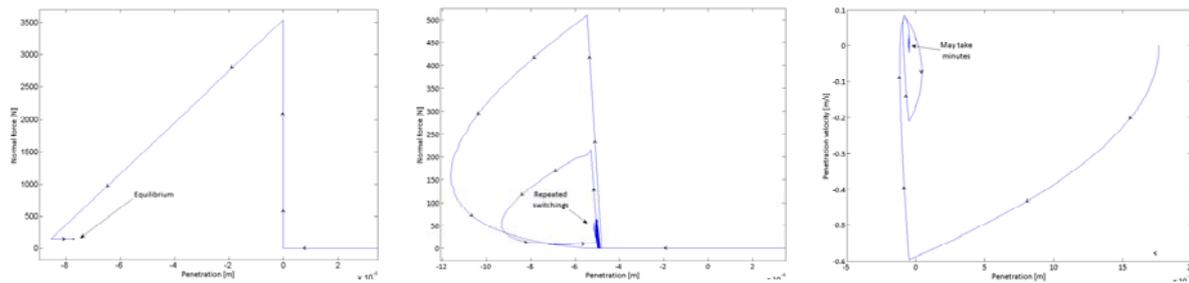


Figure 12. Problems with switching between two discrete states. Representing state-space trajectories constrained by its respective state domains.

The first problem is that of figure 12-left, there is a transition to the new discrete state and it switches immediately back to the previous one. It indicates an inconsistency; the root-finder is not capable of handle this event and the simulation stops suddenly. An unsolvable iteration may occur in the case of rigid surface driving change between rolling and sliding modes for several wheels and reaching one of the unsolvable configurations detailed in (Trinkle et al., 1997). This is rare but still possible and occurs for some configurations of vehicles with certain design parameters, we do not try any rigorous attempt to explain and overcome this problem, only detect. Previously, in subsection 4.2 some words were spent about our strategy to overcome this problem.

In figure 12-right is the known chattering behavior, where the simulation requires a large amount of discrete state transitions in a relatively small amount of time. The main difference between the first problem (figure 12a) and the second problem (figure 12b) is that the first case switches immediately back, the second case switches back after a little amount of time. It deteriorates the speed of the simulation, sometimes this deterioration is prohibitive to go onward in the numerical integration. A proper tuning of the simulation parameters eases this inopportune behavior. In figure 13 one can see an example of chattering (figure 13b) compared with a case without chattering, figure 13a. This example was extracted from one simulation of a single wheel falling on a stone, the normal force can be smoothly calculated (figure 13a) with well tuned simulation parameters, i.e. stiffness, damping, power factor and tolerances of the wheel-stone contact model. However, the result is not so good with poorly tuned simulation parameters (see figure 13b).



(a) Well tuned case (desired).

(b) Unsuitably tuned case (spurious behavior).

Figure 13. Trajectories showing desired and spurious behavior due to certain simulation parameters chosen.

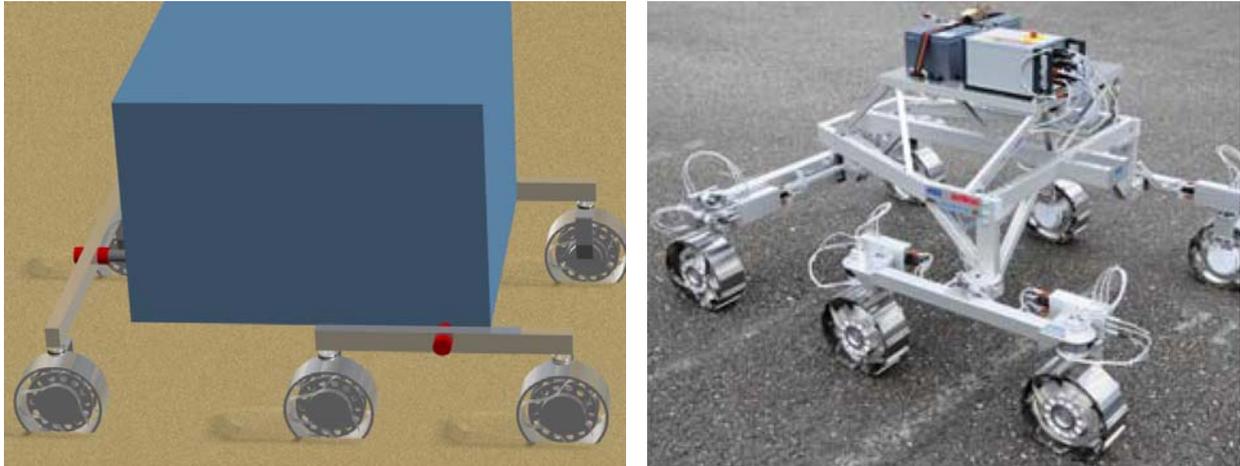
This example is particularly applicable to the case of wheel-stone contact (where the problem is dominant). That is because of the absence of jerk evaluation in its respective state machine to accurately accommodate the normal force oscillation after an impact. It means that we handle chattering by suitably tuning the simulation parameters of the wheel-stone contact normal computation.

