MODELING OF WHEEL-SOIL CONTACT FOR THE EDRES MOBILE ROBOT SIMULATOR

D. Lhomme-Desages[1], C. Grand[1], and M. Maurette[2]

[1]*ISIR, 18 Route du Panorama BP61 92265 Fontenay-Aux-Roses, France, {lhomme,grand}@isir.fr*
[2]*CNES, DTS/AE/SEA/ER, 18 Avenue E. Belin 31401 Toulouse cedex 4, France, maurette@zeus.cst.cnes.fr*

Abstract

The EDRES software, developed by the CNES of Toulouse, is a simulation tool for planetary mobile robots. It is intended to validate control algorithms and onboard software for future exploration missions, such as ExoMars. To enhance the relevance of the simulation, we have integrated a terramechanic contact model, in order to study the influence of slippage and sinkage on motion estimation and control, over various soils. A specific algorithm has been proposed, taking into account the architecture of the simulator.

Key words: rover, slippage, terramechanics.

1. INTRODUCTION

The CNES (the French space agency) has developed EDRES, a mobile robot simulation software for planetary exploration [6]. The simulator allows the development of algorithms for motion estimation by vision, locomotion control and autonomous navigation, avoiding the use of a real chassis, which would be costly in resources and time. It is currently developed in the context of the ESA ExoMars mission, as well as for EXOMADER and IARES prototypes.
In the current state of the simulator, the locomotion loop does not take into account the physicochemical nature of soil and the complex phenomena of wheel-soil interaction, which can cause skidding, slippage, or even immobilization of the vehicle. Modeling these phenomena is necessary for the realism of the simulation, for the design of effective locomotion control algorithms and, potentially, for automatic trajectory generation.
This paper presents a methodology for integration, in the EDRES simulator, of a wheel-soil contact model adapted to the specific nature of the terrain considered (loose and soft soil) and constraints related to the software architecture.
We describe the selected model and the corresponding assumptions. Then, we detail the proposed algorithm for calculating forces, sinkage and slippage of the wheels with respect to the ground.

2. EDRES SIMULATOR OVERVIEW

EDRES (*Environnement de Développement pour la Robotique d'Exploration Spatiale*) is a set of algorithms, tools and applications that cover a large panel of functions used to generate and execute autonomous motions of planetary rovers. The EDRES rover simulator is capable of graphically displaying any digital elevation model (DEM) and applying a texture to it; simulating the kinematics of the IARES and EXOMADER rovers chassis; simulating the stereovision bench used for the autonomous navigation; moving the rover over the terrain satisfying kinematic constraints; simulating the rover localization subsystem, comparing it to real position et heading.
It is based on the principle of purely geometric positioning, thus producing real-time motion according to the requests of the human operator, with any DEM based on real topographical measurements. The robot chassis is simulated at kinematic level, with the addition of hardware specifications of sensors and actuators. Dynamics is not included.
Fig. 1 shows the overall functional architecture of EDRES. It is composed of five parallel processes: the graphical interface (MMI), the locomotion control loop, the perception server, the localization server, the locomotion server. These last three modules are interfaces between the simulator and testing software or onboard applications such as visual odometry. The locomotion loop updates the robot position according to data coming from the locomotion server or from the human user. At each step of motion, a positioning function computes the new attitude of the robot.
A functions library implements rover-specific functions such as displaying the robot, calculating geometric and kinematic models, etc.
The kinematic simulation does not allow to handle certain important motion behaviors, such as skidding and slipping. This hinders the development of locomotion control, autonomous navigation and visual motion estimation. This drawback can

