Physical-Based Simulation for Mars Exploration Rover Tactical Sequencing

Jeng Yen, Brian Cooper, Frank Hartman, Scott Maxwell, John Wright, Chris Leger

Jet Propulsion Laboratory (JPL)
California Institute of Technology
M/S 198-235   4800 Oak Grove Drive
Pasadena, California, USA 91109-8099
e-mail: Jeng.Yen@jpl.nasa.gov

Abstract – The Mars Exploration Rover (MER) mission has returned tremendous scientific information on a daily basis, owing to the efficient sequencing capability of the ground system tools. For planning the mobility and Instrument Deployment Device (IDD) sequences, physical-based simulation is applied to achieve fast and effective sequencing of complex rover and IDD maneuvers. The sequence rehearsal tool of the Rover Sequencing and Visualization Program (RSVP) is based on modeling and simulation of the multi-body mechanical systems. Using Configuration Kinematics (CK) and 3D terrain models, a methodology was developed to support a real-time, interactive graphics mode for the visualization tool. The sequence simulation is carried out using the on-board flight software modules for realistic rover behavior. This enables the scientists and rover planners to effectively develop the command sequences to maximize the science return of the MER mission while maintaining rover safety. This paper describes the innovative numerical algorithms and the command sequence simulation used by the MER mission for planning surface operations.

Keywords: simulation, configuration kinematics, multi-body system, interactive visualization.

1 Introduction

Simulation of space-borne systems has been well-developed and successfully applied to many past and current NASA missions. The modeling and simulation of spacecraft has been carried out using multi-body system dynamics [2-4]. The design and operation of the spacecraft are based on behavior predicted by high-fidelity simulation tools [5]. The modeling of robotic vehicles for surface operations is, however, very different from those of traditional space-borne systems. The most important aspect of rover modeling is the need to interact with the surrounding terrain [5]. Using the knowledge of the terrain to predict the state of the rover requires effective modeling of the rover- terrain interaction. In addition, the multi-body rover model should include all the motorized mechanisms that are commandable for a comprehensive sequence simulation.

Recently, the Rover Analysis Modeling Simulation algorithm [1], which computes the configuration of the robotic vehicles on rough terrain, has been developed at JPL for planetary surface exploration missions. Based on these and related methods, the RSVP simulation tool was developed to achieve the required accuracy and support the real-time visualization application. One crucial design decision of the command sequence simulation tool is to represent the multi-body model as a collection of subsystems. Not only is the rover model partitioned by an object-oriented hierarchy of multi-body subsystems, but the numerical algorithms that provide the solutions of configuration kinematics and inverse kinematics are also implemented via object-oriented methodology. The structure-preserving simulators inherit the subsystem model and the numerical methods, which directly map the numerical solutions to the corresponding physical structures indicated by the red blocks in Fig. 1. These subsystem simulators constitute the sequence rehearsal engine that has advantages over traditional multi-body simulations. The benefits of the structure preserving design for command sequence simulation will be detailed in this paper.

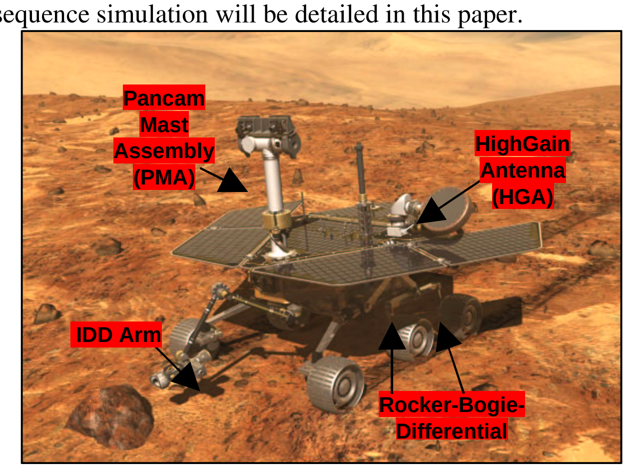

Figure 1 Mars Exploration Rover

In RSVP simulation, the rover model is represented by a set of hierarchical subgraphs of the mechanism models for the primary motion systems. These subgraphs are the foundation block for (1) receiving sensed data, (2) interpreting commands and (3) predicting the physical states of the corresponding mechanical systems.

