Proceedings of IDETC'05
ASME International Design Engineering Technical Conferences and
Computers and Information in Engineering Conference
September 24-28, 2005, Long Beach, California, USA

DETC2005-85620

WHEEL-TERRAIN CONTACT MODELING IN THE ROAMS PLANETARY ROVER SIMULATION

Garett Sohl
Jet Propulsion Laboratory
California Institute of Technology
4800 Oak Grove Drive, M/S 198-235
Pasadena, CA 91109
Email: gsohl@jpl.nasa.gov

Abhinandan Jain
Jet Propulsion Laboratory
California Institute of Technology
4800 Oak Grove Drive, M/S 198-235
Pasadena, CA 91109
Email: jain@jpl.nasa.gov

ABSTRACT

This paper describes wheel-terrain contact modeling in the ROAMS physics-based simulator for planetary surface exploration rover vehicles. ROAMS models a wide range of rover systems and provides a user interface to a simulated rover. The wheel-terrain interaction is of primary interest during rover motion over rough terrain. The wheel-terrain contact model must provide physically realistic behavior without being too computationally intensive for real-time simulation. Physically realistic behavior may be defined using empirical data and ROAMS provides mechanisms for tuning the contact model parameters to match this data. In cases where empirical data is non-existent or there are large amounts of uncertainty, ROAMS can be used to extrapolate a range of behaviors based on physical parameters. The ROAMS simulator can be used in stand-alone mode, for closed-loop simulation with on-board software or for operator-in-the-loop simulations.

1 Introduction

There has been significant growth in the number of space exploration missions devoted to planetary surface operation using mobile rover vehicles. The current Mars Exploration Rover (MER) and upcoming Mars Science Laboratory (MSL) are prime examples of such missions. The twin MER rovers have each traversed over 3km since landing in early 2004, far exceeding the mission requirements. Some of those motions were conducted in regions, such as hillsides or crater walls, where the rover wheels experienced significant slipping on the terrain. A variety of empirical methods specific to the MER rover are used to compensate for slippage during these motions. The upcoming MSL mission includes significantly extended mission life and rover traverse distances in comparison to MER. A tool for examining slip behavior of a general class of rovers over more widely variable terrain is needed for upcoming missions.

The development and testing of on-board software for planetary rovers has traditionally been done using rover hardware platforms and testbeds. These hardware resources are expensive and typically over-subscribed. To alleviate this situation, validated modeling and simulation capabilities for surface rovers are being developed in **Rover Analysis, Modeling and Simulation (ROAMS)** [1] [2] [3] to support the mission in carrying out surface system trade studies, development of new rover technologies, closed-loop development and test of on-board flight software, and for use during mission operations. ROAMS includes models for various subsystems and components of the robotic vehicle and the operating environment. These include mechanical subsystem, sensors, on-board control software, as well as wheel-terrain interactions.

This paper describes the wheel-terrain contact model used in ROAMS. The physics-based, compliant contact model provides physically realistic behavior in a high-speed (real-time or faster than real-time) simulation environment. It allows for both wheel rolling and slipping behaviors as well as separation of the wheel

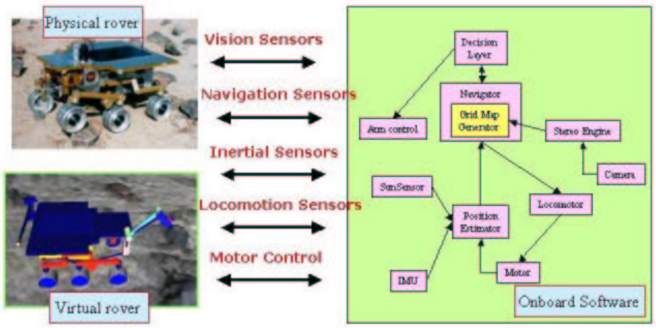

Figure 1. Typical closed-loop interfaces between on-board software and the physical/simulated rover

from the terrain. While transition between rolling and slipping is based on the available traction for each wheel, the model also allows for some statistical randomness in this transition. This randomness, when activated, allows us to smoothly transition between the rolling and slipping regimes. The contact model is highly configurable and its parameters can be tuned to match empirical data. We will describe preliminary validation testing of the contact model conduct in ROAMS.

2 ROAMS Design Goals

We describe first some of the key design goals that are driving the ROAMS development.

2.1 Validated Physics Based Models

A primary requirement on ROAMS is that it serve as a high-fidelity surrogate rover to support closed-loop testing beyond what is possible with just hardware rover testbeds. These high fidelity needs require ROAMS to implement (a) detailed physics based models of the rover mechanical platform including its kinematics and dynamics, (b) its suite of actuators and sensors such as wheel & steering motors and encoders, inertial measurement units (IMUs), sun sensors, cameras, and (c) models of the environment and the rover's interactions with the environment. Hand in hand with the model development process is an ongoing ROAMS simulator validation effort consisting of a series of experiments involving deterministic as well as statistical comparisons with physical rover data.