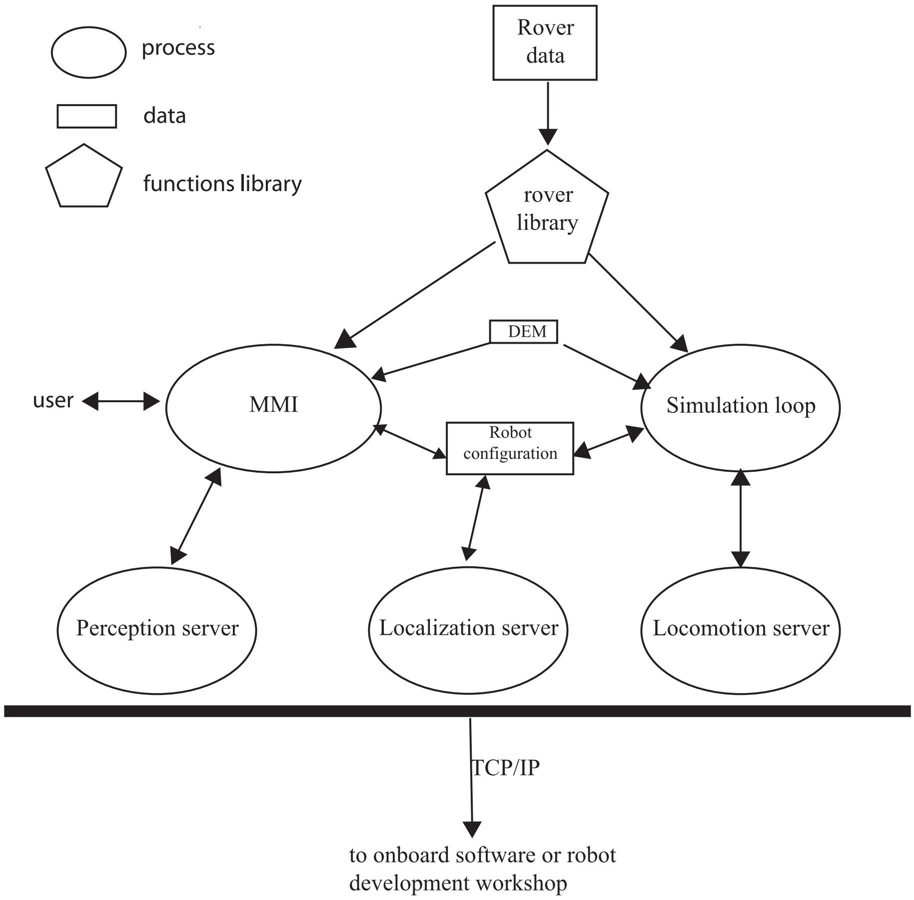

Figure 1. EDRES architecture

be solved by introducing a model that describes the kinetostatic behavior of robot wheels running over the soil. The selected model is presented in the following section.

3. WHEEL-SOIL CONTACT MODEL

Several modeling frameworks can be used to calculate the efforts involved in the wheel-soil interaction process. We use an extended version of the terramechanic model introduced by Bekker ([1],[7]). We assume that the entire wheel is very stiff compared to the ground and we can consider the wheel rigid (Fig. 2).
All wheels are supposed to be identical perfect smooth cylinders without treads. Soil deformation is purely plastic. The terrain is considered locally horizontal, flat and homogeneous. The application point of the rolling resistance is supposed to be the center of the wheel.
Considering the total mass of the robot, gravity and wheel prints, we can assume that the sinkage value is small compared to the wheel radius, so that the contact angle θ_c is small and all quantities can be reduced to first order terms. Also, we suppose that the normal force direction meets the center of the wheel. The model is valid only in stationary state and does not depend on time.

3.1. Normal Force and Sinkage

According to Bekker theory [1], the normal force depends on the sinkage z through:

$$F_n = \frac{1}{3}\left[w_w\left(\frac{k_c}{r}+k_\phi\right)(3-n)\sqrt{D}z^{\frac{2n+1}{2}}\right] \tag{1}$$