1.1 Rover Mobility and Navigation

The mobility mechanism consists of the Rocker-Bogie-Differential (RBD) suspension and the four wheel-steering systems. As shown in Fig. 1, the suspension and the rover's kinematics, e. g., position and attitude, comprise a multi-body system of 10 states. Six wheel-driving with four wheel-steering motors comprise an additional 10 degrees of freedom. These are the basic states of the rover's mobility model.

1.2 Instrument Deployment Device (IDD)

The IDD is a five-joint robotic arm that is used to position instruments on or near science targets. The modeling of the IDD predicts the manipulation and placement of the science instruments. Each joint is limited to a predetermined range of motion.

1.3 Other Motorized Instrument Units

The panorama camera and the navigation camera are mounted on the Pancam Mast Assembly (PMA) with two motors to control their pointing direction. The high gain antenna consists of motors to control its azimuth and elevation. These motorized subsystems are prescribed by the motor controller.

2 Mars Exploration Rover Simulation

The RSVP simulation module is used to rehearse the command sequences for the MER rovers. It consists of two independent simulators, one for the mobility commands and one for the IDD commands. The mobility commands, such as ARC (drive an arc) and TURNs (turn in place), are dealt with using pseudo-dynamics, where rover state is obtained by the solution of configuration kinematics, and duration of the commands is computed by using empirical data. The IDD commands are simulated using the on-board flight software. The ground operators apply the IDD placement commands with the identical computing techniques of those installed on the rovers.

2.1 Configuration Kinematics

The numerical algorithms of ROAMS, ROver Analysis Modeling and Simulation [1], are used to compute the rover's configuration kinematics. The kinematics equations of the RBD mechanism and the wheel-terrain contact constraints constitute the rover kinematics. The rover's position (x,y,z), attitude (ψ, ϕ, θ) , e.g. *roll, pitch, yaw,* and rocker-bogie angles (Θ_1,Θ_2,Θ_3,Θ_4), e.g., *right rocker*, *left rocker*, *right bogie*, and *left bogie* respectively, can be computed by solving the following kinematics equations:

$$\Theta_1+\Theta_2=0 \tag{1}$$
$$z_f^r-h(x_f^r,y_f^r)=0 \tag{2}$$
$$z_m^r-h(x_m^r,y_m^r)=0 \tag{3}$$
$$z_r^r-h(x_r^r,y_r^r)=0 \tag{4}$$
$$z_f^l-h(x_f^l,y_f^l)=0 \tag{5}$$
$$z_m^l-h(x_m^l,y_m^l)=0 \tag{6}$$
$$z_r^l-h(x_r^l,y_r^l)=0 \tag{7}$$

where the six triplets ($x_{(.)}^{(.)}$, $y_{(.)}^{(.)}$, $z_{(.)}^{(.)}$) represent the contact location on the wheels. The superscripts ***r*** and ***l*** represent the *left* and *right* side of the rover, while the subscripts ***f***, ***m***, ***r*** represent the *front, middle* and *rear* wheels, respectively. Equation (1) represents the differential constraint that the left and right rocker angles are equal with opposite sign. Equations (2-7) are the wheel-terrain contact constraints, where the triplets, i.e. (x_f^r , y_f^r , z_f^r), are the contact points on the wheels, and where $h(.,.)$ is the height map of the terrain. Equations (1-7) describe the rover's nominal configuration where all six wheels are in contact with the ground. It is important to note that the contact constraints may not all be active in some cases. The numerical solution of Eqns. (1-7) must reflect accurate physical conditions when the rover has settled on rough terrain, such as exhibiting popped up wheel(s) or joint stop constraint.

2.2 Newton-Type Method for Settling on Rough Terrain

Equations (1-7) and the rover's driving command for (x,y,θ) comprise the configuration kinematics system. The solution can be computed by a straightforward application of numerical methods. However, the standard numerical method for this set of nonlinear equations often suffers from instability and inefficiency. In particular, when dealing with a rough terrain profile, the wheels may not be in contact with the ground at all times, thus the eqns. (2-7) can be violated. Moreover, the joint limits of the rockers and bogies (e. g., the bumper-stops) must be enforced in the solution, which can be problematic to the Newton iteration. All these modeling and numerical difficulties are resolved by a novel scheme that controls the iterative solutions to follow accurate physical conditions of the wheel-terrain contacts.

