

Wheel Torque Control in Rough Terrain - Modeling and Simulation

Pierre Lamon and Roland Siegwart

Autonomous Systems Lab

Ecole Polytechnique Fédérale de Lausanne (EPFL)

pierre.lamon@epfl.ch, roland.siegwart@epfl.ch

Abstract - This paper presents a method for wheel-ground contact angle measurement and a traction control strategy minimizing slip in rough terrain. The slip minimization algorithm has been tested and compared with a standard speed control in simulation, which allows to verify the validity of the assumptions taken during the modeling phase. The simulations show clearly the advantage of torque control versus speed control. Furthermore, the proposed method has the advantage to avoid relying on complex wheel-soil interaction models, whose parameters are generally unknown in challenging terrains.

Keywords - traction control, rough terrain, slip minimization

I. INTRODUCTION

For wheeled rough terrain rovers, the motion optimization is somewhat related to minimizing slip. Minimizing wheel slip not only limits odometric error but also increases the robot's climbing performance. In order to fulfill this goal, several methods have been developed.

Methods derived from the well known Anti-lock Braking System (ABS) can be used for rough terrain rovers. This technique, essentially developed for the car industry, uses the information of wheel slip to correct individual wheel speed, and thus allows to limit slip. A similar method has been implemented for the Nasa FIDO rover [1]. It is based on a velocity synchronization algorithm which minimizes the effect of the wheels "fighting" one another. The first step consists in detecting which of the wheels are deviating significantly from the nominal velocity profile. Then a voting scheme is used to compute the required velocity set point change for each individual wheel.

Such methods account neither for the kinematic nor for the physical model of the rover. They are thus expected to have some limitation in very challenging terrains (3D). Furthermore, wheel velocities are updated when slip already occurred and the system reacts only with a certain delay. In order to limit the reaction time and minimize the probability of slip, control techniques based on rover physical models have been developed. [2] presents a control methodology which can minimize power consumption in relatively flat terrain and maximize traction in highly uneven terrains. The models and results have been developed for the 2D case only. An extension to 3D, considering the traction maximization, is proposed in [3]. (Note that traction maximization is similar to slip minimization)

There are several limitations to the physics-based methods that prevent the controller to be implemented on a real rover. The main obstacle is that the wheel-ground interaction mod-

els are considered to be known. In practice, the model parameters are generally difficult to estimate and are valid only for a specific type of soil and condition. [4] proposes a method for estimating the soil parameters as the robot moves, but it is limited to a rigid wheel traveling through a deformable terrain. In practice, these models are difficult to use because the rover wheels are subject to roll on different kinds of terrain, whose parameters can change quickly.

Another key variable for traction optimization algorithms is the contact angles between the wheels and the ground. These angles can be estimated using indirect methods such as presented in [5] and [6]. They are based on wheel encoders, inclinometer and joint angle sensors. The accuracy of the estimation strongly depends on wheel slip and terrain profile. In particular, no estimation can be computed when the rover is still and poor results are obtained in slowly changing terrain profiles. Furthermore, such methods suffer from the chicken and egg paradox. Bad wheel-ground contact angle estimations lead to unadapted motor commands, which implies wheel slip and bad angles. Thus, direct measurement is required in order to be independent of the terrain profile and characteristics and to guarantee the system stability.

This paper proposes concrete solutions for avoiding problems related to model parameters and wheel-ground contact angle estimation. Together with 3D simulations, this allows reducing the gap between the theory and the real application.

Section 2 presents the rover used in the framework of this research and a tactile wheel design. A short review of our torque control method minimizing wheel slip is presented in section 3 and 4. Section 5 contains the proposed speed-torque controller, allowing to get rid of terrain parameter estimation. The 3D physical simulation results are presented in section 5. Finally, the last two sections conclude the publication.