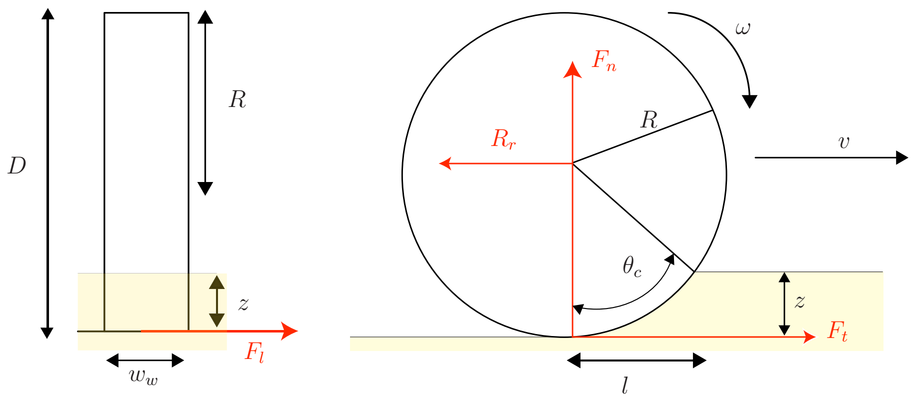

Figure 2. Model of a rigid wheel

where k_c, k_ϕ and n are soil parameters. w_w is the wheel width. $r = \min(w_w, l)$, l being the length of the contact patch.

3.2. Slippage

Note v the velocity of the center of the wheel and ω the angular velocity. In this model, the traction force depends on the slip rate s, which we define as:

$$s = \begin{cases} 1 - \frac{v}{\omega.R} & \text{if} \quad R\omega \geq v \\ 1 - \frac{\omega.R}{v} & \text{if} \quad R\omega < v \end{cases} \tag{2}$$

for $v > 0$ and $\omega > 0$. This definition can be extended to every $(v, \omega) \in \Re^2$ [5], as it is shown in Fig. 3, in order to use it in simulation. In the following, for simplifying purpose, we expose the case of traction. The braking case is similar.

3.3. Net Longitudinal Force and Rolling Resistance

The net longitudinal force DP (drawbar pull) is the difference between the raw traction force (due to shearing) and the rolling resistance:

$$DP(s) = F_t(s) - R_r \tag{3}$$

The rolling resistance is assumed to be mainly caused by soil compaction, which allows to use the expression:

$$R_r = w_w \frac{z^{n+1}}{n+1} \left(\frac{k_c}{r} + k_\phi \right) \tag{4}$$

3.4. Shearing Force

Consider a wheel moving along the direction $\mathbf{x}$, different from the longitudinal axis $\mathbf{t}$ (Fig. 4). Then a lateral slippage occurs in the wheel-soil relative motion. v_l is the lateral velocity.
The contact force projected in the tangential plan is:

$$\mathbf{f}_t = DP(s)\mathbf{t} - F_l\,\mathbf{l} \tag{5}$$

with $DP(s)$ being the net traction force as seen previously and F_l the total lateral force.
We expose the case of traction, assuming $v > 0$ and $\omega > 0$ then $s = 1 - \frac{v}{R\omega}$. Similar expressions can be obtained for other cases.
The longitudinal shearing force is given by (see [3] [4]):

$$F_t = F_m \left[1 - \frac{K \cos \xi}{sl} \left(1 - e^{-\frac{sl}{K \cos \xi}} \right) \right] \cos \xi \tag{6}$$

where: $F_m = l w_w c + F_n \tan \phi$. c, ϕ and K are soil parameters.

The lateral force is the sum of two contributions:

$$F_l = F_l^s + R_l^b \tag{7}$$

where F_l^s is the lateral shearing resistance and R_l^b the lateral bulldozing resistance. The lateral shearing resistance is:

$$F_l^s = F_m \left[1 - \frac{K \cos\xi}{sl} \left(1 - e^{-\frac{sl}{K\cos\xi}} \right) \right] \sin\xi \tag{8}$$

Note that the zero slip case ($s = 0$) leads to a singularity and has to be handled separately. In this case, we have:

$$F_l^s = F_m \quad \text{and} \quad F_t = 0 \tag{9}$$

3.5. Lateral Bulldozing Resistance

The lateral bulldozing resistance is due to the displacement of soil material by the lateral side of the wheel. Our approach is based on a quasi-static computation of the forces and the use of the Mohr-Coulomb failure criterion. We consider only the steady state. More details can be found in [4].

$$R_l^b = K_p c R^2 \theta_c^3 + \frac{1}{6} \rho g R^3 {K_p}^2 \theta_c^3 \left(1 + \frac{\theta_c^2}{2} \right) \tag{10}$$

where

$$K_p = \sqrt{(1 + \sin\phi)/(1 - \sin\phi)} \tag{11}$$

g is the gravity and ρ the soil density.