A Newton-type iterative method has been developed to handle the non-smooth wheel-terrain contact equations using global searching and relaxation techniques. The solution maintains a fast convergence rate, while handling joint limits and other special cases. Regular Newton-type iteration requires the underlying solution has second order smoothness [9] to ensure a fast convergence. This prerequisite of robust convergence is violated in our formulation since the roughness of the terrain has been embedded in the contact equations. When a rocker or bogie reaches its limits, the iteration can induce an unpredictable solution of the rover's configuration. To overcome these numerical difficulties, we applied a weight factor to the residual of each contact equation. For example, Eq. 7 becomes:

$$\Delta_r^1(z_r^1 - h(x_r^1, y_r^1)) = 0,$$

where Δ_r^1 is the weight factor of Eq. 7. During the iterations, the weight factor for a given wheel can be reduced to zero to relax the contact condition. Whenever the wheel leaves the ground, its corresponding weight factor is set to zero for a total relaxation of this wheel-terrain contact. The re-scaling of the weight factors is coupled with the global search algorithm, which can detect the joint limits associated with each wheel-terrain contact, and can sample small perturbations around the contact locations to determine when wheel-terrain separation occurs.

The step-selection strategy used in the global search is a backtracking line search algorithm that monitors the progress of the iteration [5]. For a smooth terrain profile, the iterative solution generated by the Newton method converges very rapidly to a local minimum of the nonlinear equations [9]. However, the rate of convergence can decrease tremendously when a non-smooth terrain profile appears. Special care is taken to maintain robust and efficient solution in the case of a non-smooth terrain profile. Although the problem in hand is ill posed (i.e., it is well-known that the Newton method cannot treat non-smooth equations), we developed a heuristic solution to ease the computational difficulty in the iterations via specific search directions. In practice, a search direction to the wheel-terrain contact may not be in-line with the normal direction of the terrain (at the contact location). Instead, it could be along the gravitational force of the rover projected onto the wheel's perimeter. The heuristic leads to modeling the wheel-terrain contact equation as the distance constraint between the wheel center and the terrain profile. The distance is the sum of the wheel radius and the averaged terrain height at the contact center. As shown in all the preliminary testing cases, the modification of the contact equations yields a much-improved convergence in non-smooth terrain profiles, and allows the modified Newton iteration to overcome many local irregularities.

2.3 IDD Simulation

The IDD simulation in RSVP is based on the rover's on-board flight software [8]. The IDD arm is modeled as a 5 degree-of-freedom (DOF) open chain, including actuated shoulder azimuth (Φ_1), shoulder elevation (Φ_2), elbow (Φ_3), wrist (Φ_4), and turret (Φ_5). A closed-form solution of the tool frame position, azimuth, and elevation, e.g., (x_{tool} , y_{tool} , z_{tool}) and (q_{az} , q_{el}) has been developed and implemented for RSVP's interactive simulation. Given the target position and normal vector, the simulation is carried out by using the forward and inverse kinematics of the arm mechanism. The arm's motion can be exercised in the IDD operation window (Fig. 2). To achieve the high accuracy required for instrument placement, the operator must rehearse commands by exploring various arm configurations.