2.2 Model Configurability

Development of the rover flight system typically involves test platforms ranging from experimental technology development rovers all the way to flight breadboards and spares. The configuration of these platforms typically evolves over time with updates to the sensor/actuator suite, avionics and other hardware components. ROAMS is designed to provide models that shadow these multiple rover platform configurations at any given time and track their evolution over time. This required that ROAMS avoid monolithic, rover platform specific simulation implementations. Instead a conscious design strategy has been to allow users to configure ROAMS for different rover models easily at run-time via model data files. While allowing users to easily tailor simulations to the specific platforms, this configurability has been useful during the simulation validation effort to match ROAMS to rover model configurations used in the experiments.

2.3 Closed-Loop Simulations

As a test platform, ROAMS is meant to be used in a closed-loop fashion with the on-board rover software and hardware. This requires ROAMS to be embeddable within closed-loop testbed environments containing a mix of on-board software, real hardware and simulated hardware. ROAMS provides hardware-like command and sensing interfaces similar to actual hardware to allow such loop closure. Particular attention has been paid to simulation algorithm performance in order to meet the closed-loop timing requirements. Also, ROAMS is portable across Unix and real-time VxWorks platforms. The Dmex tool [2] provides auto-generated interfaces for embedding ROAMS within a Matlab/Simulink environment for control algorithm development and testing.

2.4 Layered Toolkit Approach

While simulations are expected to do the "right" thing, i.e. provide good fidelity, they also need to provide a significant level of instrumentation and other features for them to be usable. Since the inclusion of these features adds to code size and the number of external dependencies, ROAMS has adopted a layered design, where many of the features are implemented as optional plug-in extensions so they can be included as needed at run-time. This approach has also helped increase the amount of reusable modules within ROAMS.

2.5 Spacecraft Simulation Framework

To accelerate the development of ROAMS, ROAMS is built upon the existing DARTS & Dshell simulation framework [4] developed for spacecraft simulations. This strategy has allowed the ROAMS development effort to focus on the extensions needed for the surface rover domain. Likewise this has had the effect of making available these extensions to other simulators sharing the same simulation infrastructure. A case in point here is the DSENDS entry, descent and landing simulation tool [5] that uses the same DARTS & Dshell simulation framework and shares several modules with ROAMS including those for dynamics simulation and terrain environment modeling.

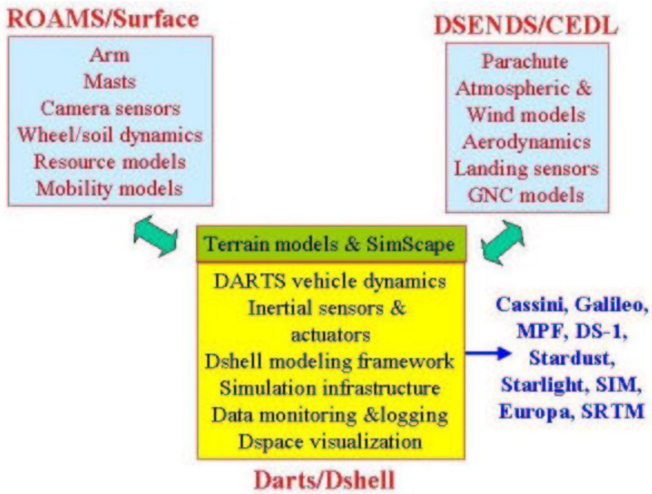

Figure 2. Common Dshell simulation infrastructure for ROAMS and DSENDS

2.6 Open source tools

Complementing our goal of using established spacecraft simulation capabilities, we have placed emphasis on using and adapting open source software wherever possible. This has led to the use of computational libraries such as SWIFT++ and ANN, visualization layers such as OpenInventor, POVRAY, graphical user interface tools such as Tk, Tix, Gtk, Gnocl, TCL & SWIG scripting interfaces, and documentation generation tools such as Doxygen within ROAMS.

2.7 Usability

With the increase in detail and functionality of ROAMS, we recognize the need to provide user interfaces to facilitate the use of ROAMS and reduce the learning curve. While the ROAMS core is implemented in C/C++, It includes a TCL scripting interface (auto-generated by the SWIG wrapper generation tool) to the core C/C++ classes to facilitate simulation configuration and regression testing. This scripting capability is also used to develop graphical user interfaces for users to change simulation modes, set rover goals, change simulation speed, take time steps, exercise rover degrees of freedom, select terrain models etc. The Dspace 3D visualization tool [2] provides run-time visualization of the rover simulation state.

3 ROAMS Wheel-terrain Contact Model

ROAMS provides a high-fidelity virtual rover as described in [3]. In addition to vehicle modeling, ROAMS must also model the rover environment. As a surface vehicle, the rover interacts with the environment primarily through the terrain. ROAMS uses a digital elevation map (DEM) to model the terrain geometry. The goal of the wheel-terrain contact model is the determination of the contact forces and torques exerted by the terrain on the rover wheels. These contact forces, along with wheel motor torques, provide the motive force for the rover and allow it to traverse over the terrain. The forces and torques must also excite on-board sensor models, such as gyros and accelerometers in a realistic fashion.