5 Validation and application

After modeling and implementation of the model in the simulation environment the next step is the validation of the model against experimental data to properly proceed towards the main application. The procedure and results of partial validation of the model and a hint of the main application are given in the next two subsections.

5.1 Partial validation of the simulation model

The contact model is rather complex to be completely validated on a few tests, although the most important contact model is that of soft soil. A planetary rover will drive the most part of its life on such condition, nevertheless sinkage, slippage and traction prediction are key factors to the prediction of performance of a rover. Based on this assumption we did a partial validation of the soft soil contact model in only one testing condition. The experiment to evaluate the model is the drawbar pull experiment, it was conducted by the ETH Zurich (Zurich University, Switzerland) at the facilities of the Swiss company Oerlikon Space (Michaud et al., 2008). Only the drawbar pull versus slip ratio were compared, states of the joints were not taken into account for this comparison due to lack of information at that time. Figure 14a shows the visualization of our simulation model driving on sand, and figure 14b shows the breadboard model used in the experiments.



(a) Simulation model on soft soil.

(b) Breadboard model on rigid soil.

Figure 14. Simulation model and breadboard model (ExoMars phase B1).

The soil parameter values used for the soil of the experiment are in table 1.

Table 1. Bekker parameters used for certain soil simulant of type DLR-D2, (Schäfer et al., 2010).

Parameters and physical dimensions			Values
Cohesive modulus	k_c	$(\text{N}/\text{m}^{n+1})$	-6.67×10^5
Frictional modulus	k_ϕ	$(\text{N}/\text{m}^{n+2})$	1.92×10^8
Exponent sinkage	n	(-)	1.4
Cohesion	c	(Pa)	13.0
Internal friction	ϕ	(deg)	32

The drawbar pull curve versus slippage values with the translational velocity of 0.0108m/s is shown in figure 15. The time-domain measurements are strongly compressed and put on the graph to give an idea of the drawbar pull variation of the measurements; there are frequency components due to the vibration of the experimental payload and grousers' contribution. Our simulation model is not considering wheels equipped with grousers. Note that the simulation model with DLR-D2 parameters gave us a very good correlation thanks to the shaping approach of the pressure distribution on the contact patch.

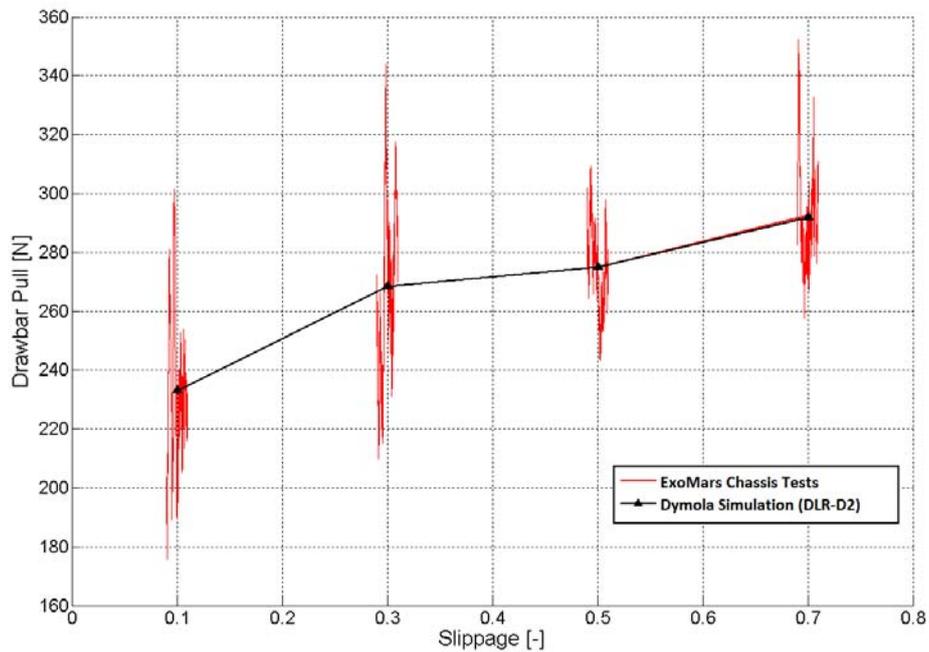


Figure 15. Drawbar pull as a function of slip: comparison between measurement data (*ExoMars Chassis Tests*) and simulation (*Dymola Simulation*).