4. PROPOSED ALGORITHM

Our algorithm is based on the data available through the simulator. The vehicle has 6 identical cylindrical wheels of radius R and width w_w. Its position and orientation in the DEM is measured with respect to the mission frame.
The motion of the vehicle is supposed to be quasi-static and that its dynamics is negligible, which is a weak constraint given the operational robot velocity (some centimeters per second).
Fig. 5 shows the principle of the algorithm (for traction).

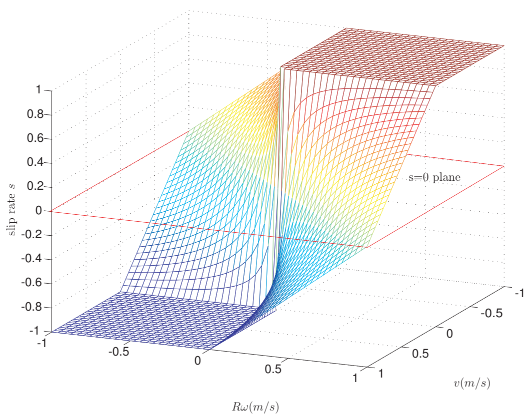

Figure 3. Slip rate depending on the velocities at the wheel-soil interface

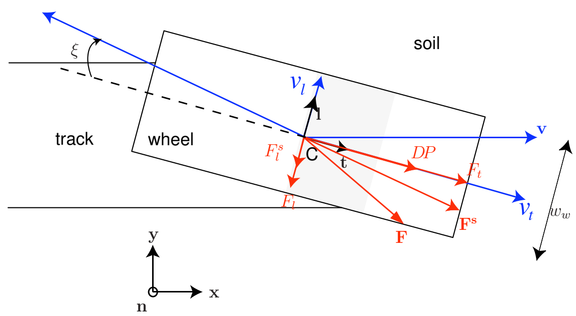

Figure 4. Model of the wheel-soil interface in the tangent plane

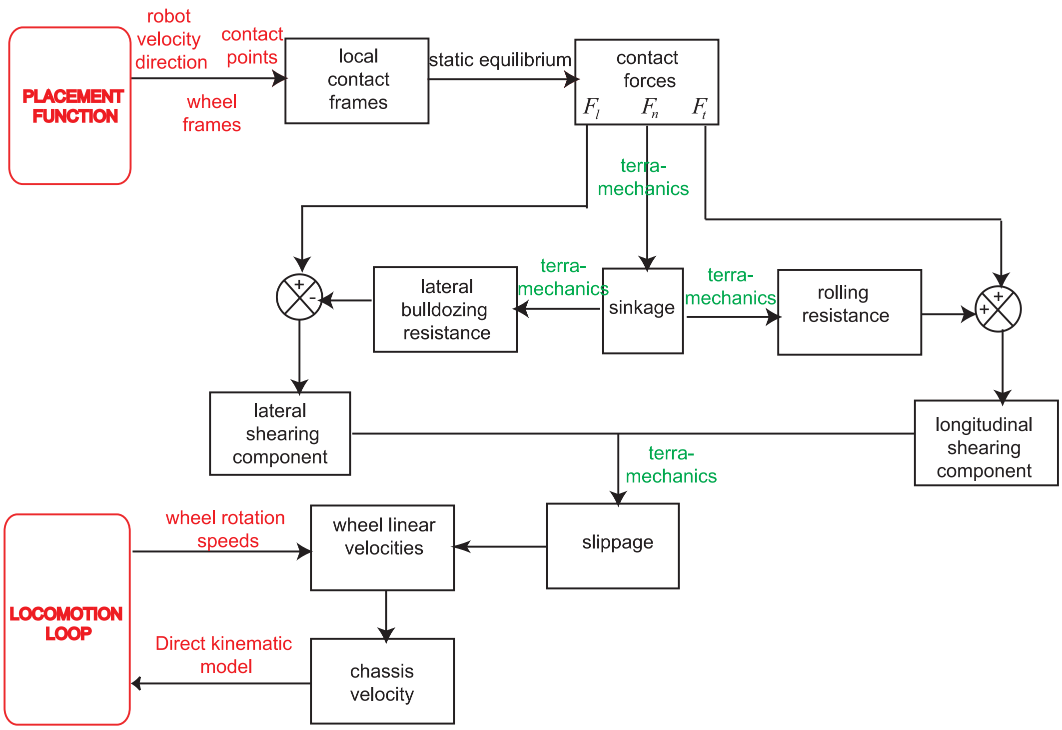

Figure 5. Algorithm overview

4.1. Contact Forces

We suppose that the geometric placement of the robot is the same as the one obtained through integral tridimensional dynamic simulation. The simulator supplies a contact point P_i for each wheel i ($i = 1..6$). A contact point, supposed to be unique, is the point of the wheel rim which is nearest of the soil tangent plane. Then it still exists when the wheel sinks into the soil. These contact points are computed at each elementary move of the robot on the planetary surface.
First, the algorithm determines the 6 contact frames ($\mathbf{n}_i$, $\mathbf{t}_i$, $\mathbf{l}_i$), i.e. normal, longitudinal and lateral unit vectors respectively. The contact forces are determined by the resolution of the static equilibrium assuming the robot is a rigid body in very slow motion:

$$\mathbf{A}\mathbf{f} = \mathbf{w}_g \tag{12}$$

where $\mathbf{f}$ collects the 18 unknowns expressed in the contact frames:

$$\mathbf{f} = (F_{n1}, F_{t1}, F_{l1}, F_{n2}, F_{t2}, F_{l2}, ..., F_{n6}, F_{t6}, F_{l6})^T \tag{13}$$

and $\mathbf{w}_g$ is the 6x1 wrench vector corresponding to rover weight, expressed at the robot CoM G (M is the total vehicle mass):

$$\mathbf{w}_g = (0, 0, Mg, 0, 0, 0)^T \tag{14}$$

$\mathbf{A}$ is defined as:

$$\mathbf{A} = \begin{pmatrix} \begin{bmatrix} \mathbf{n}_1 \end{bmatrix} & \begin{bmatrix} \mathbf{t}_1 \end{bmatrix} & \begin{bmatrix} \mathbf{l}_1 \end{bmatrix} & ... & \begin{bmatrix} \mathbf{n}_6 \end{bmatrix} & \begin{bmatrix} \mathbf{t}_6 \end{bmatrix} & \begin{bmatrix} \mathbf{l}_6 \end{bmatrix} \\ \begin{bmatrix} \mathbf{p}_1 \times \mathbf{n}_1 \end{bmatrix} & \begin{bmatrix} \mathbf{p}_1 \times \mathbf{t}_1 \end{bmatrix} & \begin{bmatrix} \mathbf{p}_1 \times \mathbf{l}_1 \end{bmatrix} & ... & \begin{bmatrix} \mathbf{p}_6 \times \mathbf{n}_6 \end{bmatrix} & \begin{bmatrix} \mathbf{p}_6 \times \mathbf{t}_6 \end{bmatrix} & \begin{bmatrix} \mathbf{p}_6 \times \mathbf{l}_6 \end{bmatrix} \end{pmatrix} \tag{15}$$

where $\mathbf{p}_i$ is the position of the contact point with respect to the rover frame ($\mathbf{p}_i = \vec{GP_i}$). G is computed in real-time depending on the vehicle configuration. Eq. (12) has an infinity of solutions, since the size of $\mathbf{A}$ is $6{\times}18$. It can be solved using the Penrose pseudoinverse matrix:

$$\mathbf{f} = \mathbf{A}^+ \mathbf{w}_g \tag{16}$$

leading to the solution such as $||\mathbf{f}||_2$ is minimized.