In a general formulation, there are three unknown force and three unknown torque components that define the net effect of each contact on the rover wheel. For a six wheeled rover, there are a total of 36 unknowns. However, a static force analysis of the rover provides only 6 equations (three linear and three angular) in these 36 unknowns. For the specific case of a rocker-bogey mechanism, three additional constraint equations can be generated (one rocker differential and two bogey constraints). However, even with those added moment constraints, the problem is still statically indeterminate: 9 equations in 36 unknowns.

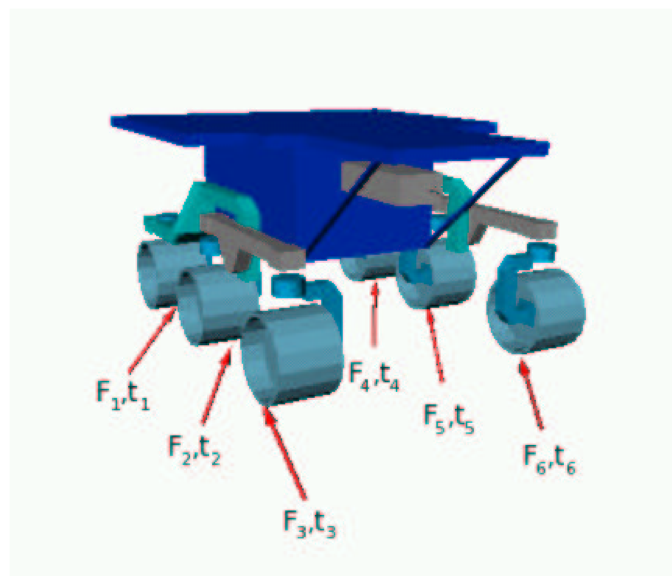

Figure 3. Rover contact forces and torques

3.1 Statically Indeterminate Techniques

The contact forces and torques on the six wheels of a rover in contact with the terrain are statically indeterminate. Sometimes it is possible to simplify the problem by eliminating unknowns until the problem can be solved. Many contact model formulations assume that the soil can only exert moments on the wheel about the terrain normal direction. This eliminates two unknowns for each wheel (reducing the total number of unknowns to 24). It is also possible to assume an effective point-plane contact between a point on the wheel and the "plane" of the soil. In this case, all the moments are zero (reducing the total number of unknowns to 18) [6]. ROAMS uses the point-plane contact assumption in determining the contact forces. ROAMS also estimates a surface contact area on each wheel based on wheel sinkage. This contact area is used in determining maximum allowable traction 2.

Another common assumption used to reduce the complexity of the problem is to assume that the vehicle roll angle is small [7]. Under this assumption, the transverse force component is zero for each wheel contact. This assumption is often used to decompose the six wheel rover into two planar problems [8]. Each planar problem contains three wheels on the left or right side of the rover. The planar problem is then analyzed in detail. This assumption, however, is not suitable to a general dynamics simulation since transverse forces can not be assumed to be zero for motions over undulating terrain.

Some approaches attempt to compute contact forces satisfying a set of validity constraints [9]. For example, contact forces that fall within the friction cones at each contact point and satisfy the static force/moment equations can be found. While these techniques can determine if a suitable set of contact forces exist, they do not guarantee uniqueness or continuity of the solution. In addition, they may fail to find solutions in cases that are outside the assumed solution space. This makes them very unsuitable for a general dynamics simulation where high, or even total, slippage conditions will be encountered.

Another approach is to solve for contact forces that optimize a given criteria (instead of merely satisfying a given criteria) [10]. Assuming a well posed problem and good optimization routines, a unique solution can be found. Continuity of solution as the rover moves over the terrain is not guaranteed, but can reasonably be expected in many cases. Criteria such as minimum energy are often used. This procedure is generally very time consuming and while the minimization of a single criteria produces a unique answer, there is no guarantee that answer is a correct solution for the contact forces. The time consuming nature of optimization techniques makes this method unsuitable for a real-time dynamics simulation where rapidly changing contact forces for each wheel must be computed hundreds of times per second.

3.2 Compliance Based Techniques

Adding compliance to a statically indeterminate system can allow a solution to be found [11] at the cost of adding system states and increasing numerical stiffness of the problem. The forces on an object as simple as a four legged table on a flat surface are statically indeterminate. However, the introduction of compliance in the four legs allows the contact forces to be directly computed. Adding compliance provides a unique solution that is generally correct under the limitations of the compliance model. However, allowable stiffness is often restricted by the choice of numerical integration algorithm [12]. If the actual stiffness can not be used (due to numerical stability problems) compliance techniques may only approximate the correct answer. However, it is observed that compliance techniques will degrade gracefully and still provide good approximations with less than ideal stiffness parameters.

Compliance techniques provide a solution for contact forces based on the deflection of a spring-damper system. This deflection is directly related to the state of the system. Since the system state is always available to a dynamics simulation, these computations can be done very quickly. This is highly desirable in a real-time simulation environment, where optimization or other search techniques are too slow.