With our simulation model is possible to achieve even the experimental behavior slightly different from the classical exponential behavior known from (Bekker, 1956).

5.2 Main application – rover design optimization

The main purpose of our work is to develop an efficient optimization tool to support rover design. It is based on dynamic simulation, optimization and clear visualization of the achieved results. Figure 16 shows the software packages used to implement such tool and the data flow between them.

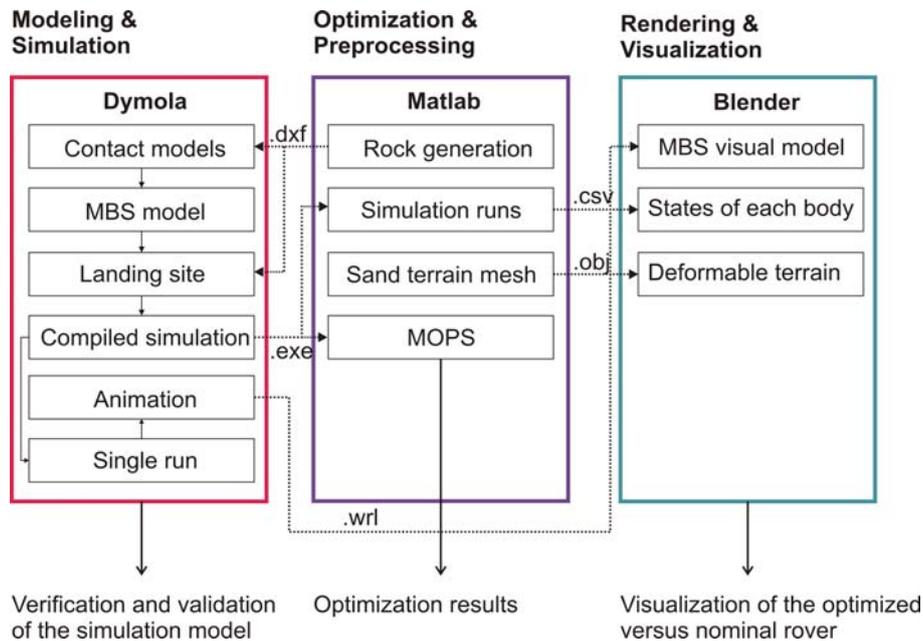


Figure 16 – Data flow in the rover optimization tool among used software packages.

Contact modeling is conducted in Dymola through a modeling language called modelica and external calls to some complex calculations (e.g. collision detection, mesh discretized calculations and Chaikin curve generation) coded in C and C++ to increase speed of the simulation. The Multibody Simulation (MBS) structure is developed with standard tools of Dymola, the landing site is composed of stones' models generated in Matlab and exported as `.DXF` files, sand and relief pattern are both described in Dymola environment. The simulation model is compiled and the executable file can run inside or outside Dymola. On the one hand when it runs inside, it provides results for verification and validation of a single structure against project specifications and experimental test respectively. This is performed through rough animation and numerical simulation data.

On the other hand, the compiled simulation model can be used by MOPS (an in-house optimization tool of DLR) to change the parameters and compute user-defined objective functions and find an optimal instance of a planetary rover. The single simulation results can be still visualized with a higher quality after proper texturing, lightning and animation of sand movement through Blender. This software receives the rough animation model generated by Dymola in VRML and integrates sand movement (generated as `.OBJ` files in Matlab), detailed drawings of mechanical connections and proper textures.

The tool gives three distinct possibilities of usage: modeling improvement, rover design optimization and higher quality rover animation of a physical simulation. The three general outputs shown in figure 16 constitute the main features of our rover design assistance tool.

6 Conclusion and outlook

Modeling is always an ongoing task, primarily because of the simulated system's needs. In the present case the performed validation has shown satisfactory results which explicit the accuracy achievable by our model. Despite the good correlation achieved in the drawbar pull experiment, other tests must be employed to validate the entire contact model driving over uneven terrain, rigid soil and obstacles. The current model is capable of dealing with contact between cylindrical

rigid wheels and rigid surfaces with Coulomb friction behavior or Pacejka's "magic formula"-like behavior; complexly-shaped convex/non-convex rigid bodies with the same force computation variants as in the case of simple rigid surfaces; and soft surface determined by Bekker parameters. Note that further improvements are needed to deal with flexible wheels and grousers' contribution, which will probably be the actual case of the ExoMars rover. The model allows such additions; they will be incorporated soon in the contact models. Collision detection with flexible wheels would be a difficult task, but this is supported by the employed collision detection library (SOLID 3.5.6).