4.2. Slip Rates

We draw the sinkage z_i from the normal force F_{ni} (Eq. (1)):

$$z_i = \left[\frac{\sqrt{2}\, F_{ni}\, \Gamma(n + \frac{3}{2})}{w_w(\frac{k_c}{l_{wi}} + k_\phi)\sqrt{R}\, \Gamma(n+1)\sqrt{\pi}} \right]^{\frac{2}{2n+1}} \tag{17}$$

where Γ is the Euler function and $l_{wi} = \min(w_w, l_{ci})$, with l_{ci} being the contact length:

$$l_{ci} = R \sin\left(\arccos\left(1 - \frac{z_i}{R}\right)\right) \tag{18}$$

Then, the rolling resistance can be computed from z_i using Eq. (4). The raw traction force is the sum of the net force and this rolling resistance:

$$F^s_{ti} = sign(F_{ti})\,(|F_{ti}| + R_{ri}) \tag{19}$$

The sign is given by the sign of the net traction force.
On the other hand, the lateral bulldozing resistance is computed using Eq. (10) for each wheel. The contact angle θ_{ci} is:

$$\theta_{ci} = l_{wi}/R \tag{20}$$

We obtain the lateral raw resistance, i.e. produced by shearing:

$$F^s_{li} = sign\,(F_{li}) \max(|F_{li}| - R^b_{li}, 0) \tag{21}$$

The shearing components calculated in (19) and (21) can now be combined to compute the slip rates s_i, using (6) and (8). The shearing angle ξ is:

$$\xi_i = \operatorname{arctan2}(F^s_{li}, F^s_{ti}) \tag{22}$$

where $\operatorname{arctan2}(x, y)$ gives the four-quadrant inverse tangent of y/x.
We have to solve the equation in s_i:

$$F^s_i = \sqrt{F^s_{li} + F^s_{ti}} = F_m \left[1 - \frac{K \cos\xi_i}{s_i l_i}\left(1 - e^{-\frac{s_i l_i}{K \cos\xi_i}}\right)\right] \tag{23}$$

If F^s_i is greater than the maximal value admissible by the soil then the slip is equal to its extreme value (± 1 depending on the sign of the force). Otherwise, the equation can be solved analytically using Lambert function W [2].

4.3. Vehicle Velocity Computation

Rotation velocities ω_i are inputs of the algorithm and supplied by the simulated motor control. To simulate slippage, we modify the linear platform velocity (linear velocity of the vehicle's CoM), from computed slip rates.
The 6D vehicle velocity $(\mathbf{v}, \boldsymbol{\omega})$ is the average of every 6D velocities induced by wheel hubs' linear velocities in G. Therefore, we modify each linear velocity $\mathbf{v}_i$ which depends on ω_i and s_i, through a function $fs(s, \omega)$ derived from (2) and plotted on Fig. 6. fs comprises every cases according to the signs of s_i and ω_i.
Note that this function has several singularities, for which we are compelled to put arbitrary values. For example, for $\omega = 0$, every $v > 0$ implies $s = -1$. Hence, the case $\omega = 0$ and $0 < |s| < 1$ is theoretically impossible. However, it may happen during the simulation, as result of the computation of longitudinal forces. This inconsistence is solved by writing $fs(-1, 0) = a$ ($a > 0$) so that the behavior remains continuous.

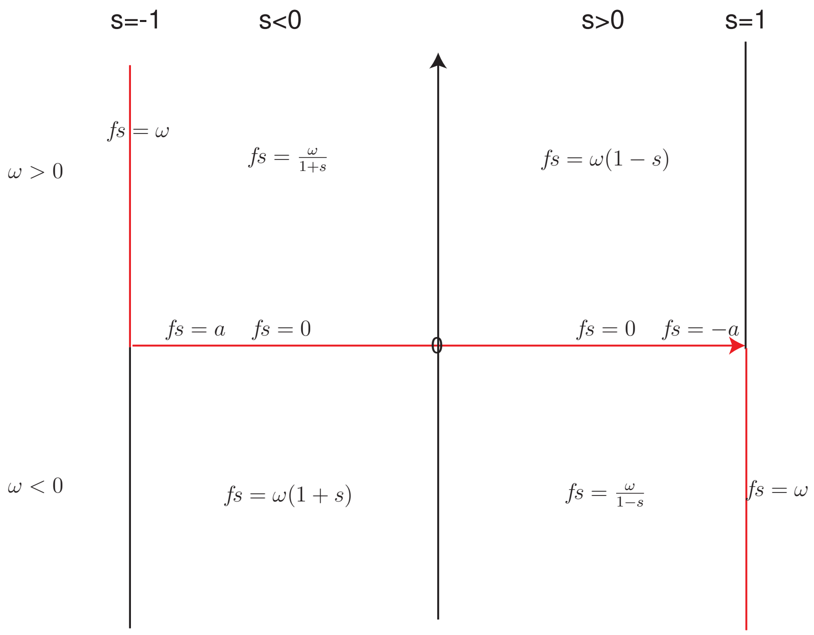

Figure 6. $fs(s, \omega)$

5. IMPLEMENTATION AND RESULTS

In order to prove the feasibility and usefulness of our algorithm, it has been partially implemented into the EDRES simulator. Only the IARES robot is used. We use a Pentium 4 3Ghz computer with 523Mb RAM and a Nvidia GeForce 7600 GT GPU. The application runs under Linux operating system.
Wheel-soil interaction parameters can be selected through a graphical menu. Terramechanic data such as sinkages and slip rates are displayed.
Some preliminary tests have been carried out with soil parameters of Table 1. Two simulation sessions are illustrated on Fig. 7 and 8. Contact forces are displayed in the form of arrows of varying lengths. One can see the soil menu window where interaction parameters, sinkages and slip rates are displayed. The influence of the contact model can be evaluated using the remote control mode (the user controls the speed and heading of the rover).
From a qualitative point of view, we obtain a significant increase of the realism of the simulation. The vehicle slips more as the terrain is sloping or as the soil, characterized by the interaction parameters, is slippery.
On a lateral slope, we note as expected a lateral skidding (see Fig. 7).
Fig. 8 presents a locking situation. The traction generated by the wheels, for this particular set of parameters, is insufficient to make the robot pass over the obstacle. This example shows the interest of the addition of the wheel-soil model into the EDRES simulator.

6. CONCLUSION

A quasi-static wheel-soil interaction model for loose and non cohesive soils has been selected for the enhancement of the CNES rover simulation tool. We have proposed an algorithm to compute the slippage of a wheeled robot according to the

Table 1. Test parameters (dry sand)

parameter	value	unit
c	1150	Pa
ϕ	28	deg
ρ	1600	kg/m^3
n	0.9	-
k_ϕ	505800	N/m^{n+2}
k_c	6940	N/m^{n+1}
K	1.2	cm
g	9.81	m/s^2

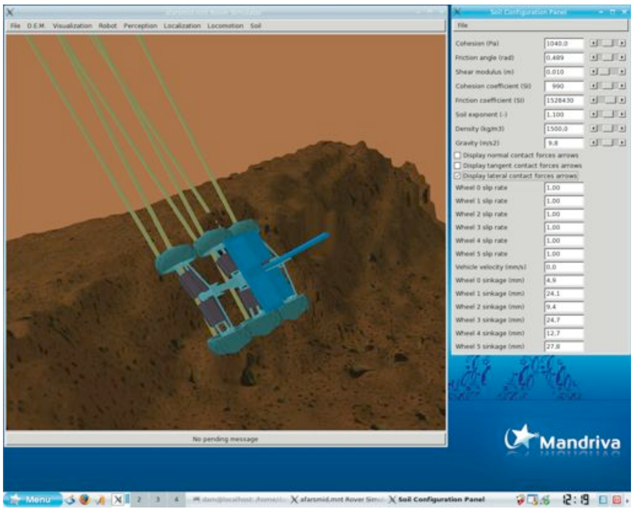

Figure 7. Lateral skidding of the robot

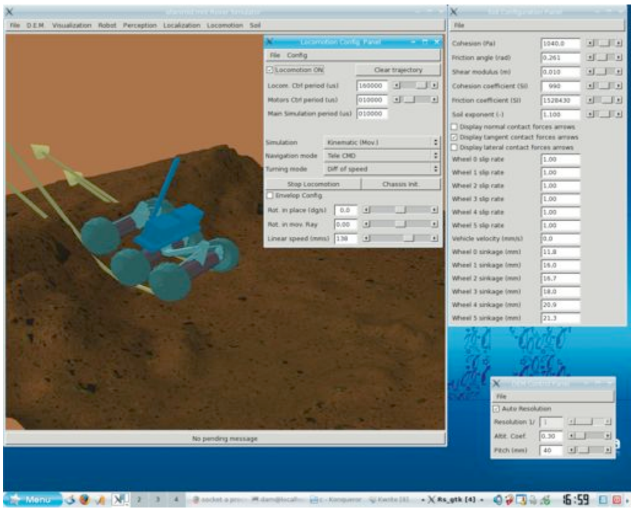

Figure 8. Blocking situation of the vehicle

physical properties of the soil and to the existing software architecture. Then, this algorithm, based on the terramechanic model, has been partially implemented (for the IARES rover) in order to show the feasibility of the total implementation. From a static calculation of the forces equilibrium, we are able to compute the tangential slip velocity for each wheel with respect to the ground. Thus, wheel rotation velocities being given, we can calculate a platform velocity for the robot that takes into account the slippage. The repartition of mass due to the robot configuration is also considered.

The physical modeling is based on several simplifying hypotheses, particularly on the contact geometry. Furthermore, the software architecture itself implies some constraints on the choice of the model, the algorithm and its implementation. This leads to a poorly reliable result from a quantitative point of view.

Thus, generally, linear velocities of wheel centers are not consistent. The global computed velocity of the vehicle does not allow to recalculate the slip rates through the inverse kinematic model. Net forces resulting from the static equilibrium are not actually applied to the system. Consequently, the proposed algorithm must be viewed fundamentally from a qualitative point of view.

A comparison with measurements using the IARES rover is still to be done in order to estimate precisely the parameters of the interaction model and quantify the degree of realism obtained.

Several extensions may be conceivable. It is possible to make a physical soil properties map (as for the DEM) to model a heterogeneous soil. The definition of the tridimensional wheel-soil contact area can be improved, in order to refine the sinkage computation.

However, the algorithm presented in this paper is a basis to study the influence of the physical nature of terrain on robot control and locomotion. As an exemple, it is possible today to test visual odometry in presence of slipping.

REFERENCES

[1] M.G. Bekker, *Introduction to Terrains-Vehicles Systems*, The University of Michigan Press, 1969.

[2] R.M. Corless, G.H. Gonnet, D.E.G. Hare, D.J. Jeffrey and D.E. Knuth, “On the Lambert W Function,” *Advances in Computational Mathematics*, Vol. 5, pp. 329-359, 1996.

[3] A. Grecenko, “Some applications of the slip and drift theory of the wheel,” in *International Conference of Society of Terrain Vehicle Systems*, Detroit, Michigan, USA, 1975.

[4] D. Lhomme-Desages, “Commande d’un robot mobile rapide roues non directionnelles sur sol naturel,” PhD Thesis, Pierre et Marie Curie University, Paris, April 2008.

[5] D. Lhomme-Desages, C. Grand and J.C. Guinot ,“Trajectory Control of a four-wheel Skid-steering Vehicle over soft Terrain using a Physical Interaction Model,” IEEE ICRA, Rome, April 2007.

[6] M. Maurette and L. Rastel, “Planetary rover simulation and operation,” ASTRA, November 2002, Noordwijk, The Netherlands.

[7] J.Y. Wong, *Theory of ground vehicles*, Wiley-Interscience, 3^{rd} ed., 2001.