The contact model in ROAMS assumes point-plane contact. The terrain under the wheel is assumed to be locally planar and the contact forces are applied to a single point on the wheel. Under this assumption, there are three unknown force components for each wheel contact. Two separate and independent compliance systems are used to compute the three force components. One compliance system is used to compute the force component in the normal direction. The normal direction is defined as perpendicular to the “plane” of the terrain. Once the normal force is computed, it can be used to estimate wheel sinkage [13] and resulting contact area. Even though the force is applied at a single point on the wheel, the sinkage and resulting contact area is important in estimating maximum traction. The second compliance system is a two degree of freedom system. It is used to compute the two components of the force in the plane of the terrain. The next two sections will describe how the normal and tangent plane systems are implemented in ROAMS.

3.3 Normal Force

The magnitude of the normal force is the foundation of almost every contact mechanics formulation. From a simple Coulomb friction model to a complex terra-mechanics model [13], they all use the magnitude of the normal force to determine the available traction force in tangent directions. However, these formulations do not describe *how* to compute the normal force. It is assumed to be a given for the problem. In the case of a six wheeled rover in contact with undulating terrain (where the terrain normal at each contact point is different), these forces are statically indeterminate.

In order to compute the force in the normal direction, ROAMS uses a single degree of freedom, Hunt-Crossley [14] compliance system at each wheel:

$$F_N = k_N \delta_N^n + \frac{3}{2}\alpha_N \dot{\delta}_N \delta_N^n \tag{1}$$

where F_N is the force in the normal direction, k_N is a spring constant, α_N is a damping constant, n is the non-linear deflection exponent, and δ_N is the deflection. Figure 4 shows a side view of the normal direction compliance system. The normal direction

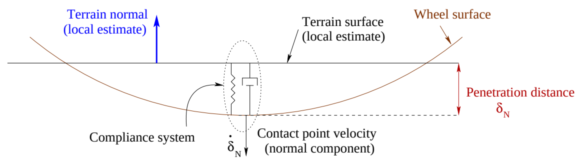

Figure 4. Normal direction compliance system

is determined by an examination of the terrain DEM for a small area under the wheel. There is an assumption that the terrain normal is relatively smooth and that the sinkage of the wheel into the terrain does not significantly effect the terrain normal at the contact point. The deflection of the compliance system is based on the penetration of the wheel into the terrain in the terrain normal direction. The location of the contact point on the wheel is defined as the point on the wheel that penetrates the farthest into the terrain. As the wheel penetrates the terrain, the compliance system applies an opposing force at the contact point. The opposing force will rapidly reach equilibrium with the weight of the rover supported by that wheel.

Equation 1 requires an estimate of the penetration of the wheel into the terrain. To simplify this computation, wheels are assumed to be cylindrical and the terrain under each wheel is assumed to be locally planar. These assumptions allow rapid computation of the penetration distance and contact point on the wheel based on the local terrain height field, wheel position and wheel attitude. Since the penetration of the wheel into the terrain is generally small and the weight supported by a given wheel may be substantial, ROAMS uses a very stiff spring in the normal direction (k_N is large) to prevent excessive penetration. This can lead to stability problems during numerical integration and often requires tuning the normal direction spring and damping constants based on the total rover mass. It is also important to note that the Hunt-Crossley compliance model in Equation 1 does not attempt to model the sinkage of a wheel into soil, but is rather an algorithmic convenience used to compute the statically indeterminate normal force. After a normal force has been found, it can be used to estimate the actual sinkage of a wheel into the soil based on soil properties such as density and cohesion. This results in two separate concepts of wheel-soil penetration: one from the compliance model and one from terra-mechanics.

3.4 Tangent Plane Forces

While the force in the normal direction provides “support” for the vehicle, it is the tangent plane forces that allow a wheeled vehicle to move. The tangent plane is defined as the plane perpendicular to the normal vector direction. Most contact models follow the basic premise that there is a limit to the magnitude of the tangent forces. Once that limit is reached, the tangent force can no longer prevent relative motion between the contact point and the terrain and the wheel starts to slip. The transition between rolling contact (where the contact point has zero velocity relative to the terrain) and slipping is a crucial concept. Allowable traction force for soil is given as [15]:

$$||F_{Tmax}|| = cA + ||F_N|| \tan\phi \tag{2}$$

where F_{Tmax} is the maximum force in the tangent direction, c is the soil cohesion, A is the contact area, F_N is the normal force and ϕ is the soil friction angle. The contact area A is estimated

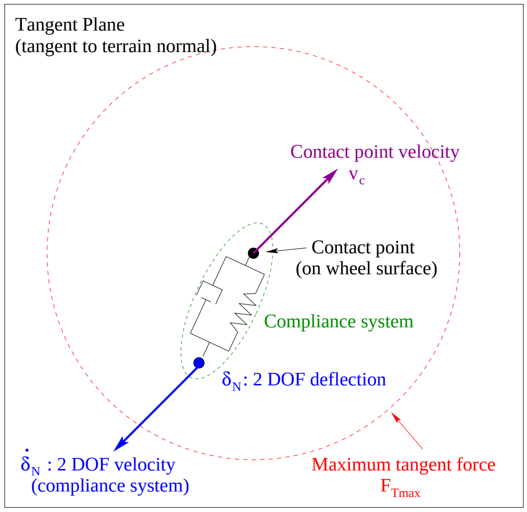

Figure 5. Tangent plane compliance system

based on the sinkage due to normal force and the geometry of the wheel. Equation 2 shows that the normal force should be computed first since its magnitude limits the maximum traction at the contact. Figure 5 shows an overhead view of the tangent plane compliance system.