The model do not need many adjustments during its verification, it means that the process was conducted in a consistent way and it is reliable to perform new validation efforts and optimization tasks. Since the very beginning of the project, our main purpose was optimization and not modeling. But optimization is not meaningful without a good model, and our modeling effort was *ipso facto* indispensable.

Acknowledgements

Financial support from the German Academic Exchange Service (Deutscher Akademischer Austausch Dienst - DAAD) is gratefully acknowledged.

References

- Barber, C. B., D.P. Dobkin, and H.T. Huhdanpaa. 1996. The Quickhull algorithm for convex hulls. *ACM Trans. on Mathematical Software*, 22(4):469-483.
- Bekker, M. G. 1956. *Theory of land locomotion*, Ann Arbor – The University of Michigan press.
- Chaikin, G. M. 1974. An Algorithm for High-Speed Curve Generation. *Computer Graphics and Image Processing*, 3: 346-349.
- Gibbesch, A., R. Krenn, F. Herrmann, B. Schäfer, B. Rebele, E. Allouis, T. Dietrich, Multibody system and contact simulation within the design development of planetary surface exploration systems. Manuscript submitted for publication. The 10th International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS 2010), Sapporo, Japan, 2010.
- Golombek, M.P. Rock statistics calculations for the MER landing sites. In *Landing Site Workshop*, 2002.
- Hegedus, E. 1962 A preliminary analysis of the force system acting on a rigid wheel. Technical report, U. S. Army Ordinance Tank-Automotive Command, Detroit 9, Michigan. Technical Report No. AD0401805.
- Hunt, K., and F. Crossley. 1975. Coefficient of restitution interpreted as damping in vibroimpact. *ASME Transactions Journal of Applied Mechanics*, 97: 440–445.
- Ishigami, G. 2008. Terramechanics-based Analysis and Control for Lunar/Planetary Exploration Robots, Ph.D. Thesis, Tohoku University, Japan.
- Kraus, P. R., A. Friedriksson, and V. Kumar. Modeling of frictional contacts for dynamic simulation. In *IROS 1997 Workshop on Dynamic Simulation: Methods and Applications*, September 1997.
- Michaud, S., M. Höpflinger, T. Thueer, C. Lee, A. Krebs, B. Despont, A. Gibbesch, and L. Richter. ExoMars locomotion system test campaign, 10th ESA Workshop on Advanced Space Technologies for Robotics and Automation (ASTRA 2008), ESA-ESTEC, Noordwijk, The Netherlands, 2008.

- Patel, N., R. Slade, and J. Clemmet. 2010. The ExoMars rover locomotion subsystem. *Journal of Terramechanics*, 47: 228-242.
- Schäfer, B., A. Gibbesch, R. Krenn, and B. Rebele. 2010. Planetary rover mobility simulation on soft and uneven terrain. *Vehicle System Dynamics*, 48: 149-169.
- Scharringhausen, M., D. Beermann, O. Krömer, and L. Richter. A wheel-soil interaction model for planetary applications. 11th European Regional Conference of the International Society for Terrain-Vehicle Systems, Bremen, October 5-8, 2009.
- Sohl, G., and A. Jain. Wheel-terrain contact modeling in the ROAMS planetary rover simulation. In *2005 ASME International Design Engineering Technical Conference*, Long Beach, California, September 2005.
- Trinkle, J. C., J.-S. Pang, S. Sudarsky, and G. Lo. 1997. On Dynamic Multi-Rigid-Body Contact Problems with Coulomb Friction. *ZAMM - Journal of Applied Mathematics and Mechanics / Zeitschrift für Angewandte Mathematik und Mechanik*, 4(77):267-279.
- Wong, J. Y. 2001. *Theory of Ground Vehicles*. John Wiley & Sons, 3rd edition.
- Yoshida, K., T. Watanabe, N. Mizuno, and G. Ishigami. Slip, Traction Control and Navigation of a Lunar Rover. In *Proceedings of the 7th International Symposium on Artificial Intelligence, Robotics and Automation in Space: iSAIRAS 2003*, NARA, Japan, May 19-23, 2003.
- Zimmer, D., and M. Otter. 2010. Real-time models for wheels and tyres in an object-oriented modeling framework. *Vehicle System Dynamics*, 48:189-216.