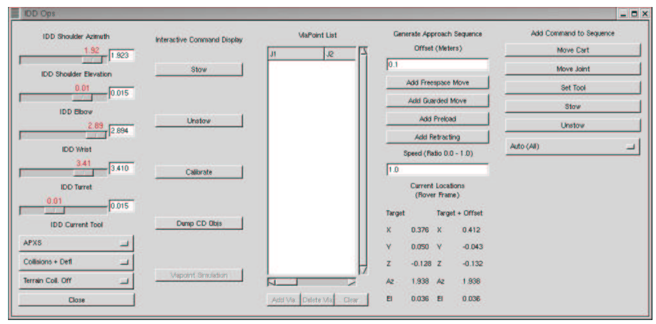

Figure 2. IDD Operation Window

2.4 Sequence Rehearsal

Once a sequence has been created, it is rehearsed to verify that it is within the capabilities of the rover and that the rover will perform the sequence as expected. The operator initiates the rover mobility sequence rehearsal tool and specifies the sequence to be rehearsed. The simulation tool then produces a visualization of the rover activities for comparison with the operator's expectations and alerts the operator to any problems detected in the sequence. Problems could include activities which are too close together or overlapping in time as well as the inability to traverse to the desired waypoint, within a given time window. The duration of the mobility and IDD sequences is computed using the RSVP simulation.

The sequence rehearsal in RSVP is carried out in a three-step procedure: simple-sim, quick-sim, and deep-sim. The simple-sim parses the command sequence, generates the visual components, and sets the initial states for each command. The quick-sim method then computes the rover states based on the numerical algorithm, e.g., inverse kinematics and configuration kinematics. In addition to the configuration kinematics, the deep-sim method uses the rover's on-board control software for accurate prediction of the rover's state and command durations. In both quick-sim and deep-sim steps, collision detection of the rover's instruments is carried out using the terrain model and a geometric model of the rovers. Joint limits are taken into account in the simulation. The high-fidelity command sequence simulation also supports the real-time interactive simulation mode in the visualization tool. This allows the rover planner to create safe and robust command sequences.

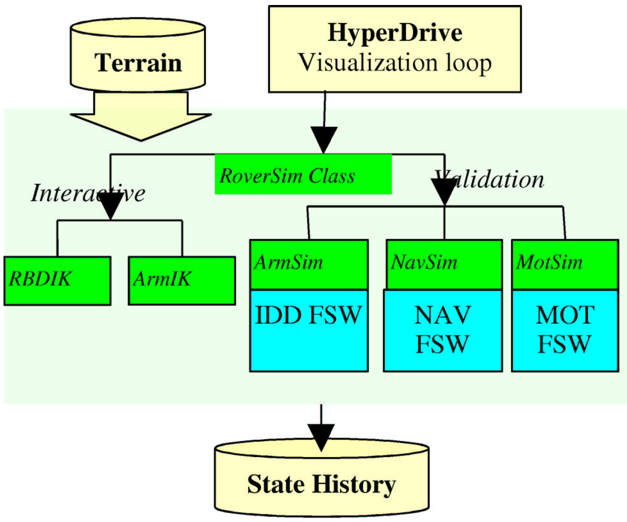

Figure 3. Simulation Architecture

To support the command sequence simulation, the rover is modeled at the subsystem level. As shown in Fig. 3, the rover simulation module consists of subsystem simulators, NavSim, ArmSim and MotSim for the validation mode. The subsystem simulator supports the real-time interactive graphics mode of the visualization tool using the inverse kinematics of RBD and the Arm mechanisms, e.g. RBDIK and ArmIK modules in Fig. 3.

Using on-board navigation and arm control software, the verification mode of the simulator carries out the sequence simulation via a comprehensive set of MER command simulation methods. Sequences are loaded into the simulator and rehearsed using the terrain models and imagery displayed by the visualization tools. The on-board subsystems (FSW NAV and IDD modules) provide the synthetic input, i.e. next desired location, to the simulation. The simulation tool computes the predicted rover and arm states, which are used as sensor inputs to the FSW modules. The resulting series of rover poses constitutes a behavior that can be compared to the expected behavior from the respective mobility sequence.

Another important application in the RSVP simulation is the creation of the on-board navigation map used for traversibility analysis. Using the terrain mesh generated by the down-linked images, the simulation module can create the navigation map using the on-board NAV code [6], see the left picture in Fig. 4. The resultant map represents a snapshot of the rover's perception of the world. This capability enables the planner to oversee the on-board navigation process.

3 Conclusions

The RSVP simulation has been applied to all the mobility and IDD related sequence since the beginning of surface operations of the MER mission. It generates highly accurate predictions that can be used to validate the planned sequences. On both rovers, most of the driving and IDD sequences are validated with extremely high correlation between the actual and predicted telemetry. For the driving commands, the simulation predicted rover states have been consistently within 10% of the actual on-board observation. The simulation of the IDD sequences exhibited accuracy at the millimeter range.