In the **slipping regime**, the tangent plane force has reached its maximum allowable value (corresponding to the red, F_{Tmax} circle in Figure 5). The tangent plane force opposes the relative motion of the contact point. The tangent force is pointed in a direction opposite the relative velocity vector of the contact point. If the velocity of the contact point is not in the tangent plane, it will be projected into the tangent plane to determine the direction of the tangent force vector. The magnitude of the tangent force in the slipping regime is equal to the maximum allowable force:

$$\vec{F}_T = ||F_{Tmax}|| \frac{-\vec{v}_c}{||\vec{v}_c||} \tag{3}$$

where $\vec{F}_T$ is the 2D tangent force and $\vec{v}_c$ is the velocity of the contact point relative to the terrain projected onto the 2D tangent plane.

Computing the tangent plane force in the **rolling regime** is more problematic. In this regime, the velocity of the contact point with respect to the terrain is ideally zero. However, this does not mean that the tangent plane forces are zero. The only constraint is that the tangent plane force magnitude remain less that the maximum:

$$||\vec{F}_T|| \leq ||F_{Tmax}|| \tag{4}$$

To solve the problem of computing tangent forces for the rolling regime, ROAMS uses a two degree of freedom compliance system as outlined in [11]. A linear spring-damper model is used:

$$\vec{F}_T = k_T \vec{\delta}_T + d_T \dot{\vec{\delta}}_T \tag{5}$$

where k_T is the spring coefficient, d_T is the damping coefficients and $\vec{\delta}_T$ is the 2D deflection in the tangent plane. The two degrees of freedom allow the compliance system to move in any tangent plane direction. Unlike the normal force compliant system, whose deflection is based on the position of the wheel and the local terrain height, the 2D deflection of the tangent plane system ($\vec{\delta}_T$) is not based on the rover state. Instead, the *derivative* of the deflection is based on the system state. Specifically, the derivative of the deflection is defined as the opposite of the velocity of the contact point relative to the surface:

$$\dot{\vec{\delta}}_T = -\vec{v}_c \tag{6}$$

Additional system states are added and used to track the actual deflection of the tangent plane compliance systems.

It is important to understand why the *derivative* of the deflection is defined in this manner. First, consider what it means for the velocity of the contact point relative to the surface to be zero. This velocity is zero when the wheel is in the rolling regime. If a torque is applied to a wheel resting on the surface, the contact point will begin to move relative to the surface in the absence of tangent plane forces. To oppose this motion, the tangent plane compliance system will deflect in the opposite direction - opposing the relative motion of the contact point. The resulting tangent plane forces propel the vehicle forward. The tangent plane compliance system acts as a controller that drives the tangent force to the correct rolling force. A small amount of slipping is accepted as this controller converges to the correct tangent force. The spring-damper coefficients used for the tangent plane compliance system serve as gains for this control system. Any torque applied to the wheel will change the equilibrium point for the control system and the tangent plane compliance system will attempt to deflect to a new point. The maximum traction force defines a limit on the deflection of the tangent plane compliance system. Once this deflection is reached, the wheel enters the slipping regime and no further deflection of the tangent spring is allowed.

Since the type of contact (slipping or rolling) is not known a-priori, ROAMS uses the following procedure to compute the tangent plane forces:

1. Assume contact is rolling and compute $\vec{F}_T$ as

$$\vec{F}_T = k_T \vec{\delta}_T + d_T \dot{\vec{\delta}}_T \tag{7}$$

2. If $||\vec{F}_T|| > ||F_{Tmax}||$, contact is sliding:

$$\vec{F}_T = ||F_{Tmax}|| \frac{-\vec{v}_c}{||\vec{v}_c||} \tag{8}$$

The derivative of the tangent deflection, $\dot{\delta}_T$ is computed by inverting Equation 7:

$$\dot{\vec{\delta}}_T = F_T^{-1}(\vec{\delta}_T, k_T, d_T) \tag{9}$$

3. If $||\vec{F}_T|| \leq ||F_{Tmax}||$, contact is rolling:

$$\dot{\vec{\delta}}_T = -\vec{v}_c \tag{10}$$

3.5 Soil Randomness

The contact model described in 3.3 and 3.4 was originally formulated for rigid body contact. For two rigid bodies in contact, the transition between rolling and slipping regimes is very abrupt. Wheel-terrain contacts usually transition more gradually between regimes and can experience rapid fluctuations between rolling and slipping. Some slippage can occur even on perfectly flat terrain. These fluctuations are due to a wide range of effects that are not part of a simple rigid-body contact model, including variations in local soil parameters, soil deformation, wheel surface irregularities, vibratory effects, etc. While it is tempting to include more and more physical properties in the contact model, such modeling efforts are unrealistic for use in a general, real-time simulation tool as opposed to a detailed terra-mechanics analysis tool.