II. TECHNICAL ASPECTS

A. The SOLERO Rover

The mechanical structure of SOLERO is similar to that of Shrimp, a rough terrain rover developed at EPFL in 1999 [7] [8]. This passive structure shows excellent climbing abilities without any specific active control. As shown in Fig. 1, SOLERO has one wheel mounted on a fork in the front, one wheel attached to the main body at the rear and two bogies on each side. The parallel architecture of the bogies and the spring suspended fork provide a high ground clearance

while keeping all six motorized wheels in ground-contact at any time. This ensures excellent climbing capabilities over obstacles two times higher than the wheel diameter and an excellent adaptation to all kinds of terrains.

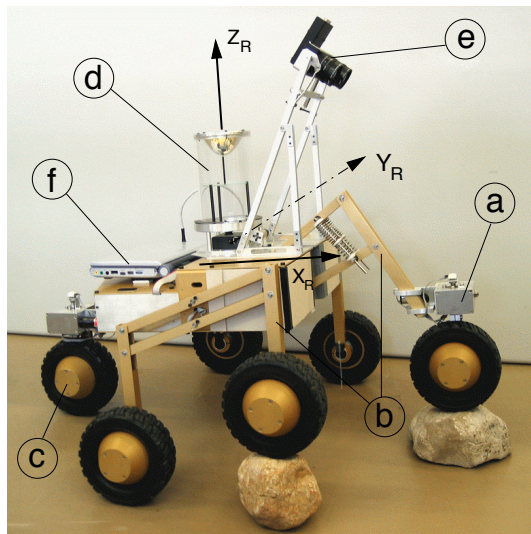


Fig. 1 Sensors, actuators and electronics of SOLERO. **a)** steering servo mechanism, the same is used for the rear wheel **b)** passively articulated bogie and spring suspended front fork (equipped with absolute angular sensors) **c)** 6 motorized wheels **d)** omnidirectional vision system **e)** stereo-vision module, orientable around the tilt axis **f)** laptop. Not visible on the picture: an additional PC104 computer, electronics and an IMU (Inertial Measurement Unit)

B. Tactile wheel

As discussed in the introduction, the estimation of wheel-ground contact angles is a critical issue. An alternative to indirect estimation methods is to directly measure the forces on the wheel periphery. This can be done using flexible wheels equipped with sensors measuring deflection. An example of such a device is depicted in Fig. 2 and more information can be found in [9].

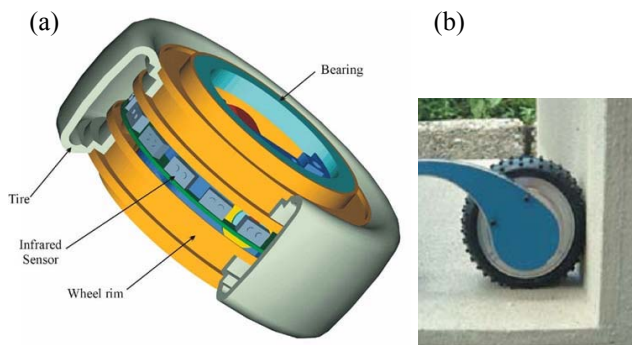


Fig. 2 The tactile wheel (developed at EPFL by Michel Lauria). **a)** sixteen infrared proximity sensors measure the tire compression all around the wheel. **b)** picture of the front wheel of the robot Octopus, equipped with tactile wheels [9].

The wheel-ground contact angle is computed with a weighted mean of the sensor signals. This way, a smooth transition is obtained when dealing with difficult terrain profile such as depicted in Fig. 2b.

III. QUASI-STATIC MODEL

The intent is to formulate a holistic model of a robot to optimize the control of the wheel motor torques in order to minimize wheel slip. Therefore it is helpful to review the governing equations on wheel slip.