3.1 Tactical Path Planning

For tactical path planning, the RSVP simulation is used to draw out the course and validate the planned path. On a steep climb as shown in Fig. 4, the predicted path (in the blue line) is plotted on the terrain mesh generated using the down linked images from the navigation camera. The red lines show the wheel tracks of the commanded path. Along the way, the RSVP simulation computes the rover's attitude and vertical height. Other sequences used to operate science instruments and the high gain antenna are derived using these predicted rover states.

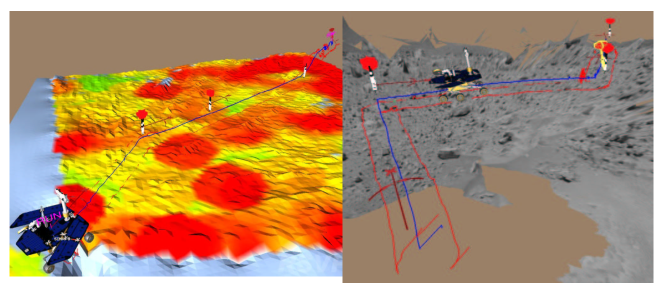

Figure 4. Tactical Path Planning

For regular driving, specifically in flat areas with no hazards, the configuration kinematics simulation generates fairly good predictions. Generally, the actual path reaches the targeted way point on flat terrain. The difference between the planned goal and the actual rover reached location is within 5% of the travel distance. When the slope exceeds 10-15 degrees, the tilt vector of the rovers can be as large as 30 degrees and significant wheel slippage can occur. Although the configuration kinematics analysis doesn't accurately predict the path in these circumstance, the operators can use precomputed slippage tables [6] and the on-board Visual Odometer (Visodom) [7] to provide for accurate driving.

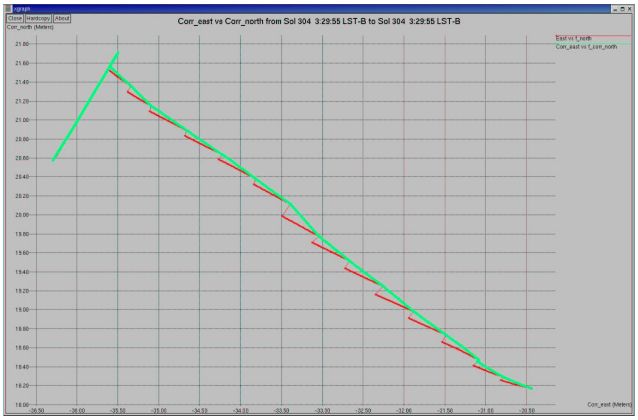

Figure 5a Visodom (green) vs. Wheel Odometer (red)

Opportunity's drive to a target on sol 304 provided one example of such motion. The slope in the drive was 20 plus degrees, and the vehicle's roll and pitch angles were -16 and -13.3 degrees at the starting location. The goal of the traverse was a target 9 meters away on the slope. As shown in Figs. 5a and 5b, the planned path consisted of short 60 centimeter ARC moves toward the target area. Using the on-board Visodom correction, the actual path is the (green) smooth trajectory in Fig. 5. In this drive, Opportunity moved 8.77 meters to within 3 centimeter of the target. Although each of the short ARC commands had 10-30% slip, Visodom corrected the on-board wheel-odometer to achieve a successful drive. The wheel-terrain slippage factor of rover traversal will be incorporated into the simulation for future missions.