Equation 2 defines the tangent force where the contact transitions between rolling and slipping behavior. The contact model in ROAMS adds a Gaussian scaling factor to this equation:

$$||F_{Tmax}|| = s(cA + ||F_N|| \tan\phi) \tag{11}$$

where s is a Gaussian curve centered about a value of 1. This scale factor serves to "soften" the abrupt transition between rolling and sliding and more closely mimic the behavior of wheel-terrain contact. Some slipping may occur even on flat terrain (s is very small) and some rolling may occur on steep terrain (s is very large). The Gaussian scale factor s is defined by two parameters in ROAMS. The first parameter defines the maximum standard deviation of the Gaussian function and the second parameter defines how the standard deviation varies with local terrain angle. The standard deviation (s_{std}) for the scale factor s is:

$$s_{std} = p_1 - p_2 R \tag{12}$$

$$R = \begin{cases} \frac{\theta}{\phi}, & \theta \leq \phi \\ \frac{\phi}{\theta}, & \theta > \phi \end{cases} \tag{13}$$

where p_1 and p_2 are the two soil randomness parameters, ϕ is the soil friction angle and θ is the angle between the terrain normal and the vertical. The first parameter (p_1) defines the maximum standard deviation for the random Gaussian function. The second parameter (p_2) defines how the standard deviation varies with soil angle. The standard deviation is minimum when the terrain normal under a given wheel is equal to the soil friction angle ($R \rightarrow 1$). The standard deviation increases as the terrain normal becomes either much smaller or much larger than the friction angle ($R \rightarrow 0$). This increased standard deviation gives a chance of slipping for situations where rolling is the nominal behavior and vice versa. The choice of parameters p_1 and p_2 allows us to tune the behavior of the simulation and create a gradual transition between the rolling and slipping regimes.

4 Contact Model Validation

In order for a rover simulation to be useful in developing rover navigation and control software, its behavior must correspond well with the operation of the actual rover in a real environment. Hence, in parallel with the ongoing development of ROAMS, we have been undertaking a validation effort for ROAMS using experimental data from rover mobility runs. Rover motion is a product of many different components and levels of the system. At the lowest level, there are rover suspension components (rockers and bogeys), wheels and motors. While these low level rover components are usually well defined, the terrain used for experimental data collection is not. The terrain shape (height field) and physical properties (density, cohesion, friction angle) are important simulation parameters that are not typically well known for natural terrain used during rover mobility tests. The experimental mobility tests can also show wide variations between runs for the same rover over the same nominal terrain. The non-repeatable nature of these tests requires that the simulation use statistical techniques for comparison with empirical test data.

4.1 Parametric Simulation

The wheel-soil contact model in ROAMS has a variety of parameters. While some of these parameters are physical (soil density, cohesion, friction angle), others are heuristic (soil randomness parameters p_1 and p_2). We use Monte Carlo testing to determine a set of model parameters that most closely matches experimental data from rover mobility testing. ROAMS has a framework to automate the collection of mobility data for a range of independent parameter sets. This data collection can be run in parallel on several computers to speed up the simulated data collection. A variety of motion primitives (straight, arc-turn, turn-in-place, etc.) on a variety of terrain types (flat, fixed slope, etc.) can be tested. The ROAMS mobility data is then compared to empirical data and the set of parameters that best matches the experimental data can be determined.

One example of tuning ROAMS parameters was the comparison of ROAMS against mobility testing on the Dynamic Test Model (DTM) of the MER rover. The DTM was designed to mimic the loads experienced by the MER rovers on the surface of Mars. The DTM rover is an earth-based testbed configured to have the same center of mass as the MER rover and approximately 120% of the MER rover's Mars weight. Several mobility tests were conducted on Earth using the DTM rover to determine slippage when driving on slopes of up to 20 degrees. Figures 6, 7 and 8 show slip data collected during uphill, downhill and cross-hill mobility testing of the DTM rover on flat terrain at slope angles of 0, 2.5, 5, 10, 15 and 20 degrees [16].

Figures 9, 10 and 11 show simulated results for straight uphill, straight downhill and straight cross-hill testing in ROAMS at 0, 5, 10, 15 and 20 degree slope angles. The ROAMS parameters were determined using Monte Carlo testing to find a parameter set that best matched the empirical data in Figures 6, 7 and 8. The data used for comparison are shown as red circles in Figures 9, 10 and 11. Since the empirical data does not have any statistical variance information, each data point was given equal weighting in determining the best set of ROAMS parameters. The resulting