The wheel is balanced if the friction force fulfills the following inequation (this case represents static friction)

$$F_f \leq \mu_0 \cdot N \quad (1)$$

where N is the normal force acting on the wheel and μ_0 the static friction coefficient. If the friction force can't balance the system, the wheel slips. In order to avoid slip, the friction force, which depends directly on the motor torque M , should satisfy the equation

$$T = \frac{M}{R} = F_f \leq \mu_0 \cdot N \quad (2)$$

where T is the traction force and R the wheel radius. Equation (2) suggests that there are two ways to reduce wheel slip. First, assume that μ_0 is known and set $T \leq \mu_0 \cdot N$. In fact, it is difficult to know μ_0 precisely because it depends on the kind of wheel-soil interaction. During exploration, the kind of soil interacting with the wheels isn't known making μ_0 impossible to pre-determine.

Another way to avoid wheel slip is to first assume that the wheel does not slip. It is then possible to calculate the forces T and N as a function of the torque and the result is optimized in order to minimize the ratio T/N . Accounting for the previous assumption

$$\frac{T}{N} = \frac{\mu_n \cdot N}{N} = \mu_n \quad (3)$$

μ_n is similar to a friction coefficient. By minimizing this ratio, then minimizing μ_n , we optimize our chances that this coefficient is smaller than the real friction coefficient μ_0 . If this is true, there is no slip. Therefore, it is possible to minimize the ratio T/N without knowing the real static friction coefficient. The second method is used here, because it is more robust. However, the determination of the parameters T and N require a model of the mobile robot.

A. Mobility analysis

The mobility of a rolling robot in straight motion should ideally be one, indicating that the robot can move in a constrained direction. Grubler's Mobility equation in three dimensions can be described as

$$MO = 6 \cdot n - 5 \cdot f_1 - 4 \cdot f_2 - 3 \cdot f_3 - 2 \cdot f_4 - f_5 \quad (4)$$

where n is the number of mechanical parts and f_j the number of joints of each type ($j = 1, \dots, 5$, for example f_1 : the number of pin joints, f_3 : the number of spherical joints). The mobility equation is a guideline for determining if a system is statically determinate. Many real systems contain redundancy in links and joints resulting in hyper statism. More sophisticated modelling methods are required to analyze the distribution of forces in a hyper static system. Another approach is to model selective joints with additional degrees of freedom. Intelligent selection of these joints can minimize

the error associated with a quasi-static solution. While the modeled kinematic chain is a simplification, it can be good enough to support motor control.

Example for the SOLERO

In a first step, we can consider the wheel-ground contacts as spherical joints and all the pin joints in the mechanism as 1DOF revolute. For the SOLERO, the calculation of the mobility using equation (4) is -20 rather than 1. The system is, therefore, significantly hyper static and requires a modified model for a possible quasi-static solution. Two significant modifications to joint degrees of freedom assist the model.

The first one involves the representation of the wheel-ground joint mobility. For a standard wheel without slip, the joint that represents the wheel-ground contact can be modelled as a spherical joint allowing three degrees of freedom (rotations about the three axes). Motor torque on the wheels will directly affect the forces in that contact plane. Lateral forces are not influenced by the motor torque. Therefore, the system was modelled with the lateral forces being carried by the wheel fixed to the body and the wheel on the front fork. The wheels on the bogies were modelled with no resistance in the lateral direction (4 degrees of freedom).

The second modification acts on the representation of the redundant kinematic chains. It is possible to model selected joints on redundant kinematic chains with greater degrees of freedom. This results in force being transmitted through direct flow patterns. Because the model is being used to optimize motor torques, inaccuracies in the internal linkage forces can have minimal effect.

Fig. 3 shows the resulting kinematic model of the SOLERO mobile robot. The numbers at the link connections indicate the degrees-of-freedom of that joint.

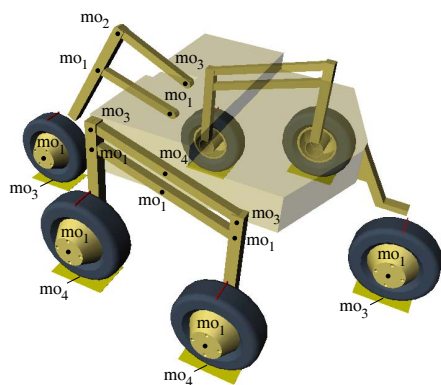


Fig. 3 Mobility of the joints

The final mobility can then be calculated using equation (4) to produce

$$MO = 6 \cdot 18 - 5 \cdot 14 - 4 \cdot 1 - 3 \cdot 7 - 2 \cdot 6 = 1 \quad (5)$$

B. Quasi-static model of a robotic system

For a 3D static model, 6 equations (3 torques and 3 forces) are applied to each body, containing ground reaction forces, gravity forces (weight) and external forces. Dynamic forces

are considered to be negligible because the speed is low (smaller than 0.1 m/s for exploration rovers [12]). Therefore, the model is referred to as quasi-static.

Model of the SOLERO

The SOLERO has 18 parts and is characterized by $6 \times 18 = 108$ independent equations describing the static equilibrium of each part and involving 14 external ground forces, 6 internal wheel torques and 93 internal forces and torques for a total of 113 unknowns. The weight of the fork and the bogies link has been neglected whereas the weight of the main body and the wheels is considered.

Of course, it is possible to reduce this set of independent equations because we have no interest in implicitly calculating the internal forces of the system. The variables of interest are the 3 ground contact forces on the front and the back wheel, the 2 ground contact forces on each wheel of the bogies and the 6 wheel torques. This makes 20 unknowns of interest and the system could be reduced to $20 - (113 - 108) = 15$ equations. This leads to the following matrix equation:

$$M_{15 \times 20} \cdot \vec{U}_{20 \times 1} = \vec{R}_{15 \times 1} \quad (6)$$

where M is the model matrix depending on the geometric parameters and the state of the robot, U a vector containing the unknown and R a constant vector.

IV. TORQUE OPTIMIZATION

The controllable inputs of the system are the six wheel torques. Since there are five more unknowns than equations it is possible to write an equation expressing the torques as linearly dependant. The 14 other equations define the external forces as a function of the torques.

The model of the SOLERO is indeterminate because there are less equations than variables and the set of solutions is of dimension five (number of wheels -1). The goal of the optimization is to minimize slip. This can be achieved by maximizing the traction forces, which is equivalent to minimizing the function $\max(T_i/N_i)$ for the wheels [2]. Because the system of equations is non-linear, a numerical method is implemented. The optimization uses a combination of different algorithms and is depicted in Fig. 4. Firstly the Equal Torques solution is checked versus the following constraints

- Motor saturation: the torques of the optimal solution must be smaller than the maximal available torque.
- Normal forces: the normal forces N_i must be greater than zero.

If this solution is valid, it is taken as the initial solution for the Fixed Point optimization (A). If it doesn't fulfill the constraints, a valid initial solution is computed using the Simplex Method (B). The optimal solution is then provided either by (A) or the Gradient optimization (C), which is a standard gradient descent optimization.

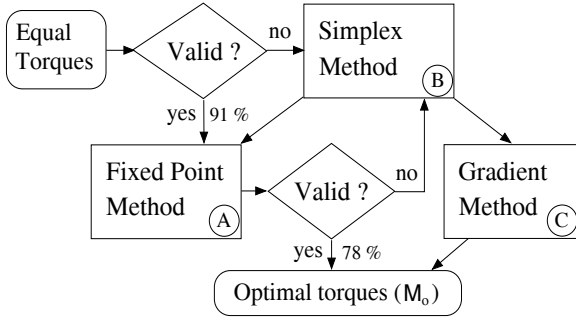


Fig. 4 Wheel torque optimization algorithm

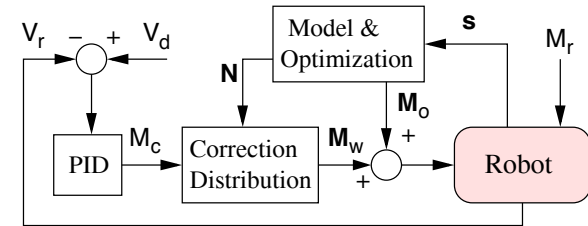
The optimization algorithm has been tested for around twenty thousand states generated automatically considering different angles for each input parameter. 80% of the computed torques correspond to a friction coefficient smaller than 0.6. This allows to limit the slip of the wheels in most of the situations.

The execution times for the algorithms A, B and C are 6 ms, 5 ms and 20 ms respectively (1.5 GHz processor). The worst case is about 31 ms. However, as it can be seen in Fig. 4, the majority of the states are handled in 6 ms.

V. ROVER MOTION

The optimal torques obtained in the previous section minimize the risk of slip in pseudo-static conditions. For constant speed motion, an additional motor torque must be added in order to compensate the rolling resistance torque.

In [3] the Hertz-Föppl rolling resistance model ([11]) has been incorporated into the global model to guarantee the rover motion. However, as discussed in the introduction, it is difficult to estimate the model parameters. Furthermore, a small estimation error would cause the rover to constantly accelerate or decelerate. For these reasons an additional control loop has been introduced in order to both ensure stability and estimate the rolling resistance. The global control system is depicted in Fig. 5.



V_d	desired rover velocity	M_o	vector of optimal torques
V_r	measured rover velocity	N	vector of normal forces
M_r	rolling resistance torque	s	rover state vector
M_c	correction torque	M_w	wheel corr. torques

Fig. 5 Global rover motion control. The vector s includes the wheel-ground contact angles, the internal links and the roll and pitch angles. V_r can be estimated by using onboard sensors fusion (see [10] for 3d motion estimation of SOLERO)

The kernel of the control loop is a PID controller. It provides the additional torque to apply to the wheels in order to reach the desired velocity V_d . M_c is actually an estimate of the global rolling resistance torque M_r , which is considered

as a perturbation by the PID controller. The rejection of the perturbation is guaranteed by the integral term of the PID. Because the rolling resistance is proportional to the normal force, the individual corrections for the wheels are distributed using

$$M_{w_i} = \frac{N_i}{N_m} \cdot M_c \quad i = 1, \dots, 6 \quad (7)$$

where N_i is the normal force on wheel i and N_m the average of all the normal forces.

The derivative term of the PID allows to account for non modeled dynamic effects and helps to stabilize the system. The parameters estimation for the controller is not critical because we are more interested in minimizing slip than in reaching the desired velocity in an optimal way. For locomotion in rough terrain, a residual error on the velocity can be accepted as long as slip is minimized. Furthermore, the system offers an intrinsic stability because the ratio between inertia and motor torques is large.

VI. EXPERIMENTAL RESULTS

A simulation phase has been initiated in order to test the algorithms and verify the theoretical concepts and assumptions. The simulation parameters have been set as close as possible to the real operation conditions. However, the intent is not to get exact outputs but to compare different control strategies and detect/solve potential implementation problems.

A. Simulation tools

Simulations have been performed with the Open Dynamics Engine (ODE). This engine is a platform independent and open source library for simulating rigid body dynamics in three dimensions. It has advanced joint types and integrated collision detection with friction. The source code being available it is possible to integrate more sophisticated simulation models such as rolling resistance, friction etc. In this application, a rolling resistance proportional to the normal force on the wheel has been implemented. This relation is generally assumed in the literature. Table 1 presents a non-exhaustive list of the real rover parameters, which have been used in the simulation. All the geometric dimensions have been extracted from the mechanical drawings.

TABLE 1 SOLERO MAIN CHARACTERISTICS

Rover's main body mass	7.4 kg
Wheel mass	0.7 kg
Steering mechanism mass	0.6 kg
Spring constant	357 N/m
Wheel diameter	0.15 m

The simulation tools allow to test and compare different traction control strategies. In these experiments, wheel slip has been taken as the main benchmark and the performance of torque and speed control have been compared. The imple-

mented speed controller is the one presented in [1].

The slip of wheel i at time step k can be computed with:

$$s_k^i = \Delta w_{(k-1,k)}^i - \Delta \theta_{(k-1,k)}^i \cdot R \quad (8)$$

where $\Delta w_{(k-1,k)}^i$ is the true wheel displacement, $\Delta \theta_{(k-1,k)}^i$ the angular change and R the wheel radius. The total slip of the rover occurring during an experiment is defined as

$$S = \sum_{i=1}^6 \sum_k s_k^i \quad (9)$$

The body collision algorithm of ODE provides n contact points around the wheel together with the normal forces. This data is similar to what can be measured with a tactile wheel (the tire compression is more or less proportional to the applied force). Moreover, the same method as presented in section II is used to compute the wheel-ground contact angles. In case the wheel does not touch the ground, the previously computed contact angle is taken.

B. Experiments

Two sets of experiments have been conducted. The first set comprises different terrain profiles in two dimensions (x and z) and the second, full 3D environments. In both cases, the nominal speed of the rover is 0.1 m/s and the friction coefficient has been set equal to 0.7.

Experiment set of type one

Terrain profiles similar to the one depicted in Fig. 6 have been generated and the simulation performed with both torque (our method) and speed control ([1]). Thanks to the terrain symmetry, the gravity center trajectories are the same whatever control type is used.

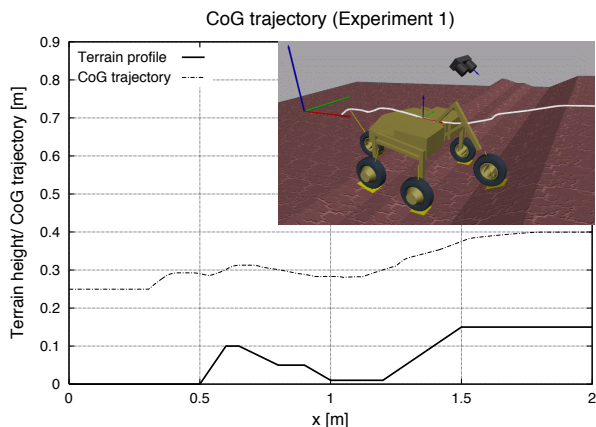


Fig. 6 Trajectory of the center of gravity for an experiment of type one. That kind of terrain is difficult for a wheeled rover because it includes many slope changes.

Fig. 7 depicts typical results that have been obtained on such terrains. For a specific wheel, slip can be locally higher with torque control than with speed control. However, the total slip remains always smaller with torque control for all the experiments. Another interesting result is that the difference between the two methods increases when the friction coefficient gets lower. In other words, the advantage of us-

ing torque control becomes more and more interesting as the soil gets more slippery.

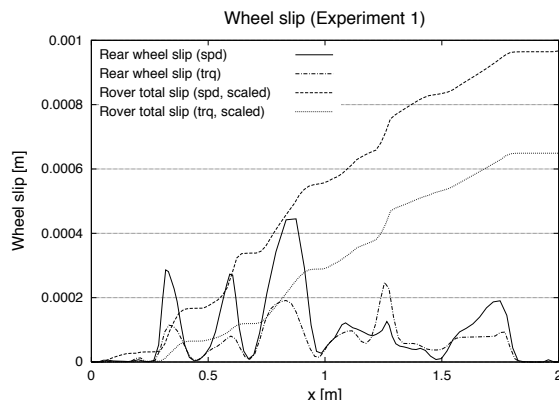


Fig. 7 Total slip and rear wheel slip for both speed (spd) and torque (trq) control. Total slip is scaled by a factor of 500. Locally wheel slip can be bigger with torque control but the total slip remains always smaller: 31% better than speed control.

Experiment set of type two

Here, full three dimensional terrains are used for the experiments. They have been generated randomly with step, sinus, circle and particle deposition functions. This time, because the terrains are not symmetric, the trajectory of the rover depends on the control strategy. Therefore it is difficult to compare performance between torque and speed control. However, we have considered an experiment as valid when the distance between the final positions of both trajectories is smaller than 0.1 m. This distance is small enough to allow performance comparison. For all the valid experiments, torque control showed better performance than speed control. In some cases the rover was even unable to climb some obstacles and to reach the final distance when driven with speed control. Otherwise, the simulations lead to the same conclusion as for the experiments of type one. Fig. 8 depicts one of the terrains used for the simulations and Fig. 9 the corresponding results.

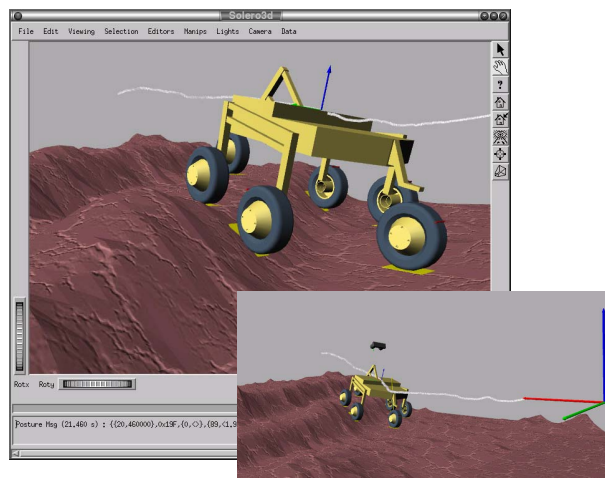


Fig. 8 Snapshots of an experiment of type two. The total travelled distance along x is 3.5m. That kind of terrain is challenging for a wheeled rover because there is much side slip when the rovers start climbing the slope.

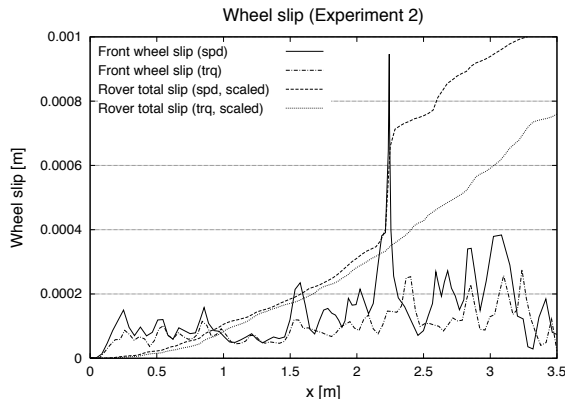


Fig. 9 Total slip and front wheel slip for an experiment of type two. The difference gets bigger as the rover deals with true rough terrain. Total slip is scaled by a factor of 800. At the end of the experiment, the torque controller performs 26% better than the speed controller.

VII. DISCUSSIONS

For these experiments, it is difficult to provide a quantitative result for describing the performance of one controller with respect to the other. Indeed, the performance depends on the terrain topology: for easy terrains the performance of both controllers is almost the same, whereas torque control performs better as the terrains become more challenging. However, very interesting behaviors of the torque controller have been systematically observed in all the experiments

- For each wheel, the slip signal is scaled down when using torque control. Such a behaviour can be observed in Fig. 7 and Fig. 9: the peaks are at the same places for both controllers but the amplitude is much smaller for the torque controller.
- The total slip of the rover is always smaller when using torque control.
- Strong assumptions have been used during the development of the torque controller i.e. no slip and the wheels touch the ground all the time. During the experiments both assumptions have been violated but the system was able to recover and keep its stability, even in difficult situations such as depicted in Fig. 8.

The simulations showed good results and promising perspectives. Furthermore, they allowed to detect potential problems and address implementation details. This is a step closer to the real application. Moreover, the design of a metallic tactile wheel is currently investigated. Such a design will not only provide precious information such as wheel-ground contact angles but also allow more drawbar-pull, especially in sandy terrains.

VIII. CONCLUSION

In this paper we have proposed a solution for wheel-ground contact angles measurement and a global control strategy minimizing wheel slip in rough terrain. The simulations show clearly the advantage of torque control versus speed control. Furthermore, the proposed method has the advantage to avoid relying on complex wheel-soil interac-

tion models, whose parameters are unknown in challenging terrains.

The algorithms can be run online and the first prototype of tactile wheel showed good results. Thus, the success probability during the implementation phase on a real rover can be considered as high.

ACKNOWLEDGEMENTS

The author would like to thank Rolf Jordi for his work on the simulator and Michel Lauria for the invention of the “universal wheel”. This project has been partly funded by the European Space Agency.

REFERENCES

- [1] Baumgartner E. T., Aghazarian H., Trebi-Ollennu A., Huntsberger T. L., Garrett M. S., "State Estimation and Vehicle Localization for the FIDO Rover", *Sensor Fusion and Decentralized Control in Autonomous Robotic Systems III*, SPIE Proc. Vol. 4196, Boston, USA, 2000.
- [2] Iagnemma K. and Dubowsky S., "Mobile Robot Rough-Terrain Control (RTC) for Planetary Exploration", *Proceedings of the 26th ASME Biennial Mechanisms and Robotics Conference, DETC 2000*, 2000.
- [3] Lamon P., Krebs A., Lauria M., Shooter S. and Siegwart R., "Wheel torque control for a rough terrain rover", *IEEE International Conference on Robotics and Automation*, New Orleans, USA, 2004.
- [4] Iagnemma K., Shibley H., Dubowsky S., "On-Line Terrain Parameter Estimation for Planetary Rovers", *IEEE International Conference on Robotics and Automation*, Washington D.C, USA, 2002.
- [5] Iagnemma K., Dubowsky S., "Vehicle Wheel-Ground Contact Angle Estimation: with application to Mobile Robot Traction Control", *7th International Symposium on Advances in Robot Kinematics, ARK '00*, 2000.
- [6] Lamon P., Siegwart R., "3D-Odometry for rough terrain - Towards real 3D navigation", *IEEE International Conference on Robotics and Automation*, Taipei, Taiwan, 2003.
- [7] Siegwart R., Estier T., Crausaz Y., Merminod B., Lauria M., Piguet R., "Innovative Concept for Wheeled Locomotion in Rough Terrain", *In Proceedings of the Sixth International Conference on Intelligent Autonomous Systems*, Venice, Italy, 2000.
- [8] Siegwart R., Lamon P., Estier T., Lauria M., Piguet R., "Innovative design for wheeled locomotion in rough terrain", *Journal of Robotics and Autonomous Systems*, Elsevier, vol 40/2-3 p151-162.
- [9] Lauria M., Piguet Y. and Siegwart R., "Octopus - An Autonomous Wheeled Climbing Robot", *In Proceedings of the Fifth International Conference on Climbing and Walking Robots*, Published by Professional Engineering Publishing Limited, Bury St Edmunds and London, UK, 2002.
- [10] Lamon P., Siegwart R., "Inertial and 3D-odometry fusion in rough terrain - Towards real 3D navigation", *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sendai, Japan, 2004.
- [11] Kalker J.J., "Three dimensional elastic bodies in rolling contact", *Kluwer Academic Publishers*, Dordrecht, 1990.
- [12] Weisbin, C., Rodriguez, G., Schenker, P., Das, H., Hayati, S., Baumgartner, E., Maimone, M., Nesnas, I., and Volpe, R., "Autonomous Rover Technology for Mars Sample Return", *1999 International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS '99)*, 1999.