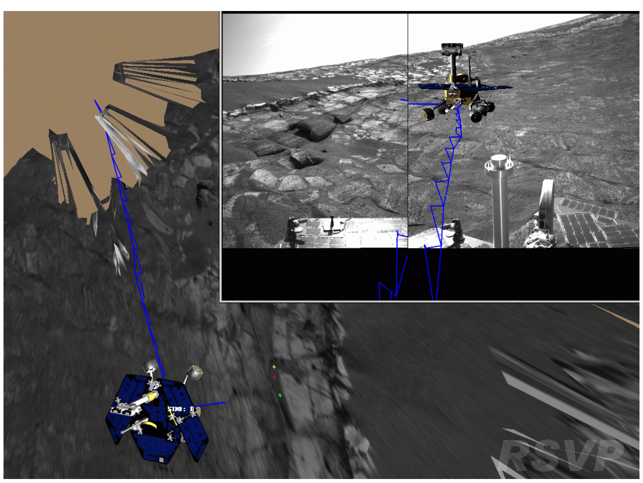

Figure 5b Opportunity Sol 304 Drive

3.2 IDD Sequence Simulation

Using the stereo images generated by the front hazard cameras, the terrain in the IDD work volume is rendered in simulation and used for planning IDD placement. A sequence of IDD commands to take micro imagery of the wall and place the instrument inside the trench were simulated, as shown in the rightt picture in Figure 6. The picture on the left is the "after" shot to confirm the sequence has been executed successfully. The joint encoders indicate that the predicted joint angles are within 0.1 degree of the actual values.

Rehearsal of the IDD sequences has become an essential ingredient for the successful operation of the Mars Exploration Rover mission. To date, the validated sequences have never missed their targets, and the IDD placement error bound is on the order of millimeters.

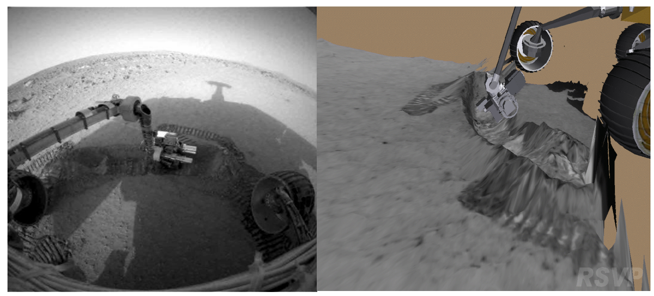

Figure 6. IDD sequence simulation

The accuracy and efficiency of the simulation enable the rover planners to maximize the science return from surface operations. Because the RSVP simulation is based on the kinematics, it is not suited for predicting time-driven events. An extension to full dynamic rover simulation is currently being considered for the future Mars exploration missions.

Acknowledgements

The work described in this paper was performed at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

4 References

[1] J. Yen, A. Jain and B. Balaram, *"ROAMS: Rover Analysis, Modeling and Simulation,"* in the proc. Of 5^{th} iSAIRAS, ESA, Netherland, 1999.

[2] E. J. Haug, *"Computer Aided Kinematics and Dynamics of Mechanical Systems"*, Allyn-Bacon, 1989.

[3] Biesiadecki, J., Henriquez, D., and Jain, A., *"A Reusable, Real-Time Spacecraft Dynamics Simulator,"* Proceedings of the Sixth Digital Avionics Systems Conference, Irvine, CA, October 1997.

[4] A. Jain and G. Man, *"Real-time simulation of the Cassini spacecraft using DARTS: functional capabilities and the spatial algebra algorithm,"* in 5^{th} Annual

Conference on Aerospace Computational Control, JPL, Pasadena, California, Aug. 1992.

[5] Hacot, H., Dubowsky, S., and Bidaud, P. *"Modeling and Analysis of a Rocker-Bogie Planetary Exploration Rover."* Proceedings of the Twelvth CISM-IFToMM Symposium (RoManSy 98), Paris, France, July 1998.

[6] *http://merfsw.jpl.nasa.gov/README-slip.txt*, by Jeffrey Biesiadecki, JPL, 6/22/2004.

[7] S. Goldberg, M. Maimone, and L. Matthies , *"Stereo Vision and Rover Navigation Software for Planetary Exploration,"* in March 2002 IEEE Aerospace conference proceedings, volume 5, Big Sky, Montana, USA, pp 2025-2036.

[8] JPL NTR-41877: *Mars Exploration Rovers Flight Software,* by Glenn Reeves, et. al., January 2005.

[9] Jeng Yen and Linda R. Petzold, "*An Efficient Newton-Type Iteration for the Numerical Solution of Highly Oscillatory Constrained Multibody Dynamic Systems,"* SIAM Journal on Scientific Computing, Volume 19, Number , pp. 1513-1534.