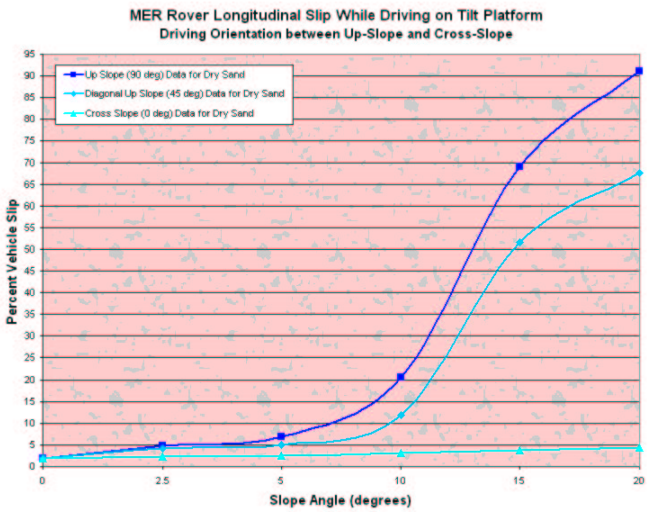

Figure 6. DTM uphill mobility test data

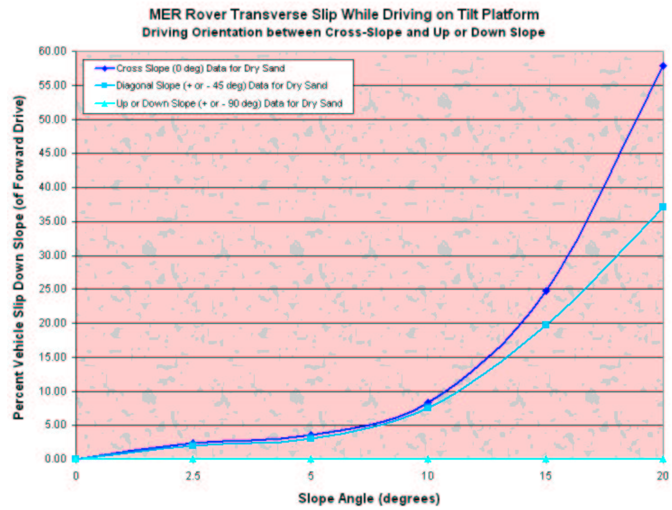

Figure 8. DTM cross-hill mobility test data

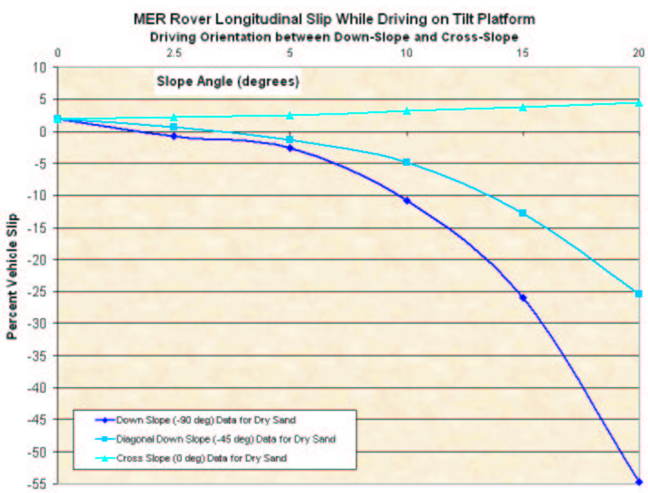

Figure 7. DTM downhill mobility test data

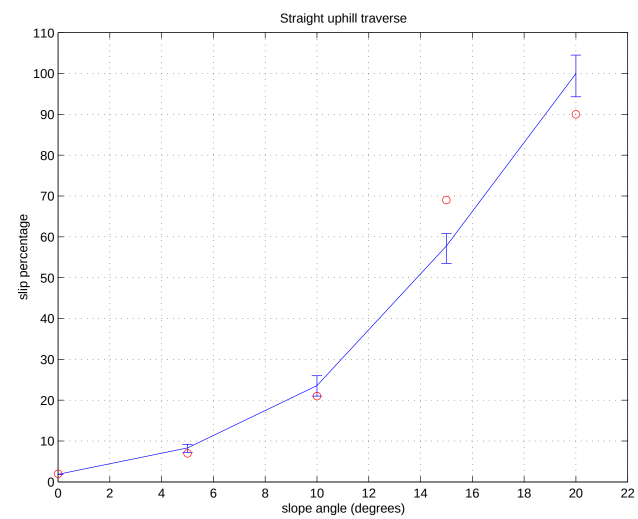

Figure 9. ROAMS straight uphill (with variance). Red circles are empirical data

simulation shows a good match for cross-hill traversals in Figure 11, but the same parameter set didn't provide as good a match for downhill motion (Figure 10). A technique for switching between parameters that best match empirical data in a given regime is being examined for use in ROAMS. This technique would allow ROAMS to modify parameters on-the-fly as the rover encounters different driving conditions.

5 Conclusions

This paper describes the wheel-terrain contact model used in ROAMS. The contact model handles both rolling and slipping contact as well as loss of contact between the wheel and the terrain. The model provides computational efficiency suitable for real-time simulation. Although based on models of rigid-body contact (where transition between rolling and slipping is abrupt), ROAMS uses Gaussian randomness to more closely mimic the behavior of wheel-terrain interactions. ROAMS also provides an automated framework for tuning model parameters to match experimental data. The validation of the wheel-terrain contact model is an ongoing process as ROAMS continues to develop for eventual use by missions such as the Mars Science Laboratory.

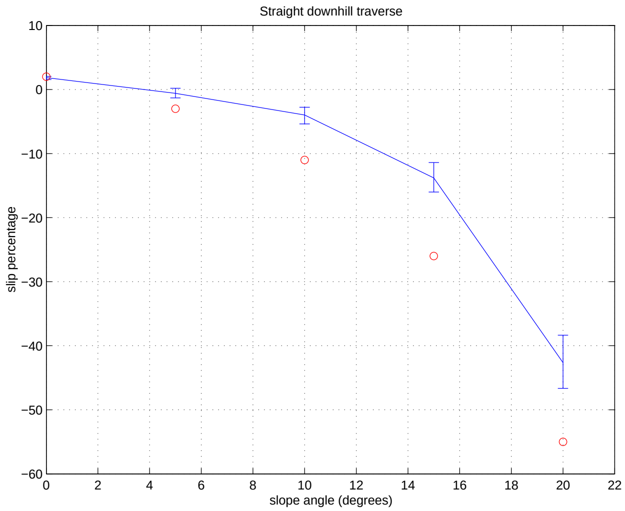

Figure 10. ROAMS straight downhill (with variance). Red circles are empirical data

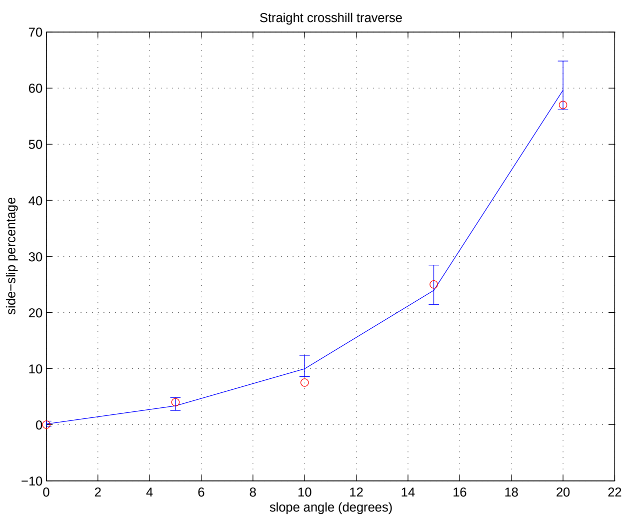

Figure 11. ROAMS straight cross-hill. Red circles are empirical data

Acknowledgments

The research described in this paper was performed at the Jet Propulsion Laboratory (JPL), California Institute of Technology, under contract with the National Aeronautics and Space Administration. We would also like to acknowledge the NASA's Mars Technology Program's support of the ROAMS development.

REFERENCES

[1] Jain, A., and Yen, J., 1999. "Roams: Rover analysis modeling and simulation software". In i-SAIRAS 1999.

[2] Jain, A., Guineau, J., Lim, C., Lincoln, W., Pomerantz, M., and Sohl, G., 2003. "Roams: Planetary surface rover simulation environment". In i-SAIRAS 2003.

[3] Jain, A., Balaram, J., Cameron, J., Guineau, J., Lim, C., Pomerantz, M., and Sohl, G., 2004. "Recent developments in the roams planetary rover simulation environment". In IEEE Aerospace Conference.

[4] Biesiadecki, J., Henriquez, D., and Jain, A., 1999. "A reusable, real-time spacecraft dynamics simulator". In 6th Digital Avionics System Conference.

[5] Balaram, J., Austin, R., Banerjee, P., Bentley, T., Henriquez, D., Martin, B., McMahon, E., and Sohl, G., 2002. "Dsends - a high-fidelity dynamics and spacecraft simulator for entry, descent and surface landing". In IEEE 2002 Aerospace Conference.

[6] Hacot, H., Dubowsky, S., and Bidaud, P., 1998. Analysis and simulation of a rocker-bogie exploration rover.

[7] Iagnemma, K., Genot, F., and Dubowsky, S., 1999. Rapid physics-based rough-terrain rover planning with sensor and control uncertainty.

[8] Farritor, S., Hacot, H., and Dubowsky, S. "Physics-based planning for planetary exploration". pp. 278–283.

[9] Hait, A., and Sim'eon, T., 1996. Motion planning on rough terrain for an articulated vehicle in presence of uncertainties.

[10] Hacot, H., 1998. Analysis and traction control of a rocker-bogie planetary rover.

[11] Kraus, P., Fredriksson, A., and Kumar, V., 1997. Modeling of frictional contacts for dynamic simulation.

[12] Alexander, R. K., 1994. "Stability of Runge-Kutta methods for stiff ordinary differential equations". *SIAM Journal on Numerical Analysis*, **31**(4), pp. 1147–1168.

[13] Bekker, M., 1969. *Introduction to Terrain-Vehicle Systems*. University of Michigan Press.

[14] Hunt, K., and Crossley, F., 1975. "Coefficient of resti- tution interpreted as damping in vibroimpact". *Journal of Applied Mechanics*, June, pp. 440–445.

[15] Terzaghi, K., 1943. *Theoretical Soil Mechanics*. John Wiley.

[16] Lindemann, R., 2005. "Dynamic testing and simulation of the mars exploration rover". In ASME International Design Engineering Technical